

Учебно-методический комплекс

МОО 01. АДМИНИСТРИРОВАНИЕ И КОНФИГУРИРОВАНИЕ
В СИСТЕМЕ 1С: ПРЕДПРИЯТИЕ



СОДЕРЖАНИЕ

1.	Лекция 1: Введение. Теория бухгалтерского учета.....	4-6
2.	Лекция 2: Модули	6-9
3.	Лекция 3: Программная платформа системы 1С: Предприятия	9-11
4.	Лекция 4: Технологические средства конфигурирования и администрирования и администрирования системы 1С:Предприятия.....	11-15
5.	Лекция 5: Постановка задачи.....	15-18
6.	Лекция 6: Разработка структуры конфигурации.....	19-23
7.	Лекция 7: Конфигуратор. Простейший кадровый учет.....	23-38
8.	Лекция 8: Администрирование Базы данных.....	38-42
9.	Лекция 9: Синтаксис языка 1С.....	43-49
10.	Лекция 10: Конструкция языка 1С.....	49-52
11.	Лекция 11: Справочники.....	52-58
12.	Лекция 12: Основы программирования.....	58-65
13.	Лекция 13: Документы.....	66-68
14.	Лекция 14: Составление и проведение документов.....	69-73
15.	Лекция 15: Отчеты.....	73-76
16.	Лекция 16: Проведение сформированного отчета.....	76-82
17.	Лекция 17: Регистры.....	82-86
18.	Лекция 19: Запросы.....	86-93
19.	Лекция 20: Функции в запросе.....	93-99
20.	Лекция 21: Таблица значений.....	99-101
21.	Лекция 22: Сортировка таблиц значений.....	101-104
22.	Лекция 23: Список значений. Текст.....	105-107
23.	Лекция 24: Выгрузка списка значений.....	107-110
24.	Лекция 25: Экспорт-импорт данных.....	110-114
25.	Лекция 26: Работа со структурой файла DBF.....	114-115
26.	Лекция 27: Файловая система.....	115-118
27.	Лекция 28: Методы, используемые объектом.....	118-124
28.	Лекция 29: Объект Диаграмма.....	125-129
29.	Лекция 30: Работа с сериями диаграммы.....	130-131
30.	Лекция 31: Форма.....	132-141
31.	Лекция 32: Слои и закладки на форме.....	142-145
32.	Лекция 33: Системные процедуры и функции. Предопределенные процедуры.....	148-153
33.	Лекция 34: Конструкторы.....	153-157
34.	Лекция 35: Компонента Расчет Зарплаты.....	157-162
35.	Лекция 36: Журналы расчетов. Правила перерасчета.....	162-169

36	Лабораторная работа № 1 Знакомство, создание информационной базы.....	170- 173
37	Лабораторная работа № 2 Подсистемы.....	174- 186
38	Лабораторная работа № 3 Справочники.....	186- 195
39	Лабораторная работа № 5.....	196- 203
40	Лабораторная работа № 6 Регистры накопления.....	204- 217
41	Лабораторная работа № 7 Простые отчеты.....	217- 222
42	Лабораторная работа № 8.....	222- 229
43	Лабораторная работа № 9 Периодические регистры сведений.....	229- 235
44	Лабораторная работа № 10 Перечисления.....	235- 239
45	Лабораторная работа № 11 Проведение документа по нескольким регистрам.....	240- 245
46	Лабораторная работа № 12 Оборотные регистры накопления.....	246- 252
47	Лабораторная работа № 14 Отчеты. Выбор данных из двух таблиц.....	253- 260
48	Лабораторная работа № 15 Отчеты. Вывод данных по дням в выбранном период.....	260- 275
49	Лабораторная работа № 16 Отчеты. Получение актуальных сведений из периодического регистра сведений.....	275- 280
50	Лабораторная работа № 17 Отчеты. Использование вычисляемого поля в отчете.....	281- 284
51	Лабораторная работа № 18 Отчеты. Вывод данных в таблицу.....	284- 286
52	Контрольно-измерительный материал.....	286- 293
53	Литература.....	293

Лекция 1: Введение. Теория бухгалтерского учета

Развитие общественного производства привело к разделению хозяйственного учета на три вида учета:

Оперативный:

это система наблюдения и текущего контроля за отдельными фактами или процессами с целью руководства ими в момент их совершения

Бухгалтерский:

это система непрерывного взаимосвязанного отображения экономической информации с целью управления и контроля за финансовой деятельностью

Статистический:

это система сбора и обобщения информации о массовых (общественных) явлениях, фактах или процессах с целью управления и контроля за ними

В международных стандартах

бухгалтерский учет определяется как информационная система, которая оценивает, обрабатывает и передает финансовую информацию об определенном экономическом объекте.

Бухгалтерский учет

Представляет собой упорядоченную систему сбора, регистрации и обобщения информации в денежном выражении об имуществе, обязательствах организаций и их движении путем:

- сплошного;
- непрерывного;
- документального учета всех хозяйственных операций

предмет бухгалтерского учета ограничивается рамками хозяйствующего субъекта;

он служит для управления производственными коллективами, выявления результатов хозяйственной деятельности и контроля за ними ;

является источником данных для статистических группировок и сводок;

основной измеритель - денежный. Кроме того, применяются натуральные и трудовые измерители.

Основные функции бухгалтерского учета

Информационная – фиксирует фактическое состояние хозяйствующего объекта;

Контрольная – обеспечивает контроль за выполнением планов, мерой труда и мерой потребления, сохранностью собственности;

Управления – отражает обратную связь в хозяйственной системе и ход выполнения управленческих решений.

Бухгалтерский учет обеспечивает информацией руководство организации и всех заинтересованных лиц путём регистрации, обработки и хранения учетной записи.



Пользователи бухгалтерской информации

Внешние

Внутренние

Внешние:

Имеющие прямой финансовый интерес – инвесторы, банки, поставщики и другие кредиторы;

Имеющие косвенный финансовый интерес- органы, уполномоченные управлять госимуществом, налоговые,

финансовые;

Без финансового интереса – органы статистики, аудиторы, арбитраж.

Внутренние:

Администрация, советы (директоров, акционеров);

Собственники, совладельцы;

Менеджеры, руководство и специалисты подразделений;

Члены коллектива предприятия.

Разновидности бухгалтерского учета

Финансовый учет

Управленческий учет

Налоговый учет

Финансовый учет

- Учет денежных средств и расчетов;
- Учет производственных запасов;
- Учет основных средств и нематериальных активов;
- Учет продукции и ее реализации;
- Учет экспортно-импортных операций;
- Учет фондов, резервов, кредитов и займов;
- Учет финансовых результатов и их использования;
- Бухгалтерская отчетность.

Управленческий учет

- Учет прямых затрат
- Учет косвенных затрат
- Обобщение затрат на производство

Налоговый учет

это система обобщения информации для определения облагаемой базы по налогу на прибыль на основе данных первичных документов в соответствии с порядком, предусмотренным главой 25 Налогового кодекса РК

Подтверждением данных налогового учета являются:

первичные учетные документы (включая справку бухгалтерии);

аналитические регистры налогового учета (формы разрабатываются самостоятельно);

расчет налоговой базы.

Состав имущества организации

Внеоборотные активы (основные средства, нематериальные активы, капитальные вложения);

Оборотные активы (материальные оборотные средства, денежные средства, краткосрочные финансовые вложения, средства в расчетах).

Источники формирования имущества организации:

Собственные (собственный капитал): уставный капитал, добавочный капитал, резервный капитал, целевое финансирование и поступление, нераспределенная прибыль;

Заемные (создание за счет обязательств): кредиты банка, заемные средства, кредиторская задолженность, обязательства по распределению.



Лекция 2: Модули

Модули – это те объекты, где содержится программный код.

В Платформе существует достаточно большое количество видов модулей, каждый из которых имеет свое предназначение и особенности. Любая строка кода должна находиться в каком-либо модуле. Различают модули общего предназначения и модули объекта. Некоторые модули могут быть скомпилированы как на Клиенте, так и на Сервере, а некоторые только на Сервере.

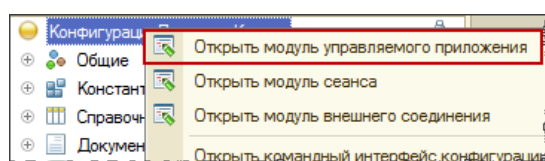
Модуль может состоять из нескольких разделов. В разделе описания переменных описываются локальные переменные данного модуля, которые впоследствии могут быть использованы в любой процедуре.

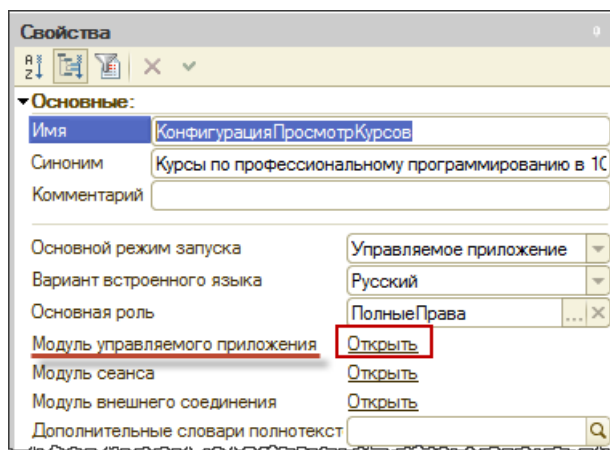
Модули бывают разные

- Модуль приложения
- Модуль внешнего соединения
- Модуль сеанса
- Общие модули
- Модуль формы
- Модуль объекта
- Модуль менеджера

Модуль приложения

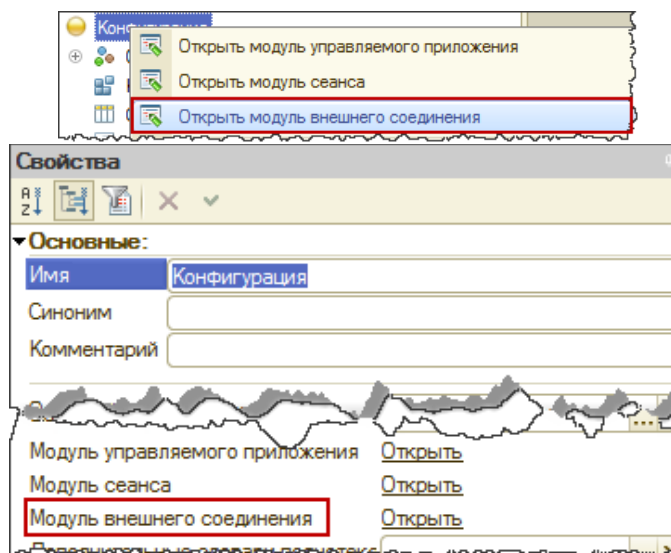
Данный модуль предназначен для того, чтобы обработать события запуска приложения и завершения его работы. Например, при запуске приложения можно загружать курсы валют из Интернета. При завершении приложения можно удостовериться у пользователя о его намерениях закончить работу. Также в модуле приложения существуют специальные обработчики, которые позволяют перехватить внешние события от оборудования.





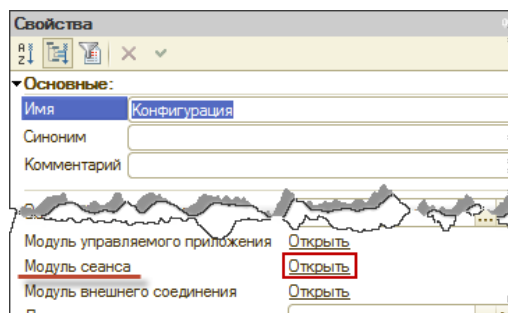
Модуль внешнего соединения

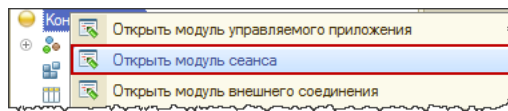
По аналогии с модулем приложения данный модуль предназначен для того, чтобы обработать событие открытия программы и событие завершения работы. В отличие от модуля приложения, который иницируется в момент интерактивного запуска приложения, модуль внешнего соединения работает в режиме com-соединения, т.е. когда создается объект 1С:Предприятие 8 и осуществляется подключение к определенной базе.



Модуль сеанса

Данный модуль нужен для того, чтобы инициализировать параметры сеанса. Параметры сеанса — это быстрые глобальные переменные, значения которых доступны в любом месте конфигурации. Открыть Модуль сеанса можно либо через контекстное меню, либо через палитру свойств корневого узла. При старте приложения данная процедура вызывается самой первой. В Модуле сеанса описываются различные действия по инициализации параметров сеанса в зависимости от разных условий. Модуль сеанса всегда исполняется в привилегированном режиме. Это означает, что не будет выполняться проверка прав доступа при обращении к базе данных.



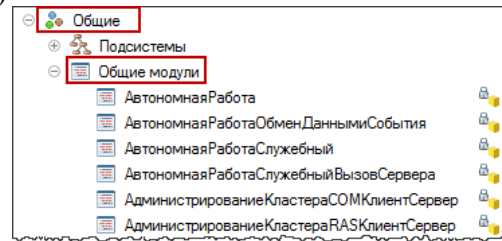


Общие модули

Как использовать общий модуль

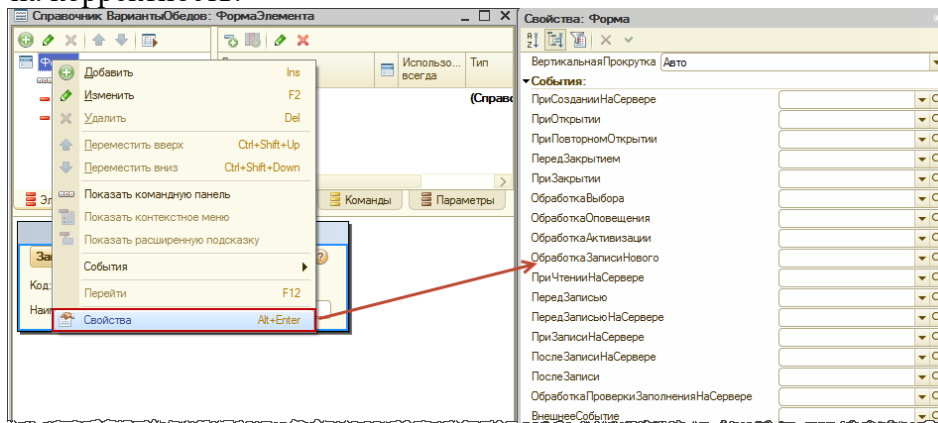
Хороший тон — поместить процедуру или функцию в общий модуль, если она вызывается в более чем одном месте. Во-первых, если процедура корректируется — её надо править только в одном месте. Во-вторых этим достигается больший порядок в коде.

Общие модули 1С — объект метаданных конфигурации 1С 8.3 и 8.2, который хранит в себе программный код, который часто вызывается в конфигурации. Функцию/процедуру можно вызвать из любого места конфигурации (если она экспортная).



Модуль формы

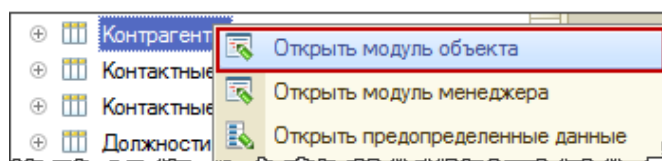
Данный модуль предназначен для того, чтобы обработать действия пользователя. Например, описать алгоритм реакции программы при нажатии кнопки. Или, например, в момент ввода в поле значения сразу же выполнить проверку на корректность.

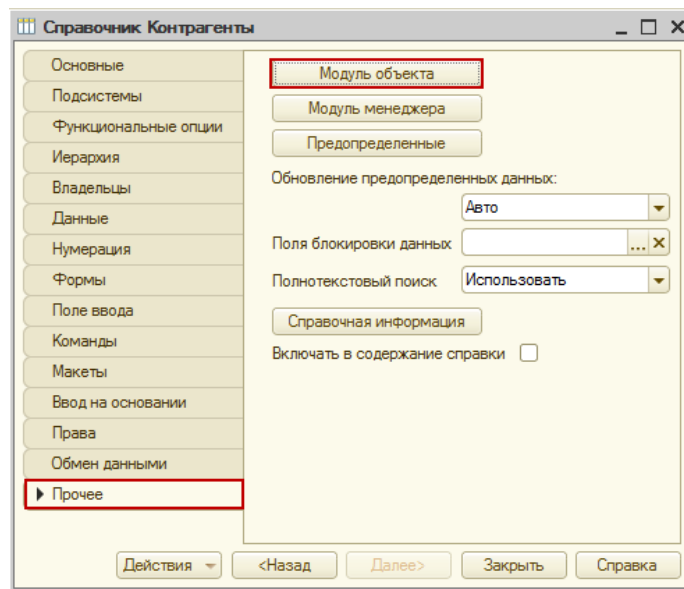


Существуют формы управляемые и формы обычные. Модули данных форм различаются прежде всего тем, что модуль управляемой формы четко разделяется на контекст. Каждая процедура (функция) должна иметь директиву компиляции. В обычной форме весь код используется на Клиенте. В модуле управляемой формы можно объявлять процедуры и функции, можно объявлять переменные и описывать раздел основной программы.

Модуль объекта

Данные модули характерны для справочников, документов, планов видов расчетов, планов счетов и многих других объектов. Модуль объекта предназначен для обработки стандартных событий. Например, событие на ввод элемента справочника, событие на запись элемента, удаление и т.д. записи существует и в Модуле формы. Но событие записи в Модуле формы возникает в процессе интерактивной записи, при работе с конкретной формой



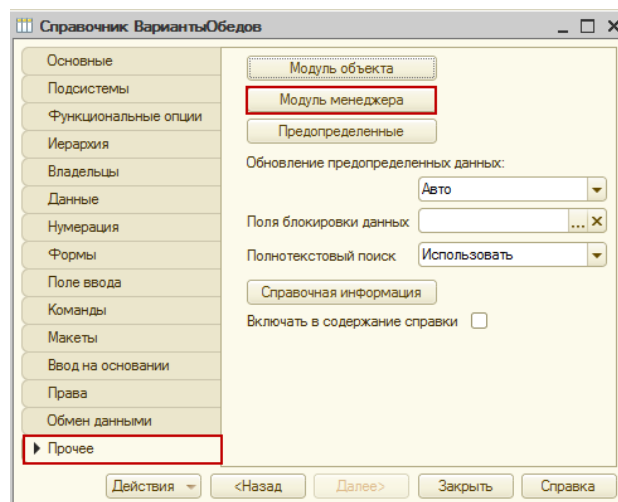
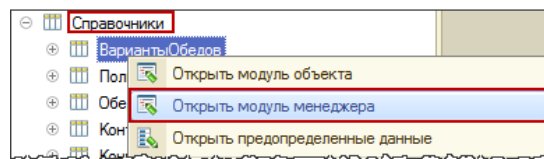


Модуль менеджера

Данный модуль существует для многих объектов (справочники, документы, регистры и др.). Модуль открывается либо через контекстное меню для объекта, либо через Палитру свойств, либо через окно редактирования.

Кроме этого в Модуле менеджера можно создать дополнительные методы и указать, что они являются экспортными. В этом случае возможно обращение к данным методам из вне.

В Модуле менеджера можно переопределить некоторые стандартные события. Например, в Обработке Получения Данных Выбора, когда выбирается элемент из справочника, можно сделать какую-то дополнительную фильтрацию или проверку.



Лекция 3: Программная платформа системы 1С: Предприятия

Система программ "1С:Предприятие 8" включает в себя платформу и прикладные решения, разработанные на ее основе, для автоматизации деятельности организаций и частных лиц. Сама платформа не является программным продуктом для использования конечными пользователями, которые обычно работают с одним

из многих прикладных решений (конфигураций), разработанных на данной платформе. Такой подход позволяет автоматизировать различные виды деятельности, используя единую технологическую платформу.



Области применения

Гибкость платформы позволяет применять 1С:Предприятие 8 в самых разнообразных областях: автоматизация производственных и торговых предприятий, бюджетных и финансовых организаций, предприятий сферы обслуживания и т.д.; поддержка оперативного управления предприятием; автоматизация организационной и хозяйственной деятельности; ведение бухгалтерского учета с несколькими планами счетов и произвольными измерениями учета, регламентированная отчетность; широкие возможности для управленческого учета и построения аналитической отчетности, поддержка многовалютного учета; решение задач планирования, бюджетирования и финансового анализа; расчет зарплаты и управление персоналом; другие области применения.

Технологическая платформа

Платформа 1С:Предприятие 8 была создана с учетом опыта применения системы программ 1С:Предприятие 7.7. С 2003 года платформа показывает активное развитие. Основная задача платформы заключается в предоставлении среды разработки (framework) разработчикам прикладных решений. Платформа обеспечивает разработчикам повышение уровня абстракции при разработке и использовании прикладных решений. Это позволяет перейти от технических и низкоуровневых понятий к более содержательным и высокоуровневым. Позволяет приблизить эти понятия к языку пользователей и специалистов в предметной области. В конечном итоге это значительно ускоряет и унифицирует разработку прикладного решения и его сопровождение. Одновременно с этим платформа решает и традиционные задачи, связанные с производительностью, эргономикой, функциональностью и тому подобное.

Прикладные решения

Тиражные прикладные решения, предназначены для автоматизации типовых задач учета и управления в коммерческих предприятиях реального сектора и бюджетных организациях. Отраслевые и региональные прикладные решения предназначены для автоматизации отдельных направлений или областей деятельности предприятий. Все они сертифицированы на требования "1С:Совместимо". В каждом программном продукте сочетается использование стандартных решений (общих для всех или нескольких программ) и максимальный учет специфики задачи конкретной отрасли или рода деятельности предприятия. Разработку тиражных и отраслевых решений выполняют 1С:Партнеры со статусом "Центр разработки тиражных решений" (1С:Центр Разработки)

1С:Библиотека стандартных подсистем

«1С:Библиотека стандартных подсистем» (БСП) предоставляет набор универсальных функциональных подсистем, готовые разделы для пользовательской документации и технологию для разработки прикладных решений на платформе «1С:Предприятие». С применением БСП становится возможной быстрая разработка новых конфигураций с уже готовой базовой функциональностью, а также включение готовых

функциональных блоков в существующие конфигурации. Использование БСП при разработке прикладных решений на платформе 1С:Предприятие позволит также достичь большей стандартизации конфигураций, что уменьшит время на изучение и внедрение прикладных решений за счет их унификации по набору используемых стандартных подсистем.

Входящие в БСП подсистемы охватывают такие области, как:

Администрирование пользователей и прав доступа;

Средства администрирования и обслуживания (установка обновлений, резервное копирование, дополнительные отчеты и обработки, оценка производительности и др.);

Сервисные подсистемы (история изменений объектов, заметки и напоминания, печать, полнотекстовый поиск, присоединенные файлы, электронная подпись и др.);

Технологические механизмы и программные интерфейсы (процедуры и функции общего назначения, обновление версии ИБ, работа в модели сервиса и др.);

Нормативно-справочная информация и классификаторы (адресный классификатор, банки, валюты и др.);

Интеграция с другими программами и системами (обмен данными, работа с почтовыми сообщениями, отправка SMS, рассылка отчетов и др.);

Прикладные подсистемы и рабочие места пользователей (анкетирование, бизнес-процессы и задачи, взаимодействия, варианты отчетов и др.)

Всего в БСП входит более 60 подсистем. Полный состав подсистем, их назначение, порядок внедрения и рекомендации по использованию приведены в документации к библиотеке (<https://its.1c.kz/db/bspdoc>).

Внедрения

Внедрения выполняются силами внедренческих центров сети 1С:Партнеры и включают обучение пользователей предприятия работе с системой, а также реализацию особенностей деятельности конкретного предприятия или специальные пожелания заказчика.

Внедрения и адаптации прикладных решений также могут выполняться и силами IT-специалистов заказчика, самостоятельно, или во взаимодействии с 1С:Партнерами.

Для комплексной автоматизации предприятия и выполнения масштабных и сложных проектов рекомендуется приглашать партнеров со статусами 1С:Центр компетенции по корпоративным решениям (1С:КОРП) и 1С:Центр компетенции по ERP-решениям (1С:ERP)

внедренные решения на базе системы программ "1С:Предприятие 8"

поиск партнеров

Лекция 4: Технологические средства конфигурирования и администрирования и администрирования системы 1С:Предприятия

Система программ «1С:Предприятие 8.0» включает в себя платформу и прикладные решения, разработанные на ее основе, для автоматизации деятельности организаций и частных лиц. Сама платформа не является программным продуктом для использования конечными пользователями, которые обычно работают с одним из многих прикладных решений (конфигураций), разработанных на данной платформе. Такой подход позволяет автоматизировать различные виды деятельности, используя единую технологическую платформу.

Области применения:

- Гибкость платформы позволяет применять 1С:Предприятие 8.0 в самых разнообразных областях;
- автоматизация производственных и торговых предприятий, бюджетных и финансовых организаций, предприятий сферы обслуживания и т.д.
- поддержка оперативного управления предприятием;
- автоматизация организационной и хозяйственной деятельности;
- ведение бухгалтерского учета с несколькими планами счетов и произвольными измерениями учета, регламентированная отчетность;
- широкие возможности для управленческого учета и построения аналитической отчетности, поддержка многовалютного учета;
- решение задач планирования, бюджетирования и финансового анализа;
- расчет зарплаты и управление персоналом;

- другие области применения.

Технологическая платформа 1С:Предприятия 8.0

Платформа 1С:Предприятие 8.0 была создана с учетом 6-летнего опыта применения системы программ 1С:Предприятие 7.7, которую используют десятки тысяч разработчиков. В настоящее время более 700 000 организаций использует для автоматизации своей деятельности продукты, входящие в систему программ «1С:Предприятие». Несмотря на значительные изменения, новая версия 8.0 сохранила идеологическую преемственность с предыдущими версиями.

Общая архитектура

- Интернационализация
- Экономическая и аналитическая отчетность
- Работа пользователя
- Обмен данными
- Web-расширения
- Интерфейсные механизмы
- Система прав доступа
- Прикладные механизмы
- Интеграция

Общая архитектура

- Общие механизмы
- Масштабируемость
- Средства программирования
- Средства администрирования

Средства администрирования

Основные средства администрирования системы 1С:Предприятие реализованы в составе конфигулятора. Однако есть ряд механизмов и утилит, которые не входят в состав конфигулятора, хотя также имеют отношение к администрированию системы 1С:Предприятие. Перечислим наиболее важные механизмы и инструменты, входящие в состав средств администрирования:

Аутентификация

- Система 1С:Предприятие обеспечивает одновременную работу нескольких пользователей с одним прикладным решением. Для разграничения прав доступа пользователей к данным информационной базы, в системе ведется список пользователей, допущенных к работе с данным прикладным решением.
- В момент подключения очередного пользователя к прикладному решению механизм аутентификации позволяет узнать, кто именно из пользователей, перечисленных в списке пользователей, подключается к прикладному решению. Эта информация может быть в дальнейшем использована для выбора соответствующего интерфейса пользователя, разрешения доступа к той или иной информации и пр.

Список пользователей

В системе 1С:Предприятие существует возможность создания списка пользователей, которым разрешена работа с системой. Ведение такого списка позволяет описывать полномочия различных пользователей системы на доступ к той или иной информации, а также назначать пользовательские интерфейсы.

Активные пользователи

- Список активных пользователей позволяет получать информацию о том, кто из пользователей работает с информационной базой в данный момент:
- Список активных пользователей содержит информацию об имени пользователя, режиме, в котором пользователь использует систему, времени начала его работы и пр. Пользователь имеет возможность отсортировать список по любой из колонок, вывести список активных пользователей на печать в виде текстового или табличного документа. Кроме этого, из списка активных пользователей можно открыть журнал регистрации системы, или просмотреть историю работы пользователя - содержимое журнала

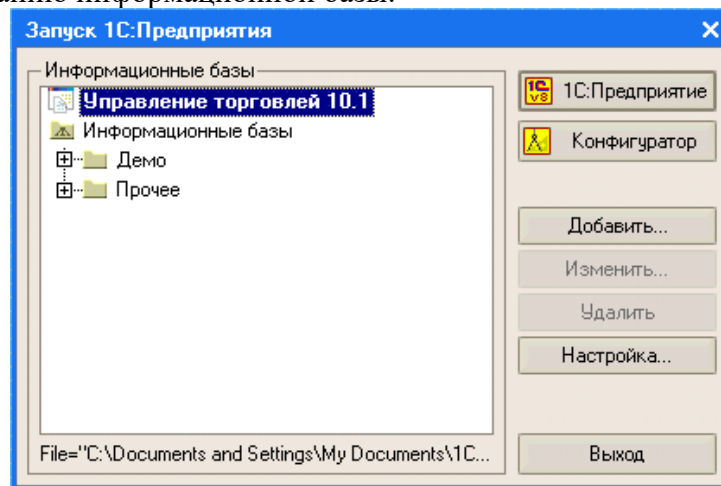
регистрации, отфильтрованное по тому пользователю, на котором установлен курсор.

Журнал регистрации

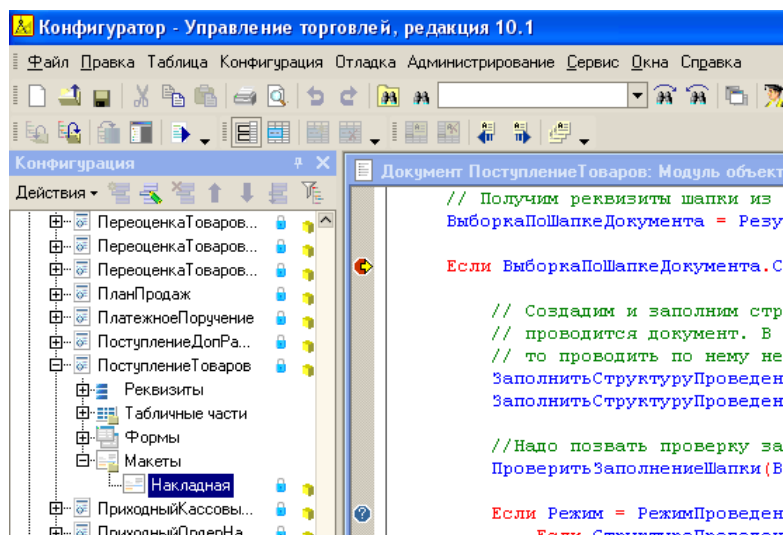
- Журнал регистрации содержит информацию о том, какие события происходили в информационной базе в определенный момент времени или какие действия выполнял тот или иной пользователь:
- Журнал регистрации доступен как в режиме 1С:Предприятие, так и в режиме Конфигуратор.
- В режиме 1С:Предприятие по щелчку мыши в полях Данные и Представление данных можно перейти к тому объекту прикладного решения, который указан в записи журнала регистрации.
- Информацию, находящуюся в журнале регистрации, можно отбирать по большому количеству критериев. Например, можно отобразить только информацию о том, какие документы изменялись определенным пользователем в заданный промежуток времени:
-
- Кроме этого, журнал регистрации поддерживает динамическую фильтрацию событий, при которой новые события, удовлетворяющие наложенному фильтру, будут появляться в списке.
- Существует возможность настройки уровня событий, отображаемых в журнале регистрации:
- Кроме этого разработчик может самостоятельно добавлять записи в журнал регистрации, используя средства встроенного языка. Такая возможность позволяет настраивать журнал регистрации под нужды конкретного прикладного решения.
- В процессе длительной эксплуатации системы в журнале регистрации может накапливаться значительное число записей. Поэтому поддерживается возможность сокращения журнала регистрации и удаления записей, ставших неактуальными. При сокращении журнала регистрации можно записать удаляемые события в файл, если предвидится необходимость их анализа в будущем:
- Используя средства работы со списками, разработчик имеет возможность выгрузить журнал регистрации в текстовый или табличный документ, который в дальнейшем может быть сохранен в собственном формате или, например, формате листа Excel или документа HTML.
- Кроме этого поддерживается выгрузка журнала регистрации в формате XML, что может использоваться для последующего анализа выгруженных записей средствами встроенного языка.

Конфигуратор

Конфигуратор представляет собой специальный режим запуска системы 1С:Предприятие 8.0. Этот режим позволяет выполнять изменение существующих прикладных решений и создание новых, а также выполнять действия по администрированию информационной базы.



Конфигуратор предоставляет разработчику все необходимые инструменты для "визуального" описания структуры прикладного решения, создания форм диалогов и выходных документов, механизмов работы пользователей с данными (интерфейсов) и прав доступа различных групп пользователей к различной информации. Кроме этого конфигуратор позволяет создать и настроить взаимодействие различных элементов прикладного решения друг с другом и написать модули на встроенном языке для обработки различных событий и реализации специфических алгоритмов взаимодействия, обработки входных и выходных данных



Управление торговлей

- "1С:Предприятие 8.0. Управление торговлей" — это современный инструмент повышения эффективности бизнеса торгового предприятия.
- Прикладное решение "1С:Предприятие 8.0. Управление торговлей" в комплексе решает задачи управленческого и оперативного учета, анализа и планирования; автоматизирует торговые, финансовые и складские операции; обеспечивает современный уровень управления предприятием. Предметная область, автоматизируемая прикладным решением, может быть представлена в виде следующей схемы.

Управление торговлей



Управление торговлей

- Прикладное решение автоматизирует следующие направления хозяйственной деятельности:
- управление продажами (включая оптовую, розничную и комиссионную торговлю);
- управление поставками;
- планирование продаж и закупок;
- управление складскими запасами;
- управление заказами покупателей;
- управление отношениями с клиентами;
- анализ товарооборота предприятия;
- анализ цен и управление ценовой политикой;
- мониторинг и анализ эффективности торговой деятельности.

Руководителю предприятия

Руководителю предприятия нужна реальная отдача от системы автоматизации.

"1С:Предприятие 8.0. Управление торговлей" повышает эффективность работы предприятия за счет автоматизации рутинных операций, за счет ведения учета в реальном масштабе времени, за счет быстрой и

удобной подготовки информации для принятия решений на разных уровнях. Система очень быстро запускается в эксплуатацию и начинает приносить отдачу. При изменении масштабов, подходов к управлению или организации работ на предприятии перестройка системы не требует больших затрат. Это достигается за счет построения бизнес-решений на мощной современной технологической платформе. Важным достоинством системы "1С:Предприятие" является ее широкая популярность: более 3000 специализированных фирм и множество сертифицированных специалистов готовы помочь вашему предприятию в установке, эксплуатации и дальнейшем совершенствовании системы автоматизации.

Руководителям и специалистам торговых подразделений

В новом прикладном решении уделен максимум внимания тому, чтобы обеспечить эффективность вашей ежедневной работы. Широкие возможности подготовки всех необходимых документов, управления товародвижением и ценообразованием, приема заказов и контроля их исполнения, оптимизации складских запасов, анализа товарооборота, планирования закупок и поставок сделают ее вашим надежным помощником — за какое бы направление или участок вы ни отвечали. "1С:Предприятие 8.0. Управление торговлей" позволит создать в вашей компании единое информационное пространство, организовать четкое взаимодействие подразделений, поможет сотрудникам работать единой командой.

Работникам учетных служб

Программы системы "1С:Предприятие" стали индустриальным стандартом учетных решений. Программа "1С:Предприятие 8.0. Управление торговлей" вобрала все лучшее, что было наработано за годы развития ее предшественницы — программы "1С:Торговля и Склад 7.7". Зарекомендовавшая себя на практике функциональность этой, наиболее массово применяемой сегодня в стране системы автоматизации торгово-складского учета, дополнена целым рядом новых возможностей. Вы получаете новый инструмент, который поможет избавиться от рутинной работы и приблизить учет к потребностям реального бизнеса. При этом регулярная информационно-методическая поддержка фирмы "1С" позволит вести его в соответствии с действующим законодательством. Разумеется, "1С:Предприятие 8.0. Управление торговлей" позволяет передать необходимые учетные данные в бухгалтерскую систему.

IT-специалистам предприятий

"1С:Предприятие 8.0" дает вам в руки отвечающий самым современным стандартам комплекс инструментальных средств и технологий для разработки, модификации, администрирования и сопровождения корпоративной информационной системы. Все эти средства входят в поставку выпускаемых нами тиражных продуктов "1С:Предприятия 8.0". Вы без дополнительных затрат получаете точно такие же инструменты, какие используют разработчики фирмы "1С". В результате установленная на предприятии система не будет для вас "черным ящиком". Опыт показывает, что большинству системных администраторов и специалистов по автоматизации достаточно нескольких дней, чтобы освоить принципы построения решений "1С:Предприятия" — после этого вы сможете не только сопровождать, но и адаптировать систему в соответствии с потребностями своих организаций.

Лекция 5: Постановка задачи

Назначение и возможности ППП 1С:Предприятие

1С:Предприятие -- это специализированная объектно-ориентированная система управления базами данных (СУБД), предназначенная для автоматизации деятельности предприятия. 1С Предприятие представляет собой систему прикладных решений, построенных по единым принципам и на единой технологической платформе.

Система 1С:Предприятие предназначена для решения широкого спектра задач авто-матизации учета и управления, стоящих перед динамично развивающимися современными предприятиями.

Система 1С:Предприятие для всех прикладных решений независимо от отраслевой специфики обеспечивает:

- возможность использования системы от локального компьютера до десятков пользователей в локальной сети;
- использование файлового варианта или варианта "клиент-сервер" (MS SQL Server);
- возможность развертывания работы на нескольких территориально удаленных точках с периодическим обменом информацией;
- возможность использования современных технологий (WEB, XML, интеграция с другими программными системами и различным торговым оборудованием). Наличие единой технологической платформы и общей методологии позволяет создавать специализированные и индивидуальные решения на базе стандартных, добавляя в них только необходимые отличия, учитывающие специфику отрасли или конкретного предприятия.
- С экономической точки зрения это позволяет обеспечить достаточно низкую стоимость отраслевых и индивидуальных решений, так как затраты на их создание существенно ниже, чем затраты на разработку программы "с нуля".
- Это обеспечивает высокую скорость создания и внедрения решений, так как максимально используется отработанная функциональность и методология, содержащиеся в типовых решениях.
- Очень важным преимуществом такого подхода является унификация обучения пользователей. Обучившись на курсах по работе 1С Предприятие 7.7 или имея опыт работы с какой либо из программ, пользователь достаточно быстро осваивает возможности специализированных или индивидуальных решений.
- Стандартизация платформы также существенно упрощает и администрирование системы, так как функции администрирования практически не зависят от конкретного прикладного решения. Большинство системных администраторов и специалистов по автоматизации уже имеют опыт администрирования и даже модификации

прикладных решений 1С Предприятия 7.7.

Важным преимуществом 1С Предприятия 7.7 является открытость системы

Для руководителя, принимающего решение о выборе средства автоматизации, достаточно важно быть уверенным, что система не будет для предприятия "черным ящиком",

и существует реальная возможность понять работу системы и, при необходимости, изменить. Эта работа может быть выполнена как франчайзинговыми организациями, специализирующимися на поддержке 1С Предприятия 7.7, так и специалистами ИТ-служб самого предприятия.

Фирма 1С обеспечивает регулярную поддержку стандартных прикладных решений и самой платформы. Платформа 1С Предприятия 7.7 обеспечивает возможность совмещения обновлений прикладного решения, производимого фирмой 1С или разработчиком специализированного решения, с индивидуальными изменениями, внесенными при внедрении системы.

Устройство системы 1С Предприятие 7.7 позволяет достаточно быстро вводить в курс дела новых специалистов и передавать поддержку прикладного решения тому, кто сможет обеспечить наилучшее обслуживание. Таким образом, наличие реальной индустрии внедрения и поддержки решений системы 1С Предприятие является для заказчика гарантией независимости от единственной фирмы или специалиста, "живучести" внедренного проекта и его беспрепятственного сопровождения и развития.

Структура пакета 1С:Предприятие

1С:Предприятие является гибкой настраиваемой системой для решения широкого круга задач в сфере автоматизации деятельности предприятий. Это обеспечивается общей структурой системы, которая представлена на рис. 13.



Рис. 13. Структурная схема 1С: Предприятия

В основе системы - технологическая платформа 1С, выполняющая общесистемные функции. Прикладные задачи (бизнес-логику) и функции взаимодействия с пользователем реализуют различные решения, т.н. конфигурации, на базе технологической платформы.

Технологическая платформа

Ключевой элемент всей системы – технологическая платформа, на базе которой со-здаются прикладные программы, с которыми имеют дело конечные пользователи. Она определяет потенциальные возможности для решения задач автоматизации предприятий-заказчиков, как в отношении функционала, так и масштабирования. Она состоит из трех основных компонентов: среды исполнения (собственно "1С:Предприятие", здесь работают пользователи), инструмента разработки ("Конфигуратор") и средств администрирования.

Масштабируемость платформы 1С обеспечивается благодаря возможности реализации трех основных вариантов ее применения: однопользовательская, файл-серверная (для группы 5–10 человек) и клиент-серверная (десятки и даже сотни одновременно работающих пользователей). При этом важно, что все эти возможности могут быть реализованы без переработки прикладного решения. Таким образом, разработчик тиражных решений может выпускать одно и то же приложение для использования в организациях самого раз-ного масштаба — с точки зрения пользователя, во всех указанных вариантах прикладное решение будет работать одинаково.

В версии "1С:Предприятие 8.0", введена поддержка современной трехуровневой ар-хитектуры "клиент--сервер". В этом случае программа, работающая у пользователя, взаи-модействует с сервером приложения, который обращается к информационной базе, храня-щейся в базе данных MS SQL Server, передавая туда клиентские запросы. Повышение производительности может достигаться путем увеличения вычислительных мощностей центральных компьютеров: запуска сервера приложений и сервера баз данных на разных компьютерах и использования многопроцессорных систем.

Технологическая платформа – не самостоятельный программный продукт, она про-дается и поставляется партнерам и конечным пользователям только в составе прикладных решений, т. е. заказчики в принципе имеют все возможности использовать ее для дора-ботки и адаптации приложений под конкретные задачи.

Технологическая платформа постоянно развивается и расширяется как в технологи-ческом, так и прикладном направлении.

Прикладные решения

Прикладные решения разрабатываются на встроенном высокоуровневом языке и исполняются в среде "1С:Предприятие". В силу исторических причин такие приложения часто называют "конфигурациями", поскольку для среды исполнения они представляют собой базу данных с набором объектов, описаниями взаимосвязей между ними, программ-ным кодом и т. д.

Готовые прикладные решения — это основная продукция фирмы "1С". Однако тут нужно подчеркнуть, что бизнес-модель "1С" подразумевает взаимодействие с потребителями только через партнеров, занимающихся продажей, внедрением, адаптацией и сопровождением прикладных решений у заказчиков. Главная сила — франчайзинговая (внедренческая) сеть, в которую входят почти 3 тыс. фирм по всей стране.

С выходом технологической платформы 8.0 фирма "1С" несколько скорректировала свой подход к созданию семейства прикладных программ. Если ранее (в версии 7.7) "1С" шла по пути создания автономных приложений, которые затем объединялись в комплекс, то в 8.0 был сразу реализован подход от общего к частному. Таким базовым продуктом является выпущенное в 2004 г. прикладное решение "1С:Управление производственным предприятием", представляющее собой законченную прикладную платформу, отдельные части которой доступны также и в виде автономных продуктов (рис. 2): "Управление торговлей", "Бухгалтерия", "Зарплата и управление персоналом".

Говоря о семействе прикладных программных продуктов "1С:Предприятие", необходимо иметь в виду, что их разработкой занимается не только сама фирма "1С", но и ее партнеры. При этом действует вполне естественная схема разделения труда: "1С" создает наиболее массовые горизонтальные приложения (спектр которых постоянно расширяется

— бухгалтерия, торговля, управление персоналом, производство, бюджетная сфера, образование и т. д.), а партнеры, в соответствии со своей специализацией, — вертикальные, отраслевые. Чаще всего партнерские решения разрабатываются на основе базовых конфигураций "1С", но бывают варианты, когда они пишутся "с нуля".

Заказчики приобретают готовые приложения "1С" или тиражируемые программы партнеров и довольно часто адаптируют их под свои задачи (с помощью внешних внедренцев или собственными силами). При этом есть два принципиальных подхода: настройку можно проводить, пользуясь высокоуровневыми средствами интегрированной среды исполнения и/или средствами разработки (в том числе с помощью написания программного кода).

Режимы работы 1С:Предприятие

Сама по себе технологическая платформа не может выполнить никаких задач автоматизации, так как она создана для обеспечения работы какой-либо конфигурации. Сама по себе конфигурация так же бесполезна, поскольку, чтобы она могла выполнить те задачи, для которых она создана, необходимо наличие платформы, которая и управляет ее работой.

Для того чтобы обеспечить такие возможности, система «1С:Предприятие 8» имеет 2 основных режима работы:

- Режим 1С:Предприятие (пользовательский), который является основным и служит для работы пользователей системы. В этом режиме пользователи вносят данные, обрабатывают их и получают выходные результаты.
- Режим "Конфигуратор" - используется разработчиками и администраторами баз данных. Именно этот режим и предоставляет инструменты, необходимые для модификации существующей или создания новой конфигурации.

Помимо указанных, в 1С поддерживаются еще 2 режима работы, которые выполняются в среде Конфигуратора и используются для разработки и администрирования:

- режим отладчика - для отладки и оптимизации приложений 1С;
- режим управления пользователями - для администрирования учетных записей пользователей системы.

Лекция 6: Разработка структуры конфигурации

Описание задачи

Мы будем рассматривать основы разработки прикладных *конфигураций* на примере автоматизации деятельности салона красоты "Марина". В салоне работают директор, *администратор*, мастера и другие сотрудники. Основная задача мастеров - оказание услуг клиентам. Салон закупает материалы у поставщиков. Материалы расходуются в ходе оказания услуг, а так же их можно продавать клиентам - физическим лицам или организациям.

Мы автоматизируем *деятельность* салона. В частности, нужно автоматизировать учет материалов, учет деятельности мастеров, учет клиентов с возможностью назначения особых условий обслуживания для постоянных клиентов. Нужно автоматизировать расчет заработной платы для сотрудников, создать отчеты, которые позволят руководству анализировать эффективность деятельности салона. В ходе решения задачи мы столкнемся и со многими другими практическими ситуациями, требующими автоматизации.

Мы не будем стремиться к созданию решения, которое можно будет использовать на практике. Наша основная задача - рассмотреть работу с *объектами* системы, используя избранную предметную область для демонстрации возможностей и особенностей системы. Хотя, с другой стороны, нашу итоговую *конфигурацию*, после доработки, можно будет использовать на практике.

Продолжим знакомство с *Конфигуратором*. А именно, приступим к изучению операций с *объектами*.

1.3. Создаем объекты

Первой операцией с *объектами*, которую мы освоим в *Конфигураторе*, будет создание *объектов*. Создать *объект* нужного вида можно несколькими способами. Например - вызвав контекстное *меню* группы *объектов* и выбрав в нем *пункт Добавить* (рис. 1.9).

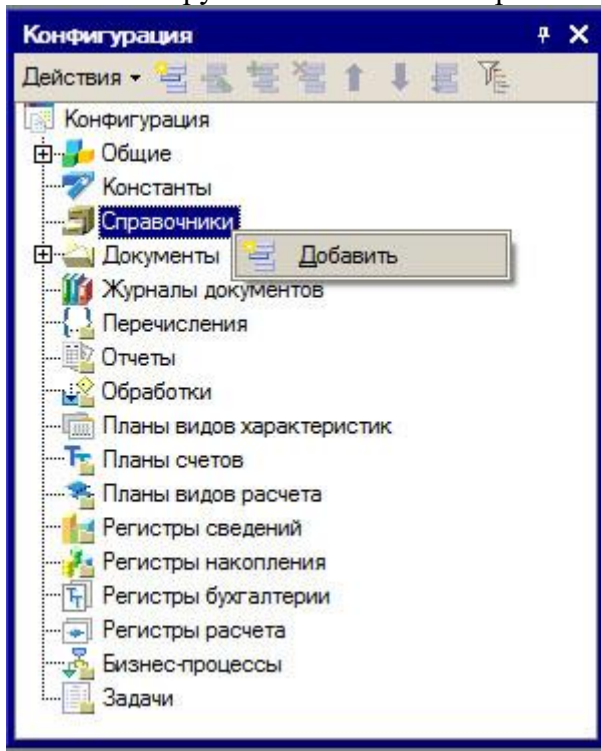


Рис. 1.9. Создаем новый объект

Обратите внимание на пиктограмму, которая сопровождает *пункт меню*. Такую же пиктограмму имеет кнопка *Добавить* на панели инструментов окна *дерева конфигурации*. Эта же команда продублирована в *меню Действия*.

Обратите внимание на то, что одно и то же действие в *Конфигураторе* можно выполнить различными способами.

Пока не будем выполнять описанную команду. В следующей лекции мы попрактикуемся в создании *объектов* и в других операциях с ними.

Выше мы использовали понятие "*объект*", однако, не поясняли его. Если вы знакомы с объектно-ориентированной *методологией программирования*, понятие "*объект*" должно быть вам знакомо. Если нет - давайте рассмотрим простой пример, который позволит понять сущность *объектов*. Представьте себе,

что *конфигурация* - это ноутбук. Ноутбук состоит из отдельных частей, связанных воедино, взаимодействующих друг с другом. Это - *монитор*, клавиатура, материнская *плата*, центральный *процессор*, оперативная *память* - *список* можно продолжать очень долго. Всё то, из чего состоит наш ноутбук, его детали - это *объекты*. Каждый *объект* обладает определенной функциональностью. Он может связываться каким-то образом с другими *объектами*, он имеет средства управления, может сообщать другим *объектам* о своем состоянии. В более широком смысле и ноутбук - это тоже *объект*. Им можно управлять, используя клавиатуру. Он может выводить сообщения с помощью монитора. Они составляют *интерфейс*, с помощью которого *пользователь* взаимодействует с ноутбуком.

Итак, *объект*, это "деталь" *конфигурации*. Существуют *объекты* различных видов - их *список* можно видеть в окне *дерева конфигурации*.

Сейчас рассмотрим некоторые другие не менее важные действия, которые выполняются в *Конфигураторе*.

1.4. Архивная копия информационной базы

Одна из важнейших сервисных операций, которую вам постоянно придется выполнять - это создание архивной копии *информационной базы*.

Архивные копии рабочих баз нужно делать достаточно регулярно - для того, чтобы обезопасить себя от потери информации. Если вы собираетесь выполнить какую-нибудь операцию, которая может повлиять на нормальную работу *конфигурации*, например - обновить *конфигурацию* - прежде чем ее выполнять, вы должны обязательно сделать архивную копию *информационной базы*.

Имейте в виду, что операция по архивированию *информационной базы* универсальна, как и многие другие *операции* в *Конфигураторе*. То есть - используя описанные команды вы сможете сделать архивную копию любой *конфигурации*, работающей на платформе 1С:Предприятие 8.

Сделать архивную копию *информационной базы* можно несколькими способами.

Первый из них заключается в обычном копировании или архивировании каталога с *информационной базой*.

Второй способ - воспользоваться инструментами архивирования, встроенными в *Конфигуратор*. А именно, если выполнить команду меню **Администрирование > Выгрузить информационную базу**, появится окно, которое содержит стандартный *запрос* о задании имени сохраняемого файла. В нашем случае это *файл* с расширением .DT. Он содержит архивную копию *информационной базы*. В случае повреждения или утери *информационной базы* вы можете восстановить ее из архивного файла командой **Администрирование > Загрузить информационную базу**.

Помните, что *информационная база* содержит не только *конфигурацию*, но и базу данных, которая формируется при работе пользователя с системой, то есть - данные, которые он вводит в систему в режиме 1С:Предприятие.

При загрузке *информационной базы* из файла вы получаете то ее состояние, которое она имела на момент выгрузки. Изменения, сделанные после этого, теряются. Предположим, мы выгрузили *информационную базу* и после этого создали новый *объект* системы, ввели какие-то данные в пользовательском режиме. Если после этих действий загрузить ранее выгруженную *информационную базу*, окажется, что сделанные изменения утрачены.

1.5. Основная конфигурация и конфигурация базы данных

Информационная база хранит две *конфигурации*. Одна из них называется **основной конфигурацией** или просто *конфигурацией*. Именно ее мы правим в ходе работы с *конфигуратором* и именно ее открываем командой **Конфигурация > Открыть конфигурацию**. Вторая *конфигурация* называется **конфигурацией базы данных**. Она используется в ходе работы пользователей. Редактировать *основную конфигурацию* можно в процессе работы пользователей с *информационной базой*. Для того, чтобы перенести изменения в *конфигурацию базы данных*, нужно, чтобы пользователи завершили работу с программой. Для того, чтобы открыть *конфигурацию базы данных*, выполните команду **Конфигурация > Конфигурация базы данных > Открыть конфигурацию БД**. Команды изменения *объектов* в окне *конфигурации базы данных* заблокированы.

Если, открыв *конфигурацию*, мы внесем в нее изменения, например, создадим новый *объект*, в заголовке окна *дерева конфигурации* можно будет наблюдать значок: "*" (признак модифицированности *конфигурации*) (рис. 1.10).

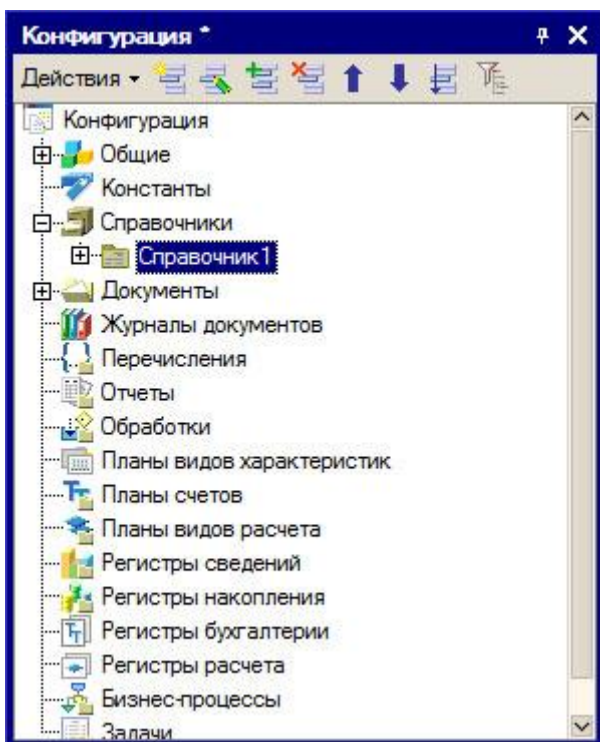


Рис. 1.10. Признак модифицированности конфигурации

Значок "*" означает, что внесенные изменения не сохранены в *основной конфигурации*. То есть он сигнализирует о том, что изменения, которые мы внесли на этапе конфигурирования, могут быть утеряны при, например, внезапном отключении питания.

Для того, чтобы сохранить *основную конфигурацию*, нужно выполнить команду меню **Конфигурация > Сохранить конфигурацию**.

После сохранения значок модифицированности *конфигурации* исчезает, но появляется значок отличия *конфигураций* "<!">" (рис. 1.11).

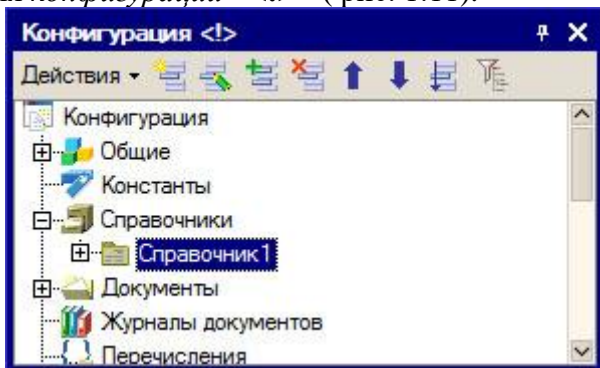


Рис. 1.11. Признак отличия конфигураций

Знак отличия *конфигураций* указывает на то, что изменения, внесенные в *основную конфигурацию*, пока не внесены в *конфигурацию базы данных*.

Для того, чтобы перенести изменения из *основной конфигурации* в *конфигурацию базы данных*, нужно выполнить команду **Конфигурация > Обновить конфигурацию базы данных**. Если изменения, внесенные в *основную конфигурацию* не были сохранены до выполнения этой команды - система, перед обновлением *конфигурации базы данных*, предложит сначала сохранить изменения.

При обновлении *конфигурации базы данных* система выводит окно с перечнем изменений, которые будут внесены в *конфигурацию базы данных* (рис. 1.12).

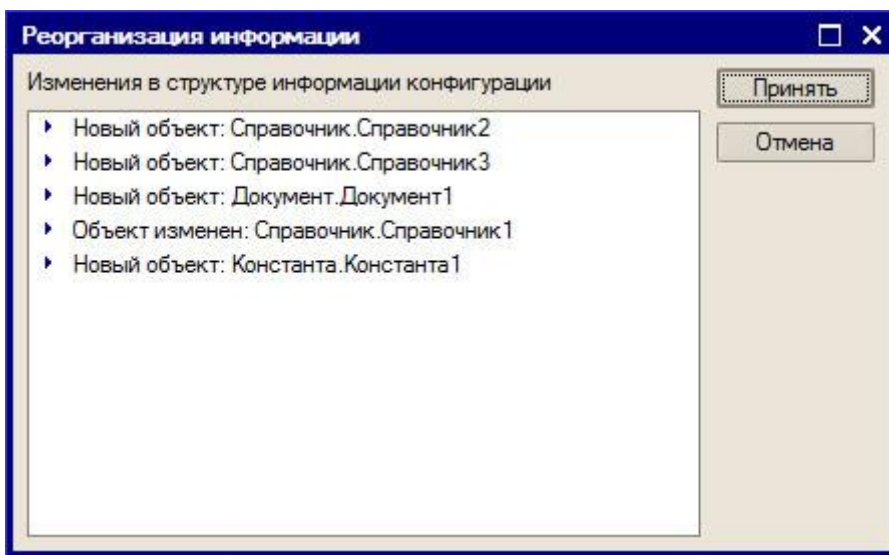


Рис. 1.12. Запрос о реорганизации информации

Если вы согласны с перечнем изменений, нажмите на кнопку **Принять**, в противном случае - на кнопку **Отмена**.

Признаки отличия *конфигураций* и модифицированности могут отображаться в заголовке окна *дерева конфигурации* одновременно. Если вы внесли изменения в *конфигурацию*, сохранили *основную конфигурацию*, не обновляя *конфигурацию базы данных*, а потом продолжили вносить изменения - оба значка отобразятся в заголовке окна.

Если вы внесли изменения в *основную конфигурацию*, сохранили их, но еще не обновляли *конфигурацию базы данных*, вы можете вернуться к *конфигурации базы данных*, отменив изменения, сделанные в *основной конфигурации*. То есть, фактически, заменить *основную конфигурацию* *конфигурацией базы данных*. Для этого нужно выполнить команду **Конфигурация > Конфигурация базы данных > Вернуться к конфигурации БД**.

Конфигурацию можно сохранять в *файл* и загружать из файла. Для сохранения *основной конфигурации* в *файл* выполните команду **Конфигурация > Сохранить конфигурацию в файл**. Для загрузки *конфигурации* из файла выполните команду **Конфигурация > Загрузить конфигурацию из файла**. Выгруженная *конфигурация* хранится в файле с расширением .CF.

В *файл* можно выгружать не только *основную конфигурацию*, но и *конфигурацию базы данных* - это можно сделать командой **Конфигурация > Конфигурация базы данных > Сохранить конфигурацию БД в файл**.

Обратите внимание на то, что выгружая *информационную базу в файл*, вы сохраняете в этом файле *информационную базу* целиком - то есть *основную конфигурацию*, *конфигурацию базы данных* и саму *базу данных*, которая содержит информацию, введенную пользователем в пользовательском режиме.

Сохраняя *конфигурацию в файл* вы сохраняете лишь *конфигурацию* - структуру данных, которая используется для описания возможностей, доступных пользователю в режиме 1С:Предприятие.

1.6. Варианты запуска 1С:Предприятия из конфигуратора

В процессе правки *конфигурации* в режиме *конфигуратора* бывают моменты, когда нужно проверить функциональность разрабатываемого решения в режиме 1С:Предприятие. *Конфигурацию* можно открыть в этом режиме прямо из *Конфигуратора*, воспользовавшись командой **Сервис > 1С:Предприятие** или соответствующей кнопкой на панели инструментов. При таком способе запуска *программа* будет работать точно так же, как она работала бы, если бы вы запустили ее в режиме 1С:Предприятие из стартового окна программы.

Еще один режим запуска, доступный из *Конфигуратора*, предназначен для отладочных целей. Для того, чтобы открыть *конфигурацию* в режиме отладки, выполните команду **Отладка > Начать отладку** или нажмите соответствующую кнопку.

Если до попытки запуска системы 1С:Предприятие из *Конфигуратора* в *конфигурацию* были внесены изменения, не отраженные в *конфигурации базы данных*, перед запуском система уточнит, хотите ли вы обновить *конфигурацию базы данных* (рис. 1.13).

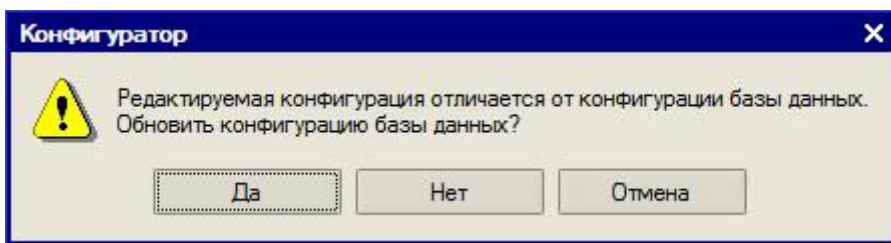


Рис. 1.13. Запрос об обновлении конфигурации базы данных

Если вы ответите **Нет** - программа запустится, но изменения из *основной конфигурации* не будут перенесены в *конфигурацию базы данных*. При ответе **Да** система сначала обновит *конфигурацию базы данных*, а потом запустится в режиме 1С:Предприятие.

1.7. Выводы

В данной лекции мы начали знакомство с *платформой 1С:Предприятие 8*. Сейчас вы должны уметь создавать новую *информационную базу*, запускать программу в режиме *конфигуратора* и 1С:Предприятие, открывать *информационную базу*, создавать *объекты*, архивировать *информационную базу* и сохранять *конфигурацию* в *файл*. Так же вы должны уметь запускать 1С:Предприятие из *Конфигуратора* для проверки функциональности разрабатываемого *прикладного решения* в различных режимах - в режиме отладки и в обычном режиме.

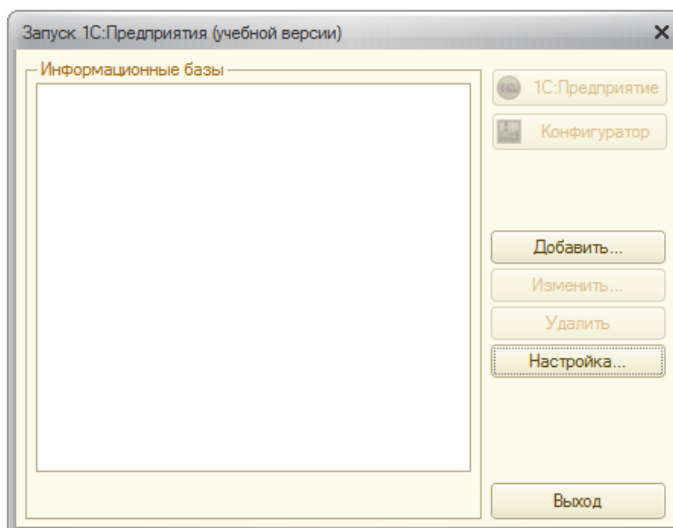
Лекция 7: Конфигуратор. Простейший кадровый учет

Сейчас мы с вами создадим мини-систему кадрового учета сотрудников на предприятии. Она позволит нам принимать и увольнять сотрудников, а также работать с кадровыми документами.

Каждая конфигурация должна храниться в отдельном каталоге (папке).

1. Создайте новую папку на любом диске нашего компьютера, например, "C:\Наши кадры".

2. Запустите 1С:Предприятие. Откроется список информационных баз. Если вы не создавали ни одной информационной базы и не добавляли существующие базы, то список будет пустым.

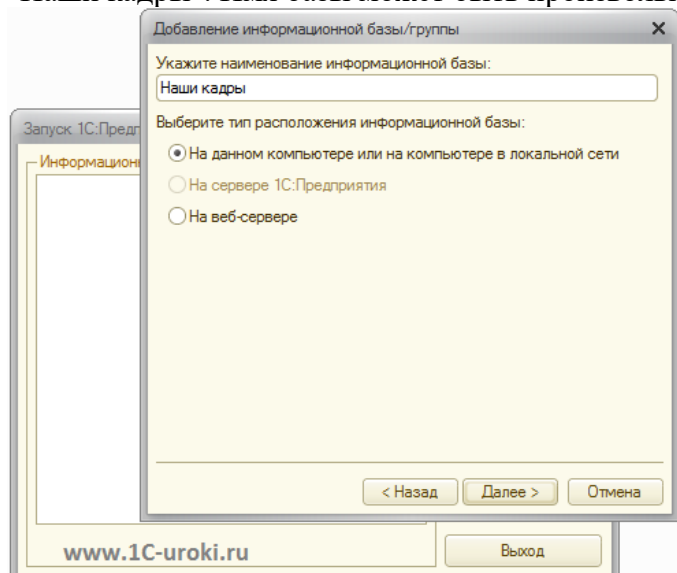


1С:Предприятие может работать с несколькими базами данных на одном компьютере. Например, в одной базе ведется кадровый учет, в другой базе - учет товаров, в третьей - бухгалтерский учет для Фирмы А, в четвертой - бухгалтерский учет для фирмы Б.

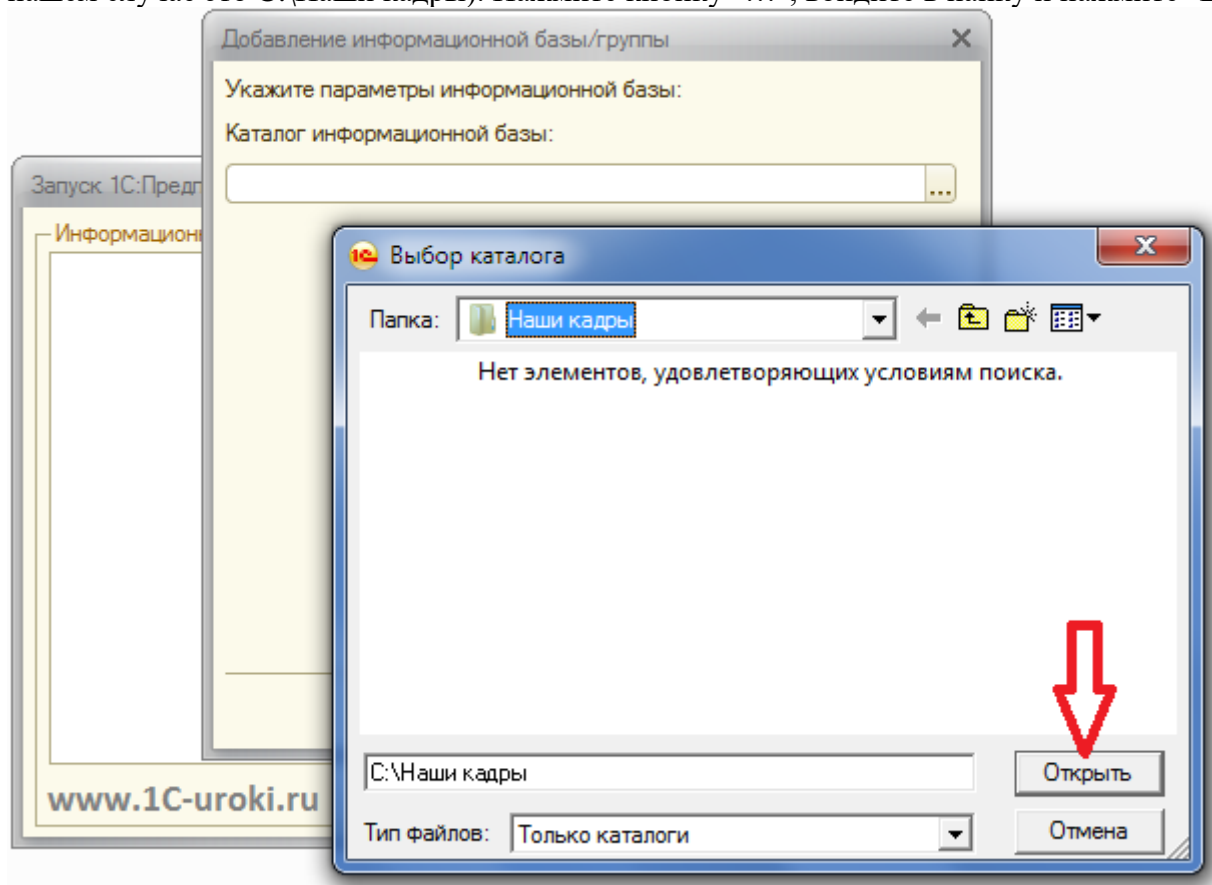
3. Нажмите кнопку "Добавить".

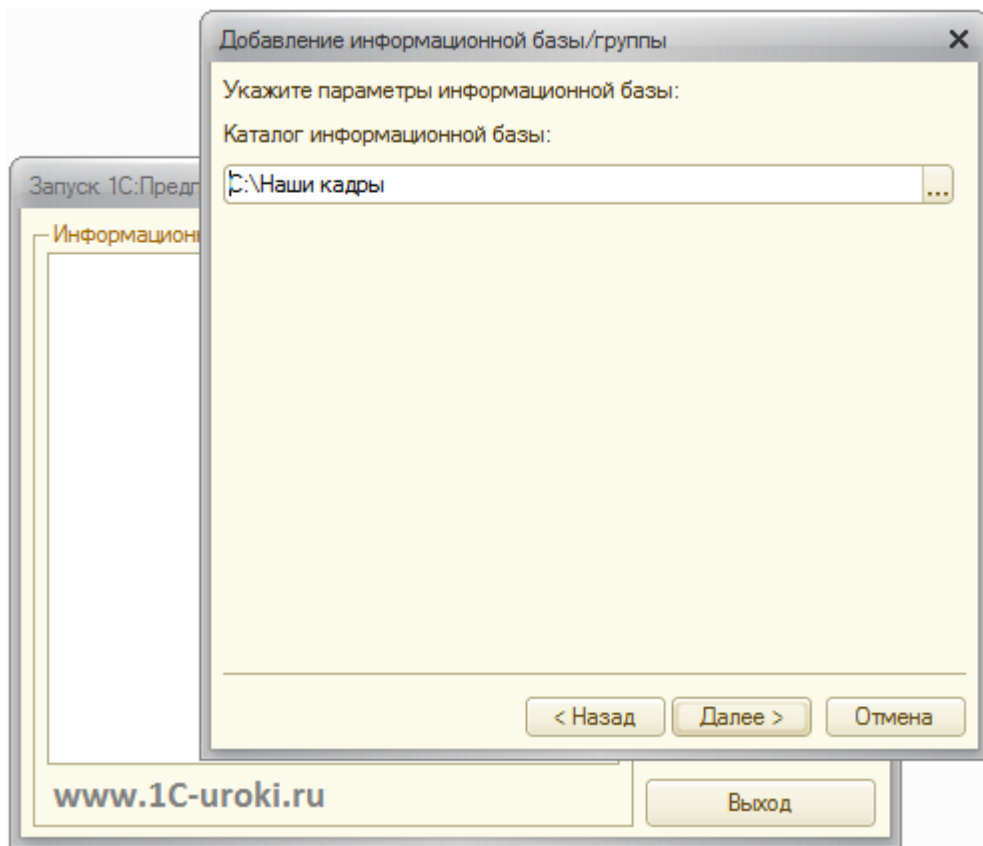
На первом шаге мастера выберите вариант "Добавление в список существующей информационной базы".

На втором шаге укажите имя информационной базы, как оно будет отображаться в списке баз, например "Наши кадры". Имя базы может быть произвольным, от него ничего не зависит.



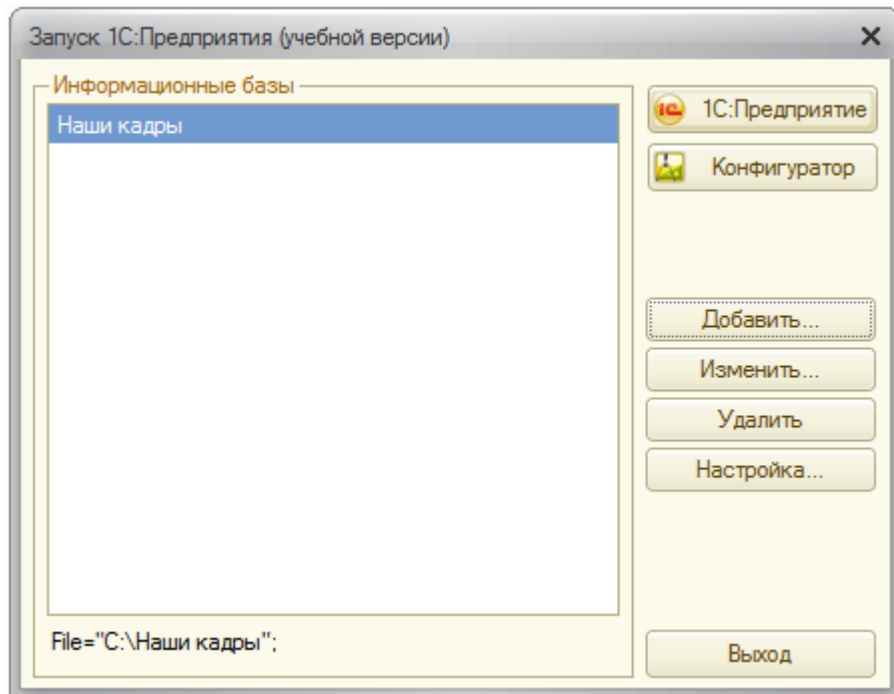
4. В следующем шаге укажите путь к папке, которую вы создали в 1 пункте нашего пошагового руководства (в нашем случае это C:\Наши кадры). Нажмите кнопку "...", войдите в папку и нажмите "Выбрать".





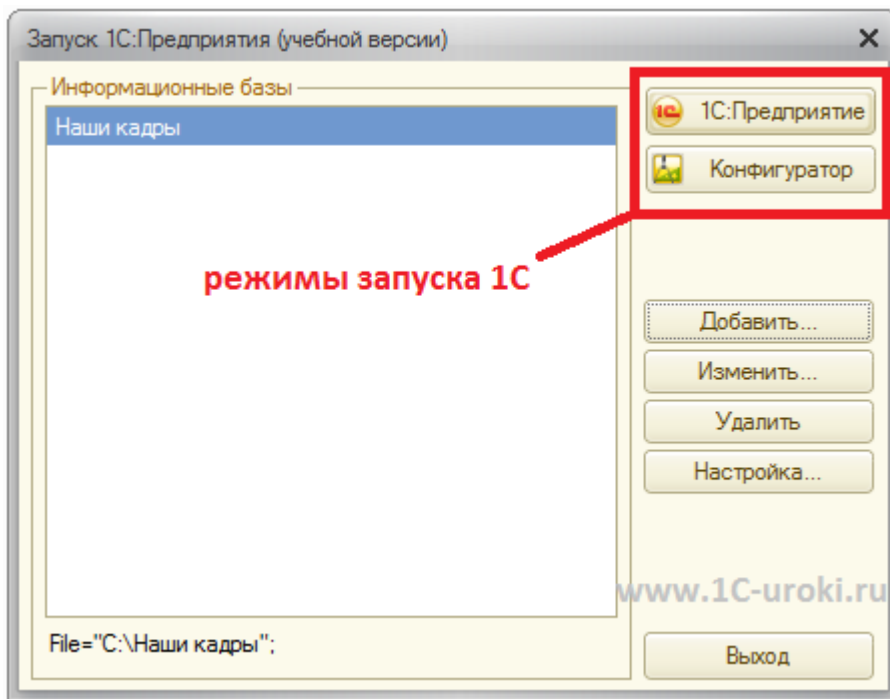
Каталог с информационной базой может располагаться на локальном жестком диске компьютера, например "C:\Наши кадры", а может на другом компьютере в локальной сети, например, "\\SERVER\KADR". При этом должны быть предоставлены полные права доступа к этой папке (чтение, запись и т.д.).

5. Нажмите кнопку "Далее". На завершающем шаге мастера оставьте все без изменений и нажмите кнопку «Готово». В результате в списке информационных баз появится новая строка с именем нашей информационной базы.



Для ее редактирования можно нажать кнопку Изменить. Например, информационную базу перенесли в другое место или вы хотите задать для нее другое название. Для удаления информационной базы из списка нужно нажать кнопку Удалить. При этом информационная база удаляется из списка, но не удаляется физически с диска.

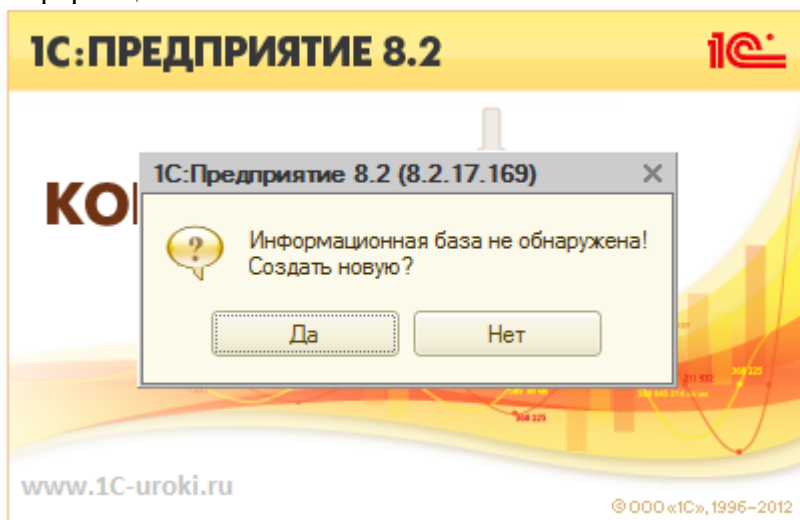
6. Выберем режим запуска "Конфигуратор". Для этого нажмем на кнопку «Конфигуратор».



1С может запускаться в двух режимах:

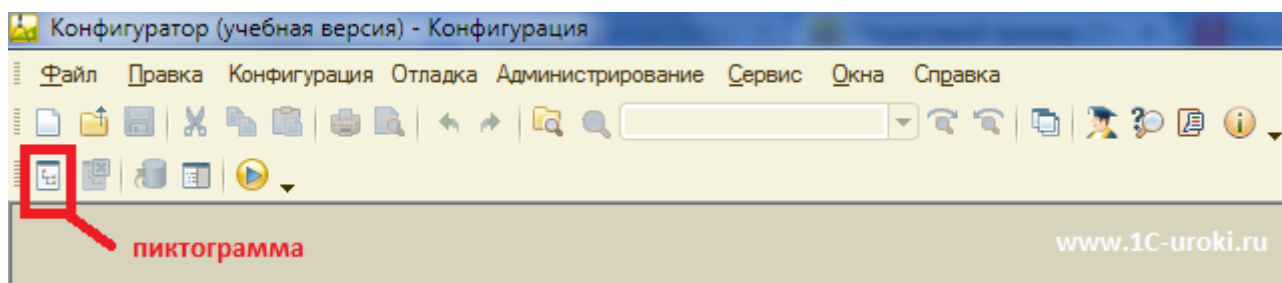
- **1С:Предприятие** - режим использования информационной базы. В этом режиме работают обычные пользователи, вводят данные, печатают отчеты и т.д. В этом режиме нельзя редактировать структуру конфигурации.
- **Конфигуратор** - режим конфигурирования (программирования). В этом режиме работают программисты, создают новые справочники, отчеты, пишут программные модули. В этом режиме нельзя вводить данные в информационную базу.

7. Так как в созданной папке пока ничего нет, то 1С запросит подтверждение на создание новой информационной базы.

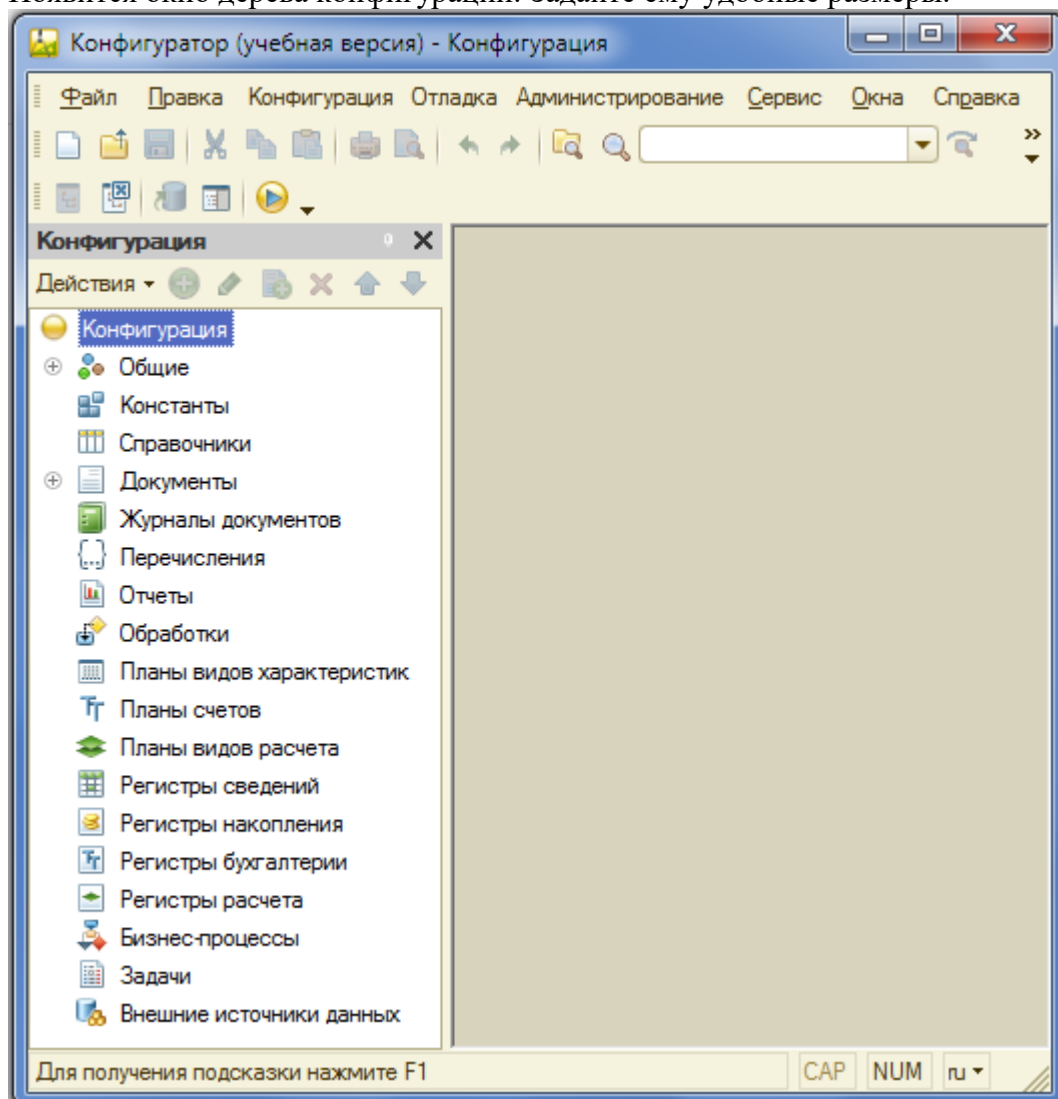


8. Нажмите "Да". Далее выбираем вариант "Создание информационной базы без конфигурации для разработки новой конфигурации или загрузки выгруженной ранее информационной базы" (2 вариант) и нажмите кнопку "Далее". На следующем шаге нажмите кнопку "Готово".

9. Запустится конфигуратор. Откройте дерево конфигурации, в котором мы будем работать очень часто. Для этого выберите в меню "Конфигурация" - "Открыть конфигурацию" или нажмите на соответствующую пиктограмму.




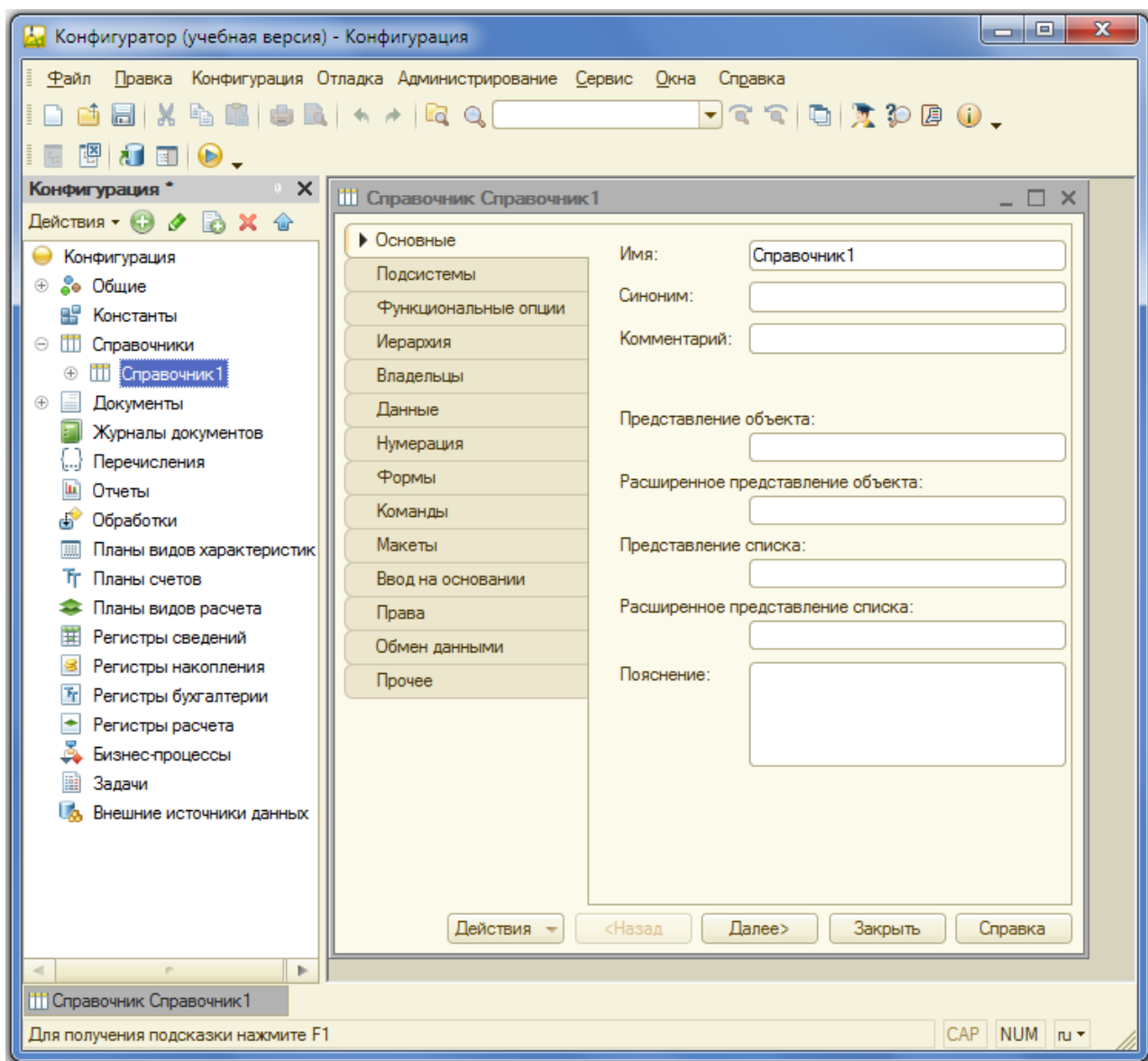
Появится окно дерева конфигурации. Задайте ему удобные размеры.



В этом окне представлены все объекты конфигурации, такие как справочники, документы, отчеты и т.д. Для раскрытия конкретной ветки нужно два щелкнуть на плюсики или два раза щелкнуть на ее названии.

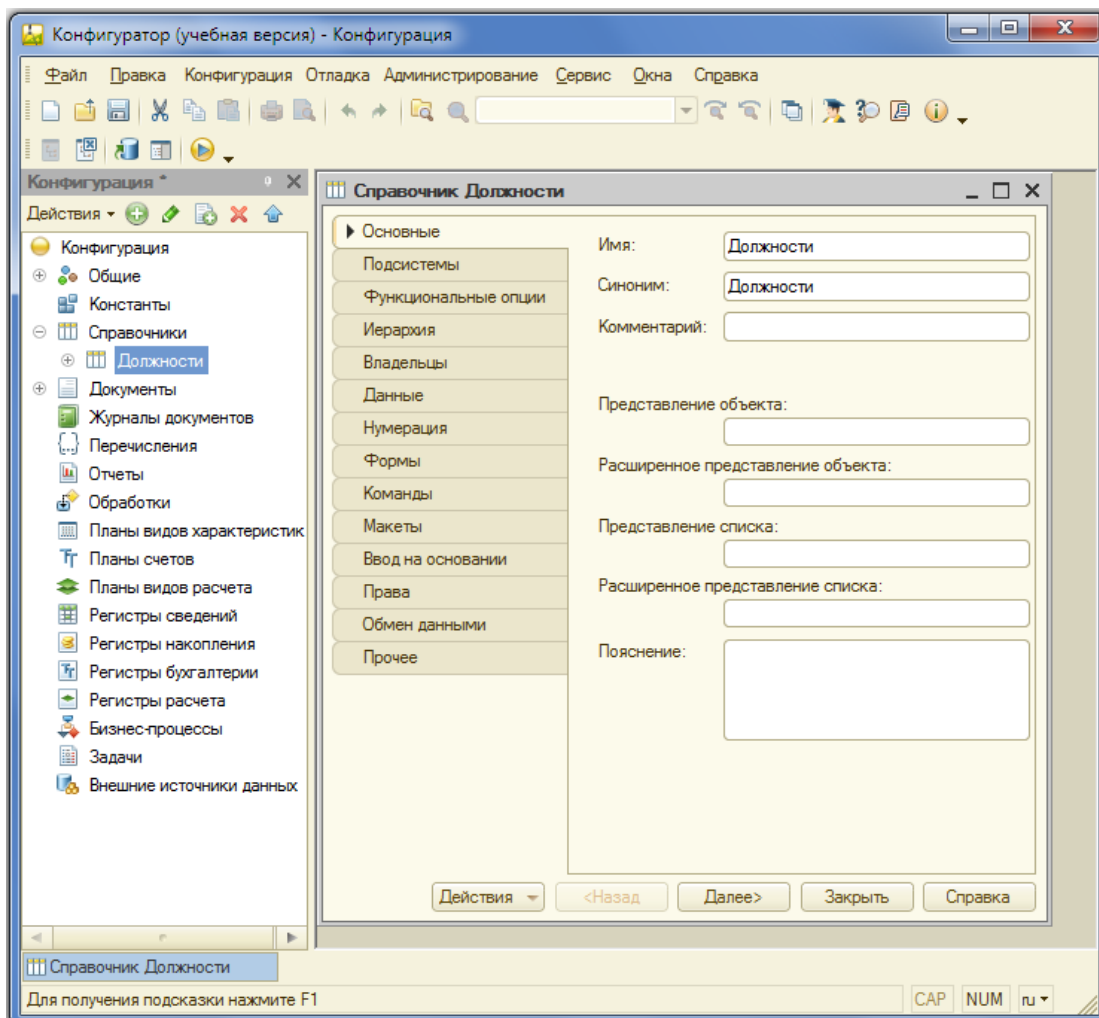
Создание объектов конфигурации

10. Установите курсор на строку "Справочники" и щелкните кнопку  (или щелкните правой кнопкой мыши и выберите Добавить). При этом откроется **Окно редактирования объекта конфигурации** (своеобразный конструктор для создания справочника).



Окно редактирования объекта конфигурации предназначены для быстрого создания новых объектов конфигурации. Последовательность ввода данных разработана таким образом, чтобы предыдущие данные могли служить основой для ввода последующих. Движение управляется кнопками в нижней части окна "Далее" и "Назад". На каждом шаге предлагается ввести группу логически связанных между собой данных. Перемещаться по шагам можно также щелкая мышкой по соответствующей вкладке (Основные, Подсистемы, Функциональные опции и т.д.).

11. На первом шаге (вкладке) задайте "Имя" нашего справочника (идентификатор), например, "Должности". После ввода Имени нажмите на клавиатуре клавишу "Enter", либо переместитесь мышкой на следующее поле. На основании "Имени" система автоматически создаст "Синоним".



Имя объекта (в нашем случае справочника) является важным и обязательным свойством любого объекта конфигурации. Именно оно будет использоваться при написании программного кода на языке 1С. Имя нельзя будет исправить просто так, поскольку придется также исправить все места в программе, где используется этот справочник.

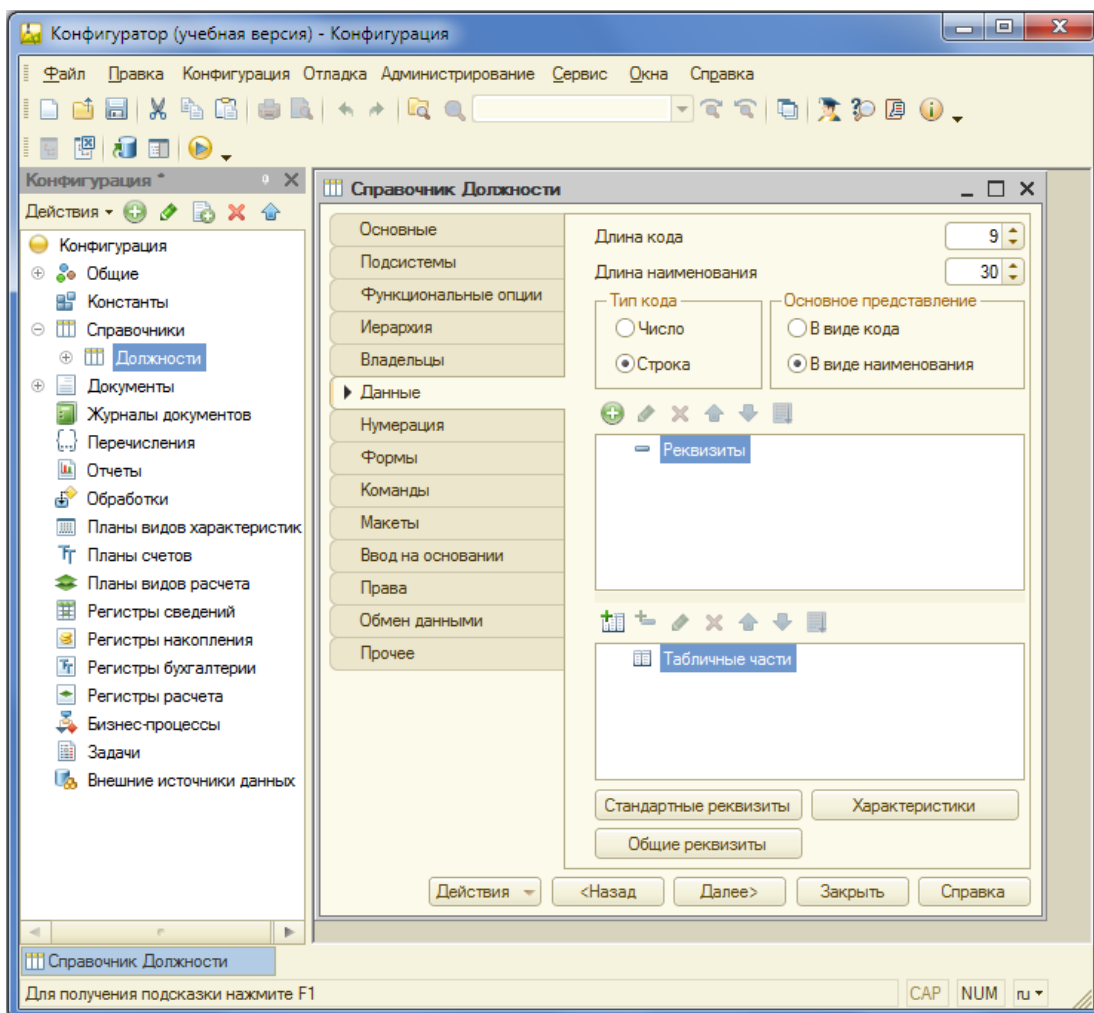
Относитесь к "Имени" справочника очень серьезно. Имя должно быть лаконичным, понятным, отражающим суть справочника, например: "Страны", "Сотрудники", "Товары" и т.д.

Имена объектов конфигурации, реквизитов, переменных, процедур и функций в 1С могут содержать буквы русского и английского алфавитов (строчные и прописные), символы подчеркивания и символы цифр. Имя не может начинаться с цифр.

Свойство "Синоним" также есть у любого объекта конфигурации. Оно предназначено для хранения "альтернативного имени" объекта конфигурации (в нашем случае справочника). Именно оно будет использоваться в элементах интерфейса нашей программы, то есть будет показано пользователю. Для «Синонима» нет никаких ограничений и его можно задать в удобочитаемом виде, например, «Перечень должностей» и т.д.

12. Перейдите с помощью кнопки "Далее" (в нижней части окна) или клика мышкой на вкладку "Данные".
 Задайте длину наименования 30 символов.

У каждого справочника есть уже заранее заданные (предопределенные) два поля "Код" и "Наименование". Вы можете редактировать длину наименования, длину кода, а также тип кода: "Число" или "Строка".



13. Закройте Окно редактирования справочника "Должности".

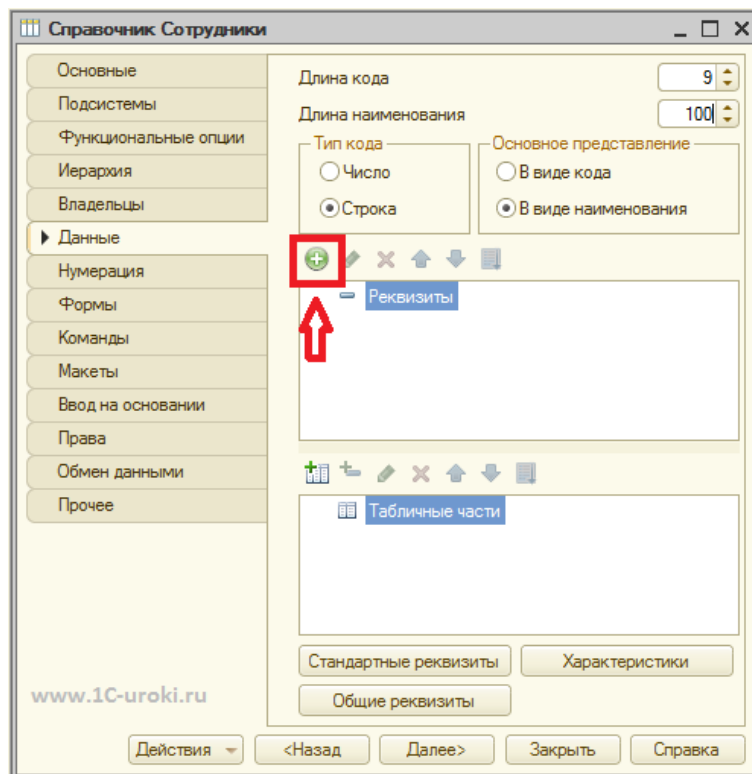
14. Создадим новый справочник "Сотрудники" (см. пункт 10).

Задайте имя справочника (идентификатор) "Сотрудники".

Задайте длину наименования 100 символов.

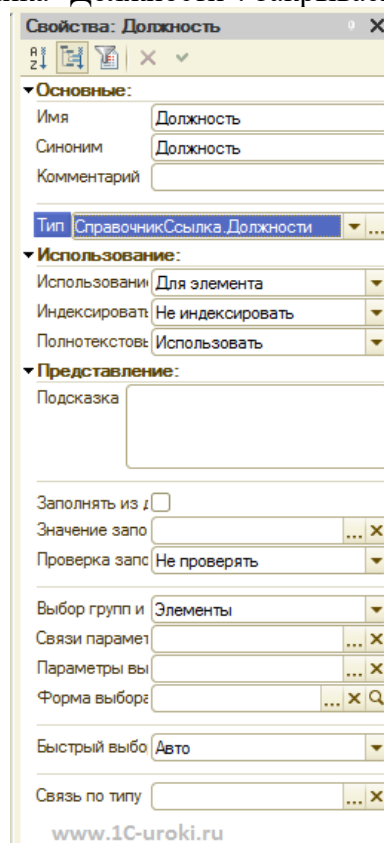
В наименовании у нас будет храниться ФИО сотрудника. Иногда бывают довольно длинные имена и фамилии, так что 100 символов в самый раз.

15. В этом справочнике кроме уже заданных полей (Код и Наименование) у нас будет еще несколько полей (реквизитов, атрибутов). Сейчас список дополнительных реквизитов (полей) у нас пустой. Нажмите кнопку "Добавить".



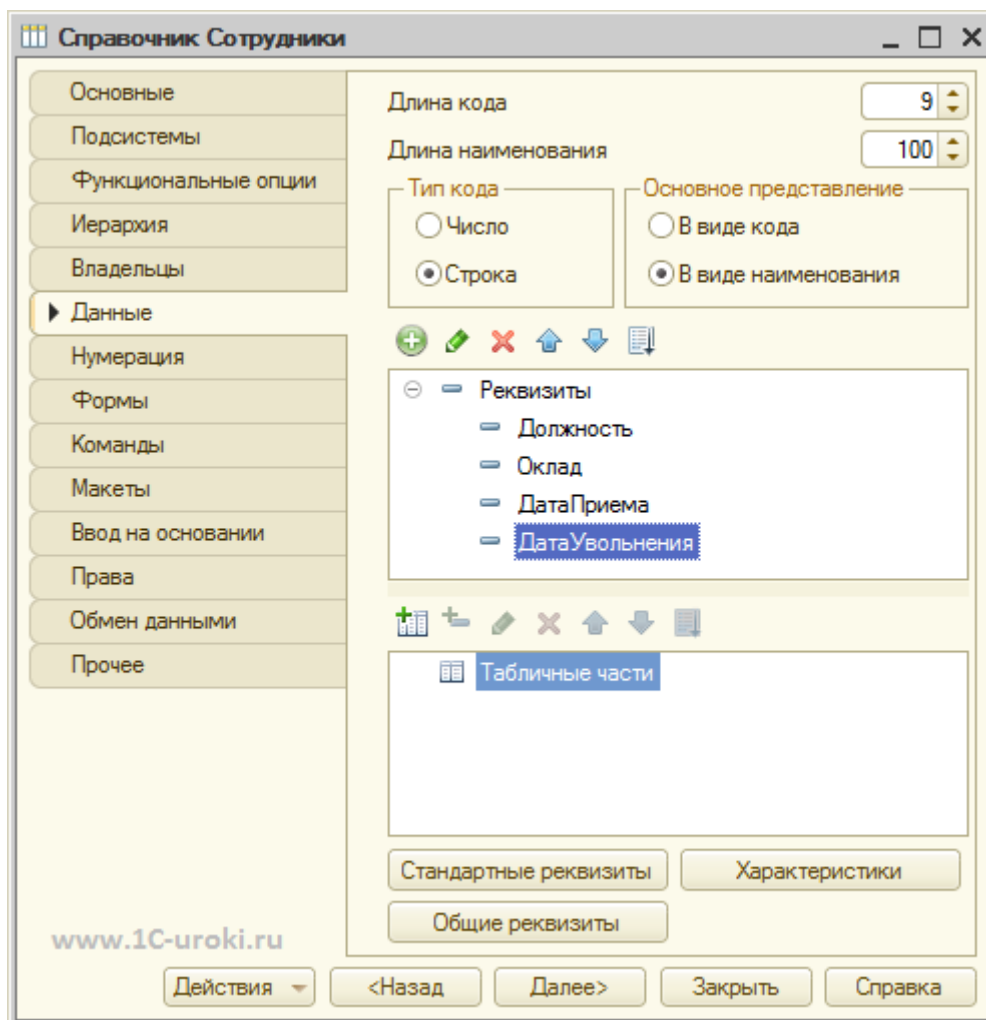
В результате откроется окно редактирования свойств реквизита (палитра свойств). Кстати, реквизит, атрибут и поле - это в принципе одно и то же. Просто в терминологии 1С принято использовать "реквизит".

16. Задайте в палитре свойств имя реквизита - "Должность". Укажите тип реквизита - "СправочникСсылка.Должности", который мы создали раньше. Это говорит о том, что значения данного реквизита будут выбираться из справочника "Должности". Закрываем окно палитры свойств.



17. Создайте реквизит "Оклад" (тип Число, Длина 10, Точность 2). У значения типа "Число" точность указывает на количество знаков после запятой (в нашем случае 2 знака).

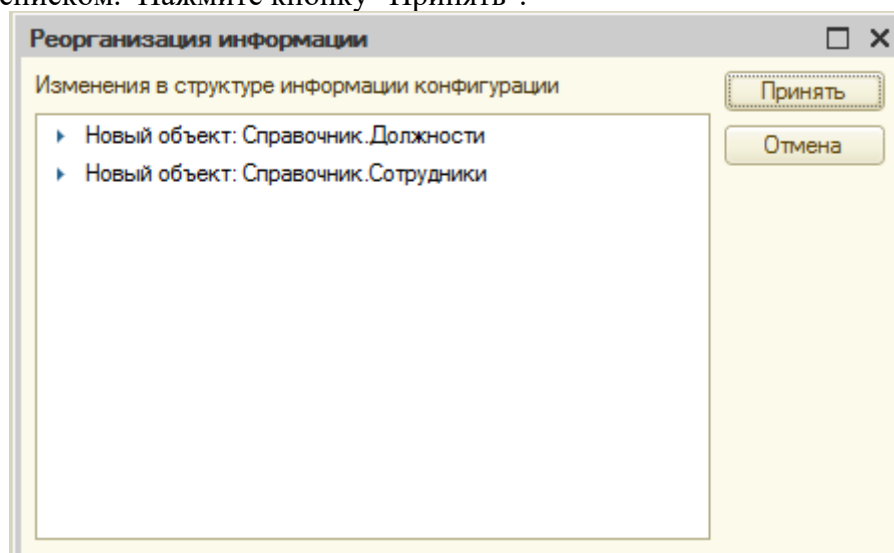
18. Создайте реквизиты ДатаПриема и ДатаУвольнения (тип Дата).




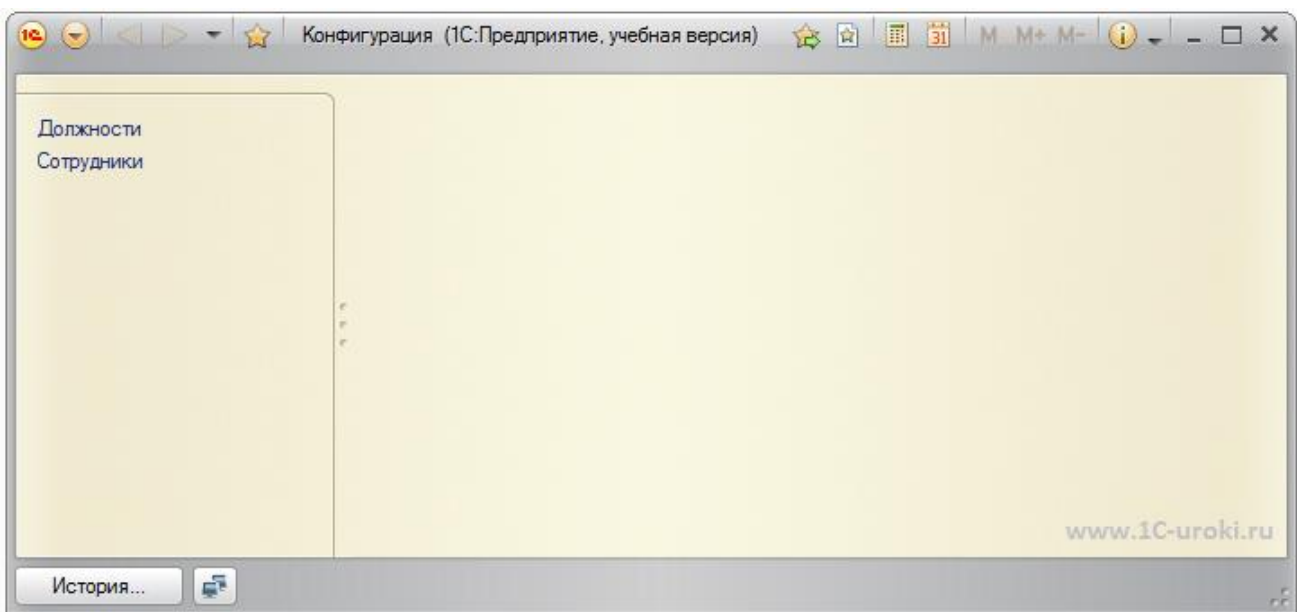
19. Теперь у нас есть два справочника и можно ввести некоторые данные.

Обновим конфигурацию базы данных (сохраним в конфигурацию все сделанные нами изменения). Сделать это можно с помощью меню программы (Конфигурация – Обновить конфигурацию базы данных), либо соответствующей пиктограммой на панели инструментов, либо клавишей F7.

Так как мы редактировали (изменяли) структуру информационной базы, то 1С проанализирует эти изменения и покажет окно со их списком. Нажмите кнопку "Принять".



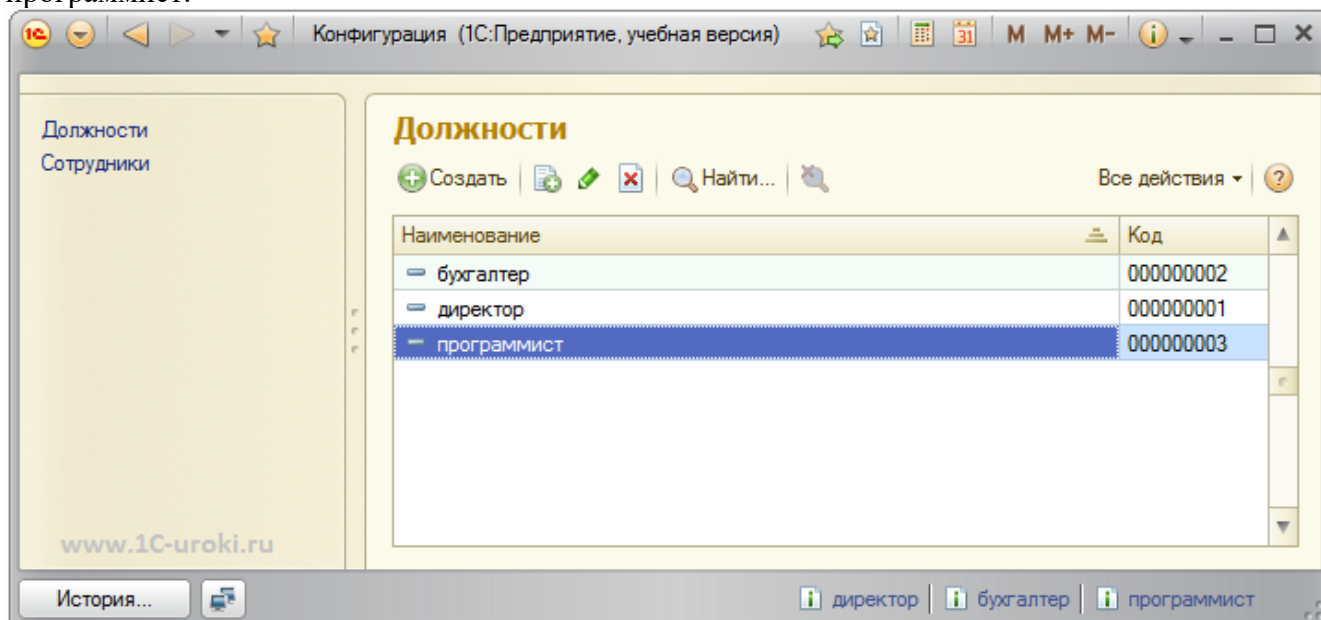
20. Запустите систему в режиме 1С:Предприятие. Это можно сделать прямо из Конфигуратора, нажав на панели инструментов на  или клавишу F5. Появится окно 1С:Предприятия.



И

21. Откройте справочник Должности. Для этого щелкните в панели навигации (в левой части окна приложения) по надписи (ссылке) Должности. Откроется окно справочника.

22. Введите несколько должностей, используя кнопку  или клавишу INS. Например, директор, бухгалтер, программист.

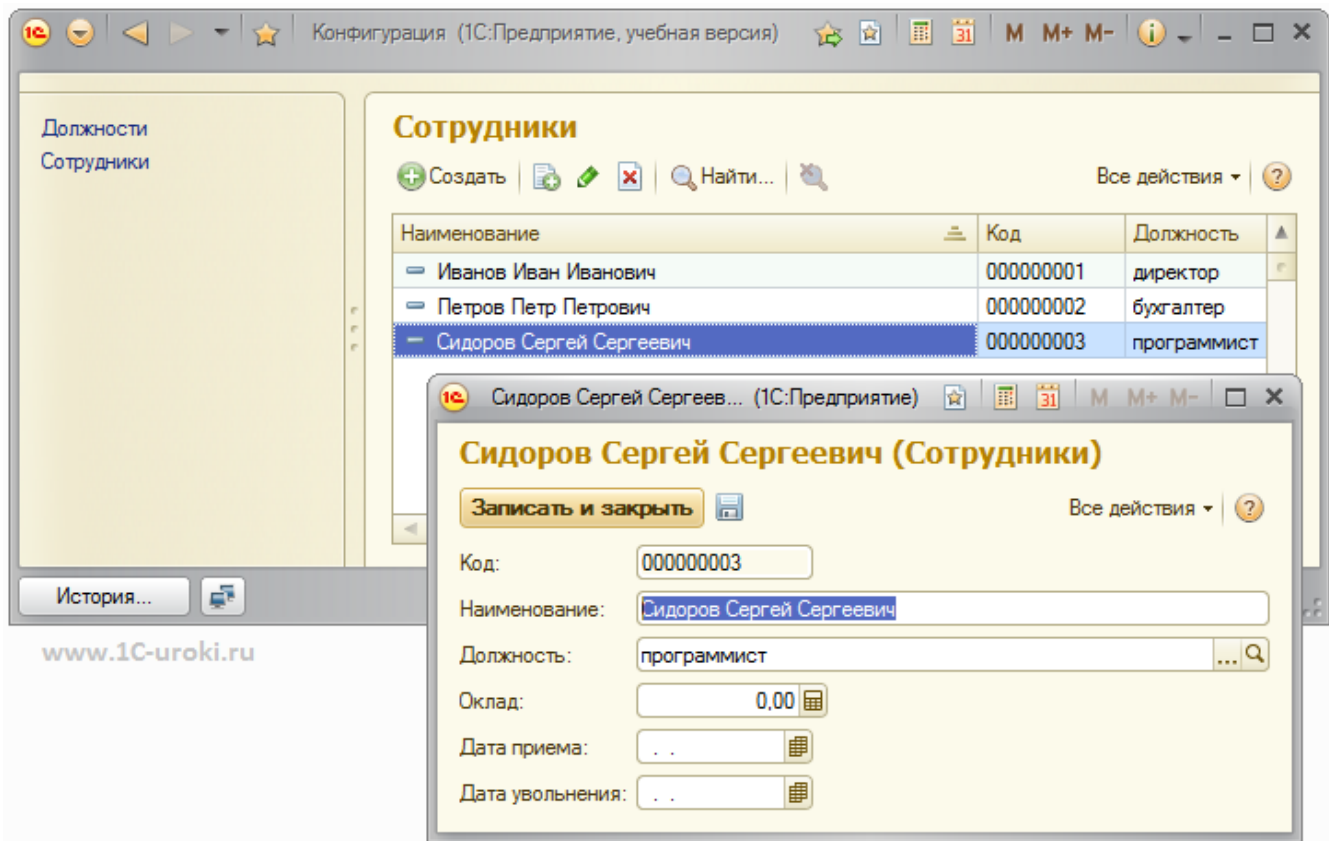


323. Откройте справочник Сотрудниками заметьте, что коды присваиваются автоматически, хотя их можно редактировать. При этом необходимо соблюдать уникальность кодов. Параметры уникальности можно настраивать в конфигураторе, например можно вообще отключить контроль уникальности кода.

По умолчанию установлена сортировка по Наименованию. Это позволяет набрать первые буквы названия должности, и курсор автоматически перейдет на нужную должность. Можно установить и другой тип сортировки: по коду или по реквизиту. Для этого достаточно щелкнуть левой кнопкой мышки по названию (заголовку) реквизита.

ики.

24. Заведите нескольких сотрудников. Например, Иванов Иван Иванович - директор, Петров Петр Петрович - бухгалтер, Сидоров Сергей Сергеевич - программист. Заполняем только реквизиты Наименование (ФИО) и Должность. Заметьте, что при указании должности открывается справочник Должности. Для сохранения нового элемента справочника (новой записи) нажмите «Записать и закрыть».



25. Закройте окно 1С:Предприятие и вернитесь в режим Конфигуратора.

В реальной жизни принятие сотрудника на работу производится кадровым документом "Приказ о приеме на работу". Это мы и реализуем во 2 части нашего Пошагового примера.

26. Теперь создайте новый документ "ПриемНаРаботу". (установите курсор на ветку Документы и щелкнув правой кнопкой мыши выберите Добавить). Задайте имя документа ПриемНаРаботу. Перейдите на закладку Данные.

27. Добавьте реквизит "ФИО", укажите ему тип значения Строка и длина 100 символов.

Каждый вид документа может иметь неограниченное количество реквизитов и табличной частей. Реквизиты относятся к документу в целом, например Номер, Дата и т.д. В табличной части содержатся строки. Например, один документ Накладная содержит информацию по нескольким товарам и каждый товар это отдельная строка в табличной части документа. Более того, в 1С предусмотрена ситуация, когда в документе нужно создавать несколько табличных частей.

Кстати, у любого документа уже есть два заранее заданных реквизита "Номер" и "Дата".

Далее нужно создать таким же образом, как и реквизит "ФИО", остальные реквизиты и укажите для каждого из них

- | | | | | | |
|---|----------|------------|---|-----|------------------------------|
| • | реквизит | Должность | - | тип | "СправочникСсылка.Должности" |
| • | реквизит | Оклад | - | тип | Число, длина 10, точность 2 |
| • | реквизит | ДатаПриема | - | тип | Дата |

Теперь в этот документ можно будет вводить данные о новом сотруднике (в режиме 1С:Предприятие). После ввода данных в новый документ пользователь нажмет кнопку "Провести и закрыть" и ваша программа должна добавить в справочник Сотрудники новый элемент (новую строку). При нажатии "Провести и закрыть" документ сначала будет записан, потом проведен, а потом закрыта его форма.

Проведение документа - очень важное понятие. Провести документ для пользователя означает пустить его в дело. При проведении документа совершаются изменения в информационной базе: добавляются данные в таблицы, увеличиваются или уменьшаются остатки товаров на складе, создаются новые бухгалтерские проводки и т.д.

Что конкретно произойдет при проведении документа, определяется программистом 1С. При проведении выполнится predefinedная процедура ОбработкаПроведения(). Именно в этой процедуре программист

пишет программный код на языке 1С, который совершает все изменения в системе. Сейчас нам нужно добраться до процедуры `ОбработкаПроведения()`. Она хранится в "Модуле объекта".

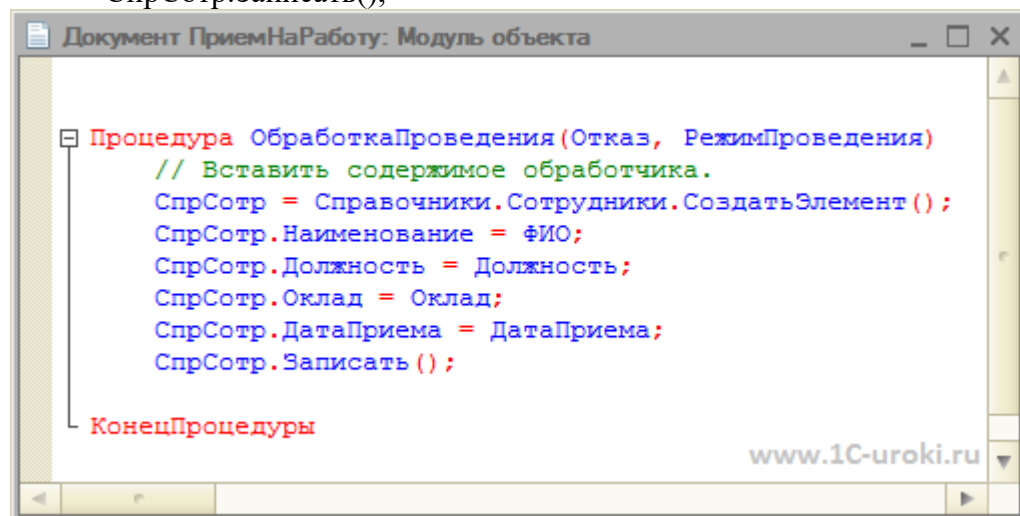
Для открытия модуля объекта перейдите на закладку "Прочее" и нажмите кнопку "Модуль объекта". Другой способ перейти в модуль объекта – выбрать в дереве конфигурации документ `ПриемНаРаботу` и кликнув правой кнопкой мышки выбрать "Открыть модуль объекта".

В результате откроется окно для редактирования текста программы.

28. На панели инструментов найдите пиктограмму "Процедуры и функции" и нажав на нее выберете "ОбработкаПроведения".

В результате в модуле объекта появилась процедура `ОбработкаПроведения`, но внутри нее пока ничего существенного. Напишите там следующий текст, пока не вникая в его смысл:

```
СпрСотр = Справочники.Сотрудники.СоздатьЭлемент();
СпрСотр.Наименование = ФИО;
СпрСотр.Должность = Должность;
СпрСотр.Оклад = Оклад;
СпрСотр.ДатаПриема = ДатаПриема;
СпрСотр.Записать();
```



Хотя, если присмотреться, то можно кое-что понять. Первая строка обращается к справочнику `Сотрудники` и создает новый элемент, пока пустой. Далее заполняются реквизиты справочника, данные берутся из документа и переносятся в справочник. И в конце происходит запись измененного элемента в справочник.

Кстати, часто начинающие специалисты 1С забывают вызвать `Записать()`. Если вы не вызовете метод `Записать()`, то все ваши изменения не сохранятся в базе.

Обратите внимание, что каждый оператор в языке 1С заканчивается символом ";" (точка с запятой). После слова **КонецПроцедуры** точки с запятой нет, потому что это не оператор, а операторная скобка (есть такое понятие в 1С). Точки с запятой нет также после слов **Цикл** и **Тогда**, но после слов **КонецЦикла** и **КонецЕсли** обязательно ставьте точку с запятой. В данной программе мы не использовали ни циклы, ни условия.

Также обратите внимание как оформляются комментарии.

// это комментарий

Комментарии не учитываются компилятором и предназначены для человека, которому нужно разобраться в программе. Комментариев не должно быть слишком много и слишком мало. Программа сама по себе должна быть понятна, без комментариев. Но в сложных местах в комментариях можно пояснить логику работы программы. Написание комментариев и понятных программ - признак хорошего тона. Будьте в этом деле профессионалами!

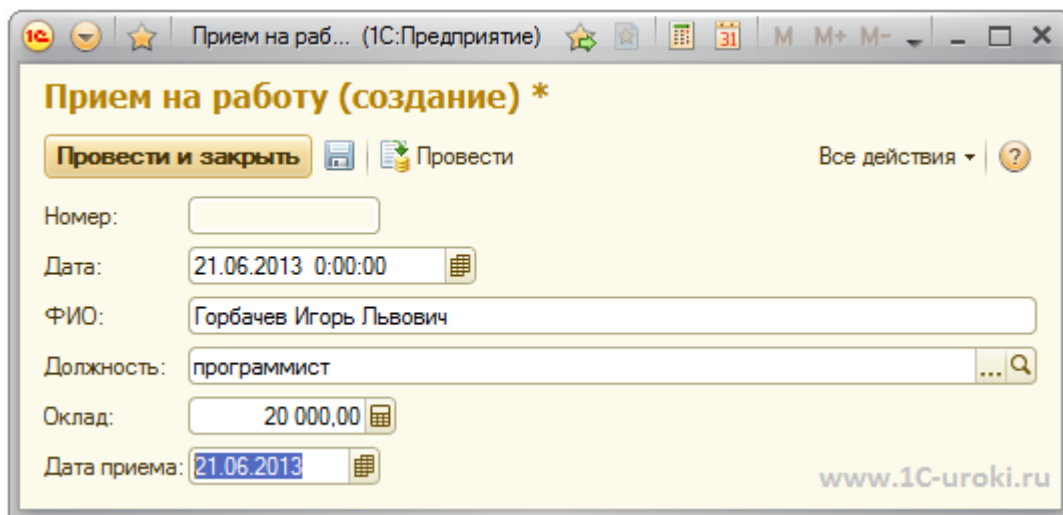
29. Закройте окно модуля и сохраните конфигурацию.

Теперь можно попробовать принять нового сотрудника по всей форме.

30. Запустите 1С в режиме 1С:Предприятие. В панели навигации выберете "Прием на работу". Перед нами откроется список документов "Прием на работу" и это список пока пустой.

31. Создайте новый документ. Откроется форма документа "Прием на работу" для ввода данных. Номер документа присвоится автоматически после его записи, дата документа - сегодняшнее число.

Введите все данные: ФИО сотрудника, укажите ему должность, оклад и дату приема на работу.

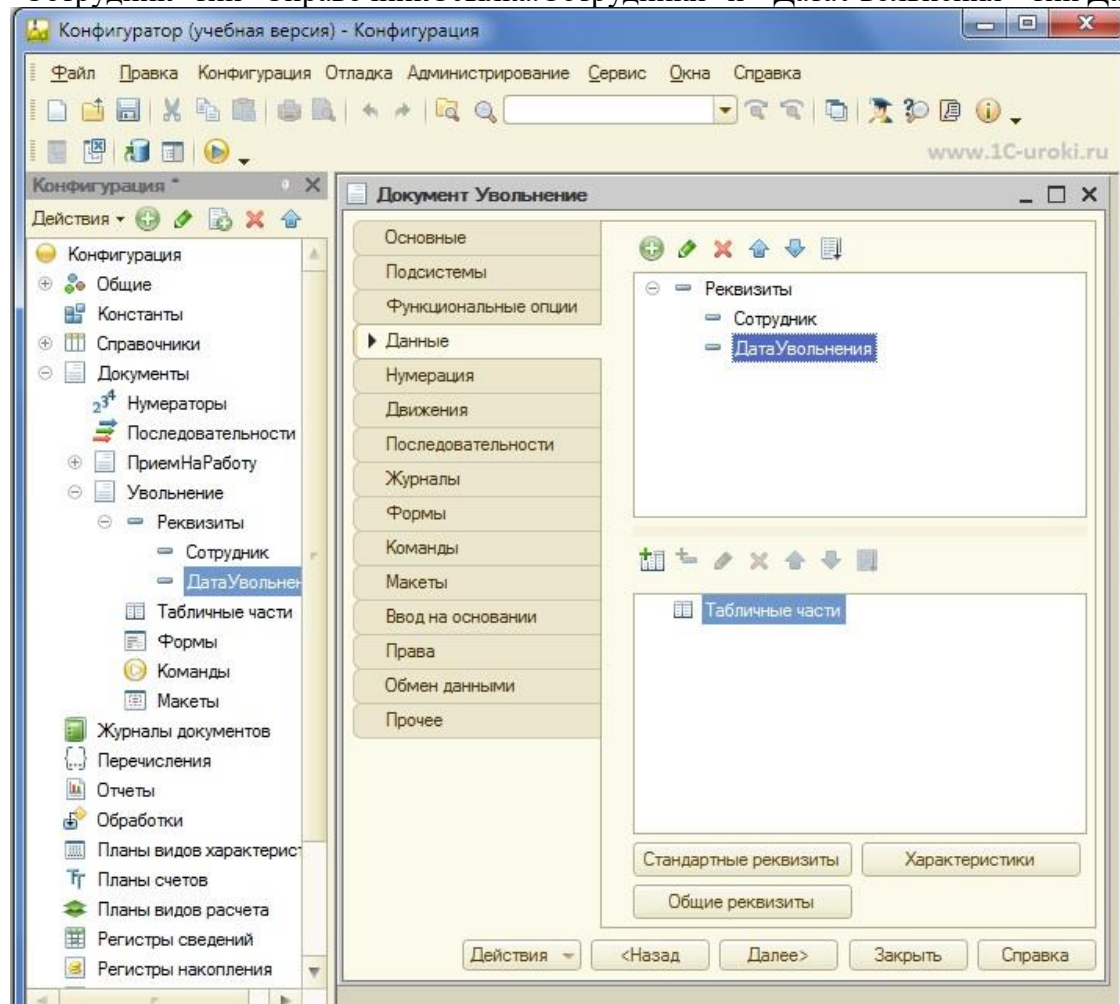


Нажмите кнопку "Провести и закрыть". Сначала документ будет записан, потом проведен, а затем его форма будет закрыта. При проведении сработает процедура **ОбработкаПроведения** и в справочник Сотрудники добавится новая строка. Проверьте это.

Если вдруг выскочит ошибка, значит, вы неправильно написали программу. Опять запускайте Конфигуратор и все перепроверяйте.

Теперь организуем правильное увольнение сотрудников с помощью кадрового приказа.

32. Создайте в Конфигураторе новый документ "Увольнение". Укажите, что он содержит реквизиты "Сотрудник" тип "СправочникСсылка.Сотрудники" и "ДатаУвольнения" тип Дата.



33. Напишите следующий текст в модуле документа в процедуре **ОбработкаПроведения()**:

```
СпрСотр=Сотрудник.ПолучитьОбъект();
СпрСотр.ДатаУвольнения=ДатаУвольнения;
СпрСотр.Записать();
```

Рассмотрим записанный программный код более подробно.

В первой строке нашего кода мы получаем из базы данных запись с сотрудником, который будет уволен. Метод ПолучитьОбъект() позволяет нам не только прочесть значения реквизитов данного объекта, но и изменять их. Следующая строка устанавливает ему дату увольнения. И последняя строка записывает сделанные изменения в справочник Сотрудники.

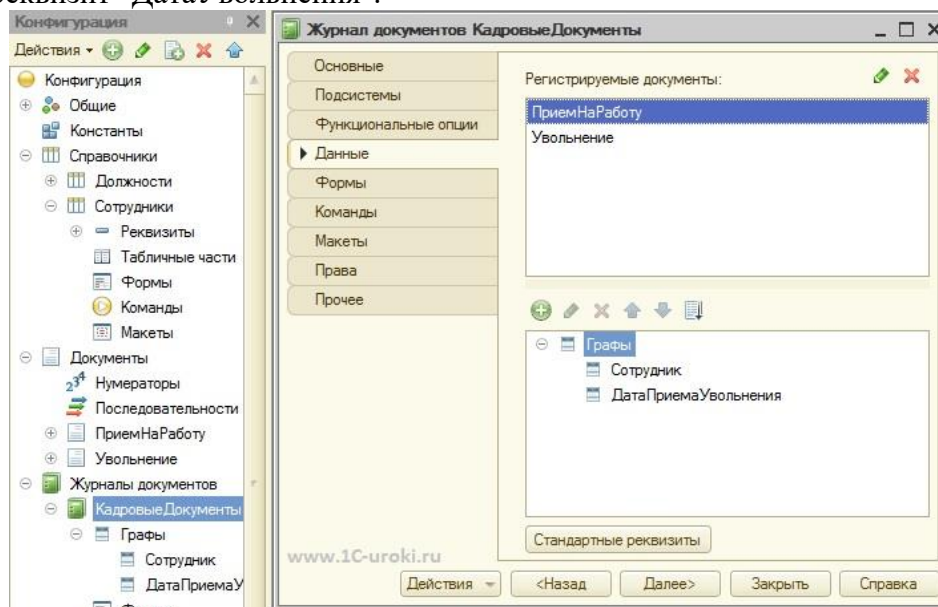
34. Создадим журнал "КадровыеДокументы". Для этого установите курсор на ветку "Журналы документов" в дереве конфигурации и щелкните "Добавить". Задайте имя журнала.

На закладке "Данные" укажите регистрируемые документы: "ПриемНаРаботу" и "Увольнение".

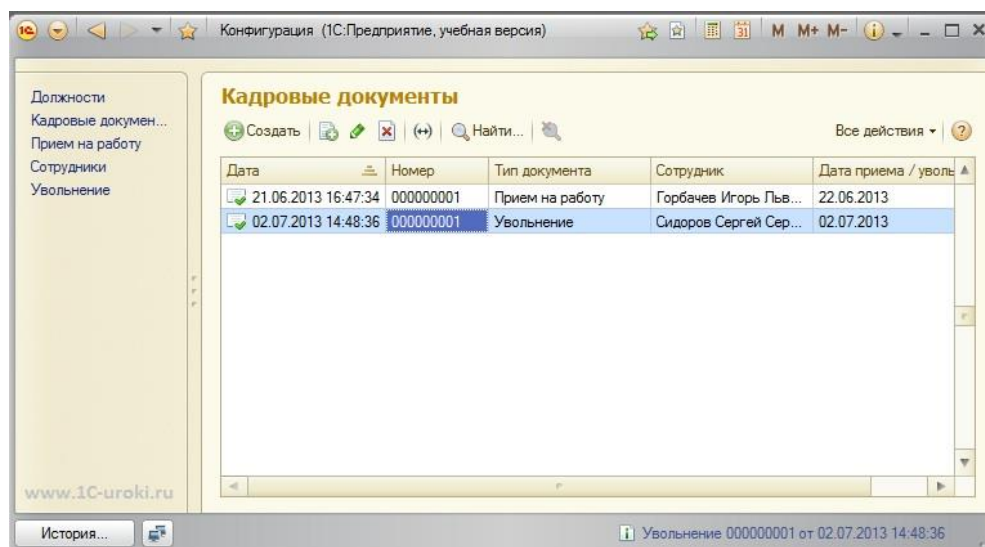
Создадим новые графы:

Сотрудник – для документа "ПриемНаРаботу" это реквизит "ФИО", а для документа "Увольнение" - это реквизит "Сотрудник";

ДатаПриемаУвольнения – для документа "ПриемНаРаботу" это реквизит "ДатаПриема", а для документа "Увольнение" - это реквизит "ДатаУвольнения".



35. Сохраните конфигурацию и запустите ее в режиме 1С:Предприятия. Проверьте работу документа "Увольнение", а затем откройте журнал "Кадровые документы". В результате он будет выглядеть примерно так:



На этом создание простейшей системы для кадрового учета завершено. Она позволяет нам принимать и увольнять сотрудников. Конечно, пока в системе реализованы лишь примитивные функциональные возможности. Уволенные сотрудники хранятся в одном списке с работающими. Реквизитов для хранения информации очень мало, нет печатных форм документов, нет кадрового перемещения, разнообразных отчетов - много пробелов в нашей системе. Но это же только простейшая кадровая система и она уже работает! И ее можно дорабатывать и развивать даже после начала ведения кадрового учета!

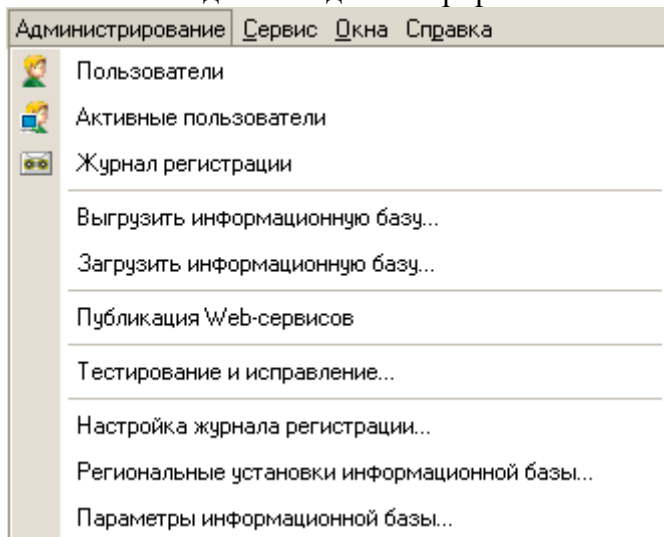
Для сравнения скажу, что написание такого же прикладного решения в другой среде или на другом языке программирования (Visual Basic или Delphi) займет в три или даже в пять раз больше времени и сил. Что бы там ни говорили недоброжелатели, но на 1С программы пишутся гораздо быстрее.

Лекция 8: Администрирование Базы данных

Система 1С:Предприятия 8 значительно облегчает администрирование информационной базы, предоставляя для этого все необходимое. К задачам по администрированию можно отнести следующие:

- установку и обновление системы,
- ведение списка пользователей,
- настройку прав доступа на основе механизма ролей,
- мониторинг действий пользователей и системных событий,
- резервное копирование,
- тестирование и исправление информационной базы,
- установку региональных настроек,
- обновление конфигураций.

Для каждой задачи система 1С:Предприятие 8 предоставляет удобные инструменты. Фактически, администрирование — это вторая профессия Конфигуратора, входящего в стандартную поставку. Основные инструменты администратора включены в подменю Администрирование:



Установка и обновление системы

Установка системы 1С:Предприятие 8 предельно упрощена и не вызывает никаких трудностей. Для запуска и работы не требуются какие-либо дополнительные компоненты или регистрация, достаточно просто наличия исполняемых файлов на локальном компьютере или где-нибудь в сети. Конфигурации (прикладные решения) также могут быть установлены с помощью инсталляторов (дистрибутивов), если это было предусмотрено разработчиками.

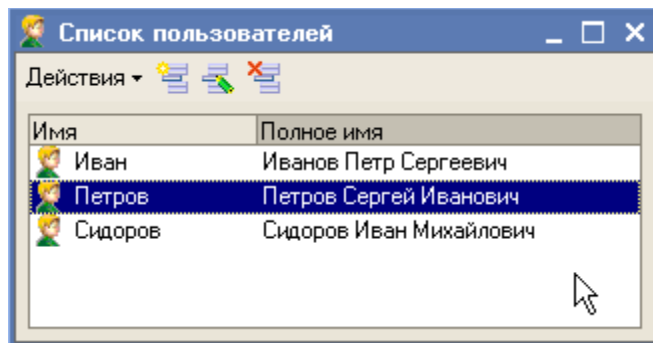
Администратор может воспользоваться "*административной установкой*", при которой исполняемые файлы системы 1С:Предприятие 8 хранятся на сервере. В этом случае при запуске программы у пользователя, если исполняемые файлы были обновлены на сервере, обновление системы на клиентских местах производится автоматически.

Несмотря на переход от двухуровневой архитектуры к трехуровневой, развертывание клиент-серверного варианта и его администрирование существенно упрощено. Например, создание базы данных производится непосредственно в процессе запуска конфигуратора (так же, как и для файлового варианта). В поставку системы входит дополнительная утилита для администрирования клиент-серверного варианта работы, позволяющая администратору управлять информационными базами и подключением пользователей.

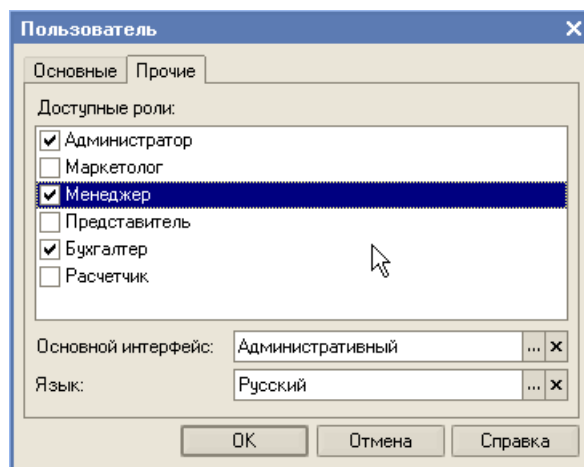
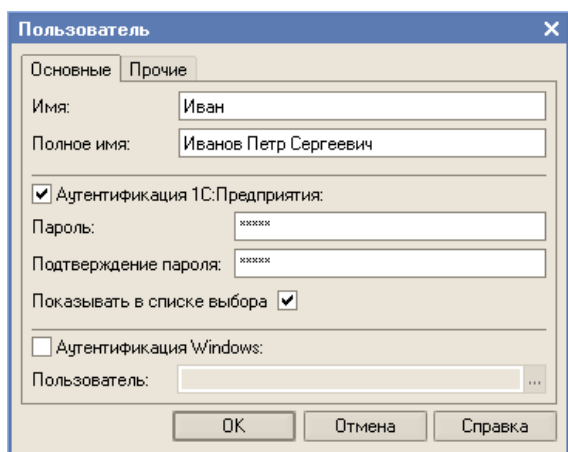


Список пользователей

Список пользователей, имеющих возможность работать с информационной базой, редактируется в следующем окне:



- Для каждого пользователя указывается его имя, полное имя, тип аутентификации, пароль, доступные роли, основной интерфейс и язык.
- В 1С:Предприятие 8 можно использовать аутентификацию пользователя средствами Windows. Для этого нужно установить соответствующий флажок и указать пользователя.
- Можно сделать так, чтобы пользователь не показывался в списке выбора при запуске системы.
- Впоследствии пользователь сам сможет поменять себе пароль через пункт меню "Сервис => Параметры пользователя".



Список активных пользователей

Всегда можно узнать, кто в данный момент подключен к информационной базе, для чего предназначен список активных пользователей. Он вызывается в обоих режимах работы 1С:Предприятия 8: Конфигуратор и Предприятие. В этом списке отображается имя пользователя, имя компьютера, режим запуска и другая информация. Оттуда же можно открыть журнал регистрации действий данного пользователя.



Журнал регистрации

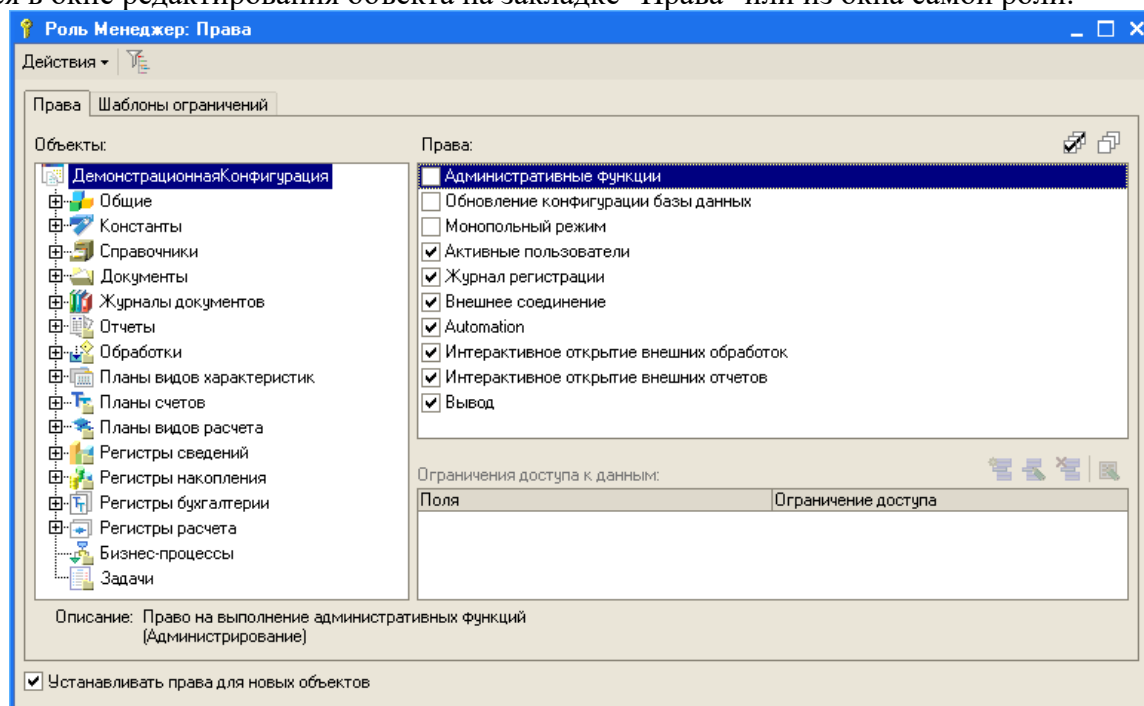
По долгу службы администратору часто требуется узнать, какие действия и события происходили в определенный момент времени. Для этого предназначен журнал регистрации, который по умолчанию отключен, так как его ведение требует некоторых дополнительных затрат. С помощью пункта "Настройка журнала регистрации" указывается важность событий, регистрируемых в журнале. К ним относятся ошибки, предупреждения, информационные сообщения и примечания.

При работе с журналом регистрации можно указать интересующий интервал времени, произвести отбор событий по определенному пользователю или компьютеру и другим параметрам. Журнал регистрации — очень удобное средство для мониторинга действий пользователей и восстановления последовательности событий, происходивших в определенный момент времени.

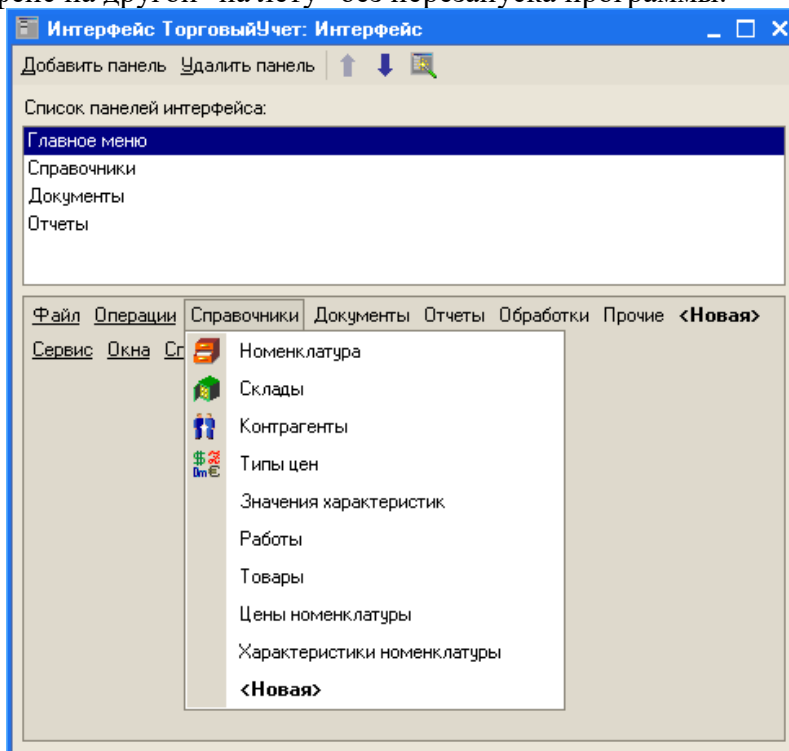


Роли и интерфейсы

Роль — это набор прав доступа к объектам конфигурации. У каждого пользователя может быть несколько доступных ему ролей, например, главному бухгалтеру доступны роли "Бухгалтер", "Расчетчик", "Учет материалов" и т.д. Редактирование прав доступа к объекту относится уже к изменению конфигурации и производится в окне редактирования объекта на закладке "Права" или из окна самой роли.



Часто на администратора возлагаются задачи по модификации интерфейса пользователя, например, добавление и удаление пунктов меню, их перегруппировка, редактирование панелей инструментов. Все эти функции доступны администратору, хотя интерфейс (как и роль) не входит в административную информацию, а является объектом конфигурации (ветвь "Общие"). Интерфейсы редактируются визуальными средствами в специальном редакторе интерфейсов. Изменения в интерфейсе для пользователя будут действительны только после обновления конфигурации базы данных, хотя пользователь может переключить свой интерфейс на другой "на лету" без перезапуска программы.





Резервное копирование

С помощью пункта "Выгрузить информационную базу..." можно выгрузить данные, конфигурацию и список пользователей в один файл. Следующий пункт "Загрузить информационную базу" позволяет восстановить информационную базу из файла. Этот механизм может также применяться для переноса баз между файловым и клиент-серверным вариантом.

В клиент-серверном варианте резервное копирование может производиться штатными средствами Microsoft SQL Server, при этом прерывать работу всех пользователей не требуется. В файловом варианте для резервного копирования можно скопировать файл информационной базы. Резервные копии рекомендуется архивировать, чтобы они занимали меньше места.

В целях резервного копирования или для других целей из информационной базы можно выгрузить только конфигурацию в cf-файл (аналог md-файла в версии 7.7). Т.е. в этот файл не попадут данные и сведения о пользователях. Другими словами, там будут только объекты конфигурации и программные модули.



Тестирование и исправление

Система 1С:Предприятие 8 имеет повышенную надежность, поддерживает целостность информационной базы при интенсивной работе. Но если с базой вдруг возникли неполадки, например, при неожиданном выключении электричества, то на этот случай в системе 1С:Предприятие 8 предусмотрены средства диагностики и исправления ошибок в базе данных. Для этого предназначена команда "Тестирование и исправление..." меню Администрирование. В появившемся окне можно указать, какие необходимо выполнить проверки и операции, требуется ли только тестирование или тестирование с исправлением ошибок, а также режим исправления. В абсолютном большинстве случаев неполадки устраняются системой автоматически и можно успешно продолжать работу.

Тестирование и исправление информационной базы

Проверки и режимы :

- Реиндексация таблиц информационной базы
- Проверка логической целостности информационной базы
- Проверка ссылочной целостности информационной базы
- Пересчет итогов
- Сжатие таблиц информационной базы
- Реструктуризация таблиц информационной базы

Только тестирование
 Тестирование и исправление

При наличии ссылок на несуществующие объекты :

- Создавать объекты
- Очищать ссылки
- Не изменять

При частичной потере данных объектов :

- Создавать объекты
- Удалять объект
- Не изменять

Продолжить прерванное ранее тестирование

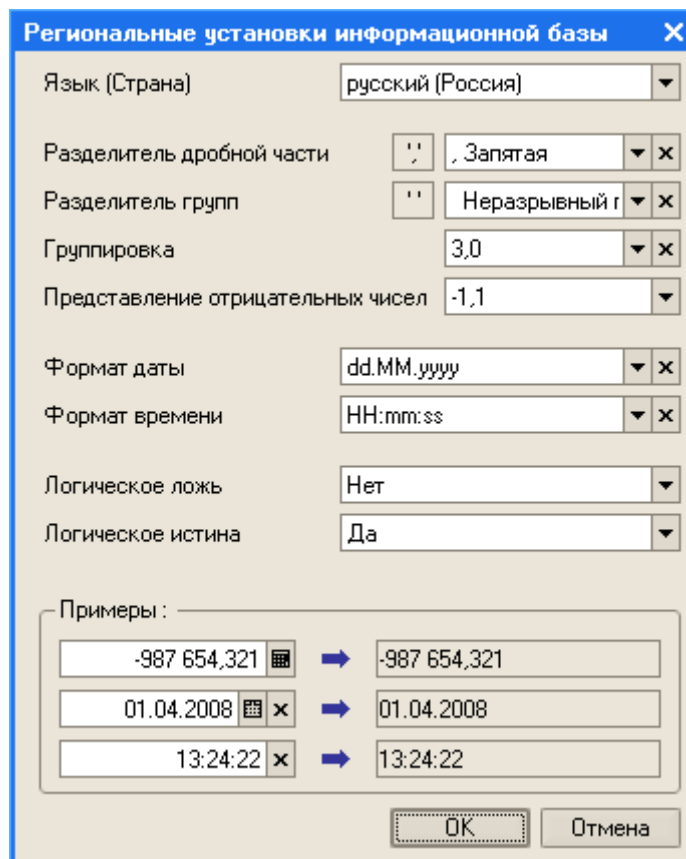
Прервать выполнение проверки через часов мин.

Выполнить
Закреть
Справка



Региональные установки

При необходимости администратор 1С:Предприятия 8 может настроить региональные установки информационной базы, к которым относятся язык, разделитель дробной части числа и разделитель групп разрядов, формат даты и времени, представление логических значений и другие параметры.



Обновление конфигураций

В современном динамичном мире вполне естественно, что многие конфигурации регулярно меняются. Изменения могут быть связаны с учетом изменений законодательства, добавлением новых функциональных возможностей или внесением исправлений. Особенно это касается типовых или тиражных конфигураций. Система 1С:Предприятие 8 содержит встроенный механизм поставки и поддержки конфигураций как для типовых, так и индивидуальных конфигураций.

Разработчики конфигураций при выпуске новых релизов подготавливают файлы поставки. Различают полную поставку и поставку обновлений. Последний вид поставки предназначен для обновления имеющейся конфигурации и содержит только изменения.

Администратор или пользователь может легко произвести обновление своей конфигурации. Чтобы его собственные изменения не были потеряны, производится детальное сравнение и объединение конфигурации пользователя и конфигурации поставщика.



Резюме

- Система 1С:Предприятие 8 содержит удобные инструменты для администратора информационной базы, включая список пользователей, систему настройки прав (ролей) и интерфейсов, журнал регистрации.
- Установка (обновление) системы и конфигураций производится простыми средствами. Достаточно легко осуществляется настройка клиент-серверного варианта.
- Как и в версии 7.7, система 1С:Предприятие 8 имеет средства тестирования и исправления ошибок в информационной базе. В абсолютном большинстве случаев неполадки устраняются системой автоматически и можно успешно продолжать работу.
- Информационная база может быть выгружена в файл, куда выгружаются данные, конфигурация и сведения о пользователях. В клиент-серверном варианте можно использовать штатные средства Microsoft SQL Server. В файловом варианте можно просто скопировать файл информационной базы.

Лекция 9: Синтаксис языка 1С

Встроенный язык системы 1С:Предприятие 7.7 предназначен для описания алгоритмов функционирования прикладной задачи. Синтаксис языка напоминает смесь Паскаля и Бэйсика, причем на русском языке.

Язык 1С поддерживает структурное программирование и вся программа состоит из набора модулей, содержащих исходный код - операторы языка и комментарии.

Программные модули

Программные модули в конфигурации не являются самостоятельными программами, поскольку являются частью всей конфигурации задачи. Программный модуль – это «контейнер» для размещения текстов процедур и функций, вызываемых системой во время исполнения в определенные моменты времени.

Место размещения конкретного программного модуля предоставляется конфигуратором в тех точках конфигурации задачи, которые требуют описания специфических алгоритмов функционирования.

Виды программных модулей:

- Глобальный модуль.
- Модуль формы списка справочника
- Модуль формы группы
- Модуль формы элемента справочника
- Модуль формы документа
- Модуль документа
- Модуль формы журнала документов
- Модуль формы списка счетов
- Модуль формы счета
- Модуль формы журнала операций
- Модуль формы операции
- Модуль формы журнала проводок
- Модуль формы отчета
- Модуль формы обработки

Каждый отдельный модуль воспринимается системой как единое целое, поэтому все процедуры и функции программного модуля выполняются в одном контексте.

Структура программного модуля

Программный модуль 1С состоит из следующих разделов:

- Раздел определения переменных
- Раздел процедур и функций
- Раздел основной программы

Разделы не требуют явного описания с использованием каких-либо ключевых слов (в отличие от, например, языка Паскаль), однако нарушение структуры модуля приведет к ошибке компиляции.

Контекст выполнения программного модуля

Глобальный контекст:

- значения системных атрибутов, системные процедуры и функции, например, функция ТекущееВремя() возвращает текущее системное время, а функция ТекущаяДата() – текущую дату на компьютере.
- значения заданных в конфигураторе констант, перечислений.
- переменные и процедуры глобального программного модуля с ключевым словом «Экспорт».

Локальный контекст образуется тем конкретным местом конфигурации задачи, для которого использован этот конкретный модуль.

Формат операторов

Общий синтаксис операторов языка 1С:Предприятие:

[~метка:]Оператор [(Параметры)][ДобавочноеКлючевоеСлово];

Идентификаторы

Именем переменной, процедуры или функции может быть любая последовательность букв, цифр и знаков подчеркивания, начинающаяся с буквы или с подчеркивания. Регистр букв не учитывается. Вновь создаваемые имена не должны совпадать с уже существующими именами, доступными на момент выполнения модуля.

Типизация переменных в языке не жесткая, т.е. тип переменной определяется ее значением. Чтобы определить, является ли слово зарезервированным, нужно воспользоваться поиском в синтакс-помощнике: установить курсор на проверяемое слово и одновременно нажать клавиши CTRL+F1. Таким же способом можно узнать синтаксис системной процедуры или функции: количество и назначение параметров, значения по умолчанию.

Пример правильных идентификаторов в 1С:

```
// Определение переменных Перем ЭтоПеременная; Перем ЭтоВтораяПеременная; // Процедуры и функции
Процедура ЭтоПроцедура() // текст процедуры КонецПроцедуры Функция ЭтоФункция() // текст функции
КонецФункции // Раздел основной программы ЭтоПременная = "123";
Процедуры и функции программного модуля
Синтаксис объявления процедуры:
Процедура <Имя_проц>([Знач] <Парам1> [=<ДефЗнач>] , ...) [Экспорт] // Объявления локальных
переменных; // Операторы; ... [Возврат;] // Операторы; КонецПроцедуры
```

После ключевого слова «КонецПроцедуры» точка с запятой не ставится. Ключевое слово «Знач» означает, что переменные передаются по значению. Ключевое слово «Экспорт» имеет смысл только в глобальном модуле и означает, что процедура будет доступна в любом программном модуле. Функция, в отличие от процедуры должна возвращать значение. Для передачи локального контекста используется ключевое слово «Контекст» в качестве параметра.

Пример. Допустим, у нас есть несколько различных документов, причем у всех есть реквизит "Склад". В каждом программном модуле формы редактирования документа нужна процедура выбора склада. В глобальном программном модуле напомним процедуру (листинг 36):

Листинг 36. Пример описания процедуры

```
Процедура УстСклада(Конт) Экспорт // создаем объект типа "справочник" Скл =
СоздатьОбъект("Справочник.Склады"); // вызываем диалог для выбора элемента Справочника
Скл.Выбрать("Выберите Склад", ""); Если Скл.Выбран() = 1 Тогда // если в диалоге
элемент Справочника был выбран, // то присваиваем его значение реквизиту Документа, // который
доступен по локальному контексту Конт.Склад = Скл.ТекущийЭлемент(); КонецЕсли; КонецПроцедуры
```

Здесь переменная Конт является формальным параметром, которому при вызове процедуры будет присвоен локальный контекст. В данном примере обращение к реквизитам и методам локального контекста происходит «через точку» после идентификатора Конт, например:

```
Конт.Склад = Скл.ТекущийЭлемент();
```

Теперь в любом программном модуле конфигурации (в данном примере в любом Модуле формы документа) для выбора склада можно вызвать процедуру, передав ей свой локальный контекст в качестве параметра:

```
УстСклад(Контекст);
```

С примерами программных модулей можно ознакомиться в типовой конфигурации «Бухгалтерский учет».

Типы данных

В системе поддерживаются базовые и агрегатные типы данных. К базовым типам относятся числа, строки и даты. Агрегатные типы данных – это специализированные типы данных, предназначенные для работы с объектами 1С:Предприятия. Агрегатными типами данных называются следующие типы данных:

- Константа – средство работы с постоянными (или условно постоянными) значениями. В константах хранится информация, которая не изменяется или изменяется достаточно редко. Например, название организации, почтовый адрес. Перечень констант, доступный в конкретной конфигурации, их названия и тип определяются в конфигураторе.
- Справочник – средство для ведения списков однородных элементов данных. Помимо наименования элементов данных, списки могут содержать различную дополнительную информацию. Физическим аналогом справочника может являться картотека. Каждая карточка – это элемент справочника, а сведения, заносимые в карточку, являются реквизитами справочника. Перечень справочников, доступных в конкретной конфигурации, их названия и реквизиты определяются в конфигураторе.

- Перечисление – средство работы с элементами данных, список возможных значений которых жестко задан (например, для перечисления «ФормаОплаты» можно задать возможные значения: «Нал», «Безнал»). В отличие от справочников, списки значений в перечислении задаются в процессе их создания в конфигураторе и при выполнении задачи не могут быть изменены. Состав перечислений, доступных в конкретной конфигурации, их названия и допустимые значения определяются в конфигураторе.
 - Документ – средство для ввода первичной информации о совершаемых хозяйственных операциях. Перечень документов, доступных в конкретной конфигурации, их названия, реквизиты и другие свойства определяются в конфигураторе.
 - Запрос – средство для выполнения обращения к документам, регистрам, документам, справочникам и журналам расчетов с целью получения сводной информации при формировании выходных отчетов. В программных модулях допускается создавать произвольное число объектов типа Запрос при помощи вызова системной функции СоздатьОбъект.
 - Текст – средство работы с текстовыми документами. В программных модулях допускается создавать произвольное число объектов типа Текст при помощи вызова системной функции СоздатьОбъект.
 - Таблица – средство работы с таблицами (отчетами). В программных модулях допускается создавать произвольное число объектов типа Таблица, при помощи вызова системной функции СоздатьОбъект.
 - СписокЗначений – средство для создания списка значений каких-либо данных и возможности в дальнейшем сортировать и выбирать нужные значения из списка. При добавлении в диалоговых формах полей типа «Список» или «Поле со списком», система автоматически создает объекты СписокЗначений, доступ к которым в языке возможен по идентификатору поля. В программных модулях допускается создавать произвольное число объектов типа СписокЗначений при помощи вызова системной функции СоздатьОбъект.
 - ТаблицаЗначений — средство для создания списка значений каких-либо данных и возможности в дальнейшем сортировать и выбирать нужные значения из списка. При добавлении в диалоговых формах полей типа «Список» или «Поле со списком», система автоматически создает объекты СписокЗначений, доступ к которым в языке возможен по идентификатору поля. В программных модулях допускается создавать произвольное число объектов типа СписокЗначений при помощи вызова системной функции СоздатьОбъект.
 - Картинка – средство для работы с графическими файлами. При добавлении в диалоговых формах и в таблицах полей типа «Картинка», система автоматически создает объекты Картинка, доступ к которым в языке возможен по идентификатору поля. В программных модулях допускается создавать произвольное число объектов типа Картинка при помощи вызова системной функции СоздатьОбъект.
 - Периодический — средство для работы с периодическими реквизитами справочников и периодическими константами. В программных модулях допускается создавать произвольное число объектов типа Периодический при помощи вызова системной функции СоздатьОбъект.
 - ФС – средство для работы с дисковыми файлами непосредственно из встроенного языка системы 1С:Предприятие. В программных модулях допускается создавать произвольное число объектов типа ФС при помощи вызова системной функции СоздатьОбъект. Кроме того, в глобальном контексте по умолчанию существует один уже созданный объект этого типа с именем ФС (имя объекта совпадает с названием агрегатного типа данных).
 - XBase – средство для работы с файлами баз данных DBF формата непосредственно из встроенного языка системы 1С:Предприятие. В программных модулях допускается создавать произвольное число объектов типа XBase при помощи вызова системной функции СоздатьОбъект.
- Следующие типы данных доступны только при наличии компоненты «Бухгалтерский учет»:**
- ПланСчетов является служебным типом данных. Он предназначен для идентификации Плана счетов, созданного в метаданных. В основном он используется для передачи в качестве параметра различным процедурам и функциям компоненты «Бухгалтерский учет» и для выбора плана счетов в формах. Тип значения «ПланСчетов» не поддерживает никаких данных в информационной базе, а список возможных значений этого типа данных определен планами счетов созданными в конфигурации. Значения типа «ПланСчетов» могут выступать как реквизиты диалога формы, как реквизиты документов, справочников и т. д. Для получения значения такого типа данных используется глобальный атрибут ПланыСчетов, который имеет в свою очередь набор атрибутов типа «ПланСчетов»

соответствующих имеющимся в конфигурации планам счетов. Кроме того, глобальный атрибут ПланыСчетов имеет методы для обхода всех существующих планов счетов.

- ВидСубконто является служебным типом данных. Он предназначен для идентификации Вида субконто, созданного в метаданных. В основном он используется для передачи в качестве параметра различным процедурам и функциям компоненты «Бухгалтерский учет» и для выбора вида субконто в формах. Список возможных значений этого типа данных определен видами субконто, созданными в конфигурации. Значения типа «ВидСубконто» могут выступать как реквизиты диалога формы, как реквизиты документов, справочников и т. д. Для получения значения такого типа используется глобальный атрибут ВидСубконто, который имеет в свою очередь набор атрибутов типа «ВидСубконто» соответствующих имеющимся видам субконто. Кроме того, глобальный атрибут ВидСубконто имеет методы для обхода всех существующих видов субконто. Операция — средство для манипулирования из встроенного языка данными бухгалтерских операций и проводок формируемых документом. Так как проводки в системе 1С:Предприятие принадлежат операциям, то управление и операциями и проводками выполняется объектом «Операция».
- БухгалтерскиеИтоги – средство для организации доступа к бухгалтерским итогам в различных разрезах, за различные периоды и с разной степенью детализации. При наличии в системе 1С:Предприятие компоненты «Бухгалтерский учет» система автоматически реализует специальный механизм работы с бухгалтерскими итогами. Данный механизм обеспечивает хранение, динамический пересчет бухгалтерских итогов и их извлечение средствами встроенного языка. Система хранения бухгалтерских итогов поддерживается системой 1С:Предприятие автоматически на основе существующих планов счетов. При редактировании планов счетов — в конфигураторе или при работе с системой 1С:Предприятие — для счета могут быть установлены ряд свойств, которые влияют на организацию хранения бухгалтерских итогов: это признаки ведения валютного и количественного учета, а также включение аналитического учета по субконто. Изменение бухгалтерских итогов может производиться только проводками бухгалтерских операций.

Каждый агрегатный тип данных, как правило, имеет набор атрибутов и методов. Атрибуты по свойствам напоминают переменные, т. е. им можно присваивать или читать их значения. Методы – это те действия, которые может выполнять агрегатный тип данных. Методы могут иметь или не иметь возвращаемое значение. Если метод имеет возвращаемое значение, то он может размещаться в правой части оператора присваивания, в выражениях, в описании фактических параметров других вызываемых методов, процедур или функций.

Типичная последовательность работы с объектом агрегатного типа данных выглядит следующим образом:

1. С помощью функции СоздатьОбъект создается объект агрегатного типа данных, и какой-либо переменной присваивается ссылка на него;
2. Объект позиционируется на нужном элементе данных;
3. Производятся различные манипуляции с объектом агрегатного типа данных через вызовы методов и обращения к его атрибутам.
4. В случае, если объект агрегатного типа данных больше не нужен, он может быть отсоединен от переменной посредством переписывания переменной какого-либо значения базового типа (например, числа 0). Отсоединение объекта выполнять не обязательно.

```
// Создаем объект Сотр = СоздатьОбъект(“Справочник.Сотрудники”); // Позиционируем созданный объект по
наименованию Сотр.НайтиПоНаименованию(“Петров”); Если Сотр.Выбран()=1 Тогда Сообщить(“Петров
”+Строка(Сотр.Оклад)); Иначе Сообщить(“Петров не найден”); КонецЕсли;
Для преобразования значения одного базового типа в другой используются функции Строка(<Значение>),
Число(<Значение>), Дата(<Значение>).
```

СИМВОЛЫ

Символ	Значение
[]	В квадратных скобках заключаются необязательные синтаксические элементы.
()	Круглые скобки заключают в себе список параметров.
	Вертикальной линией разделяются синтаксические элементы, среди которых нужно выбрать только один.

Специальные символы, используемые в исходном тексте

//	Двумя знаками «косая черта» начинается комментарий. Комментарием считается весь текст от знака "/" до конца текущей строки.
	Вертикальная черта в начале строки используется только в строковых константах и означает, что данная строка является продолжением предыдущей (перенос строки), (см. «Строковые константы»).
~	Знаком тильда начинается метка оператора.
:	Двоеточием заканчивается метка оператора.
;	Точка с запятой является символом разделения операторов.
()	В круглые скобки заключается список параметров методов, процедур и функций.
[]	В квадратные скобки заключается размерность массивов.
,	Запятая разделяет параметры в списке параметров методов, процедур и функций.
" "	В двойные кавычки заключаются строковые константы.
' '	В одинарные кавычки заключаются константы даты.
.	Десятичная точка в числовых константах. Разделитель, используемый в описаниях агрегатных типов данных.
+	Символ «плюс» обозначает операцию сложения.
-	Символ «минус» обозначает операцию вычитания.
*	Символ «звездочка» обозначает операцию умножения.
/	Символ «косая черта» обозначает операцию деления.
>	Правая угловая скобка обозначает логическую операцию «больше».
>=	Логическая операция «больше или равно».
<	Левая угловая скобка обозначает логическую операцию «меньше».
<=	Логическая операция «меньше или равно».
=	Знак равенства обозначает присвоение или логическую операцию «равно».
<>	Две угловые скобки обозначают логическую операцию «не равно».

Управляющие операторы

Для управления логикой выполнения программы используются логические выражения и управляющие операторы. Логическое выражение записывается с помощью символов сравнения = (равно), <> (не равно), > (больше), < (меньше), <= (меньше либо равно), >=(больше или равно), при этом сравнение производится только над значениями одинаковых типов (то есть нельзя, например, сравнивать число и строку). Из нескольких сравнений можно построить более сложное условие с помощью булевых операций «и», «или», «не». Порядок сравнений можно определять с помощью круглых скобок.

Оператор ветвления

Если <Логическое_выражение> Тогда

...

ИначеЕсли <Логическое_выражение> Тогда

...

Иначе

...

КонецЕсли;

Циклы

Пока <Логическое_выражение> Цикл ... [Прервать] ... [Продолжить] ... КонецЦикла; Для <Имя_переменной> = <Выражение1> По <Выражение2> Цикл ... [Прервать] ... [Продолжить] ... КонецЦикла;

Величина приращения счетчика при каждом выполнении цикла равна 1. Условие выполнения цикла всегда проверяется вначале, перед выполнением цикла.

Обработка ошибок

Попытка ... [Исключение] ... КонецПопытки;

Работа с транзакциями

Транзакции применяются для выполнения длительных и критических для функционирования системы операций. В некоторых случаях система сама начинает транзакцию и специально описывать ее не надо. Например, это происходит при проведении документов: так как в программе могут работать одновременно несколько пользователей, то при записи документа (например, продажа товара со склада) недопустима ситуация «одновременного» проведения двух документов – все равно один из них должен быть проведен раньше, чем другой (иначе можно продать один и тот же товар два раза!). Поэтому система выполняет проведение документа в режиме транзакции, – база данных становится недоступной для записи остальных пользователей. Естественно, что при большом количестве пользователей, которые активно вводят данные в программу, работа системы замедляется. Одним из решений этой проблемы может быть установка версии для SQL, в этом случае процедуры записи выполняется на отдельном компьютере (сервере базы данных) с помощью MS SQL Server 7.0.

В остальных случаях транзакцию начать и зафиксировать с помощью следующих функций:

НачатьТранзакцию(); ... [ОтменитьТранзакцию();] ... ЗафиксироватьТранзакцию();

Отмена транзакции применяется в случае обнаружения какой-то некорректной ситуации. В этом случае все изменения, внесенные в базы данных с момента начала транзакции, отменяются. Чтобы зафиксировать изменения необходимо выполнить функцию ЗафиксироватьТранзакцию().

Работа с объектом «СписокЗначений»

Объект «СписокЗначений» применяется для создания динамических списков (не сохраняемых в БД), которые могут отображаться в диалоговых формах (элемент «Список» и «Поле со списком») для выбора одного или нескольких значений из списка. Список создается с помощью функции СоздатьОбъект(«СписокЗначений»), либо визуально при добавлении на форму элементов «Список» или «Поле со списком».

Элемент списка содержит три поля – значение, представление и пометка. Значение может принимать значение любого типа, представление же всегда имеет тип «Строка». В форме диалога отображается представление (если оно задано). Пометка означает, помечено данное значение или нет.

Основные функции работы со списком следующие:

- ДобавитьЗначение(<Знач>, <Строка>)
- НайтиЗначение(<Знач>)
- ПолучитьЗначение(<Позиция>, <Перем>)
- Получить(<Строка>)
- УдалитьЗначение(<Позиция>, <Колич>)
- УдалитьВсе()
- РазмерСписка()
- Сортировать(<Направление>)
- СортироватьПоПредставлению(<Направление>)
- Принадлежит(<Знач>)
- Пометка (<Позиция>, <Отметка >)
- ИзСтрокиСРазделителями(<Строка>)
- ВСтрокуСРазделителями()

Работа с объектом «ТаблицаЗначений»

Объект «ТаблицаЗначений» применяется для создания динамических массивов (не сохраняемых в БД), которые могут отображаться в диалоговых формах (элемент «Таблица значений»). Таблица значений создается с помощью функции СоздатьОбъект(«ТаблицаЗначений»), либо визуально при добавлении на форму элементов «Таблица значений». Пример.

```
Т=СоздатьОбъект(«ТаблицаЗначений») // Определяем идентификаторы и параметры колонок
Т.НоваяКолонка(«Товар», «Строка», 30, «Наименование товара», 20); Т.НоваяКолонка(«Цена», «Число», 10,
2, «Цена», 10); Т.НоваяКолонка(«Количество», «Число», 10, 3, «Кол.»), 10); Т.НоваяКолонка(«Сумма»,
«Число», 15, 3, «Сумма», 10); // Создаем новую строку Т.НоваяСтрока(); Т.Товар= «Сыр»; Т.Цена= 50;
Т.Количество= 10; Т.Сумма= Т.Цена* Т.Количество;
```

Основные функции работы с таблице следующие:

- НоваяКолонка(<Идентификатор>,<Тип>,<Длина>,<Точность>,<Заголовок>,<Ширина>,<Формат>,<Положение>)
- НоваяСтрока()
- УдалитьСтроку(<НомерСтроки>)
- УдалитьСтроки()
- ПолучитьСтрокуПоНомеру(<НомерСтроки>)
- ПолучитьЗначение(<НомерСтроки>,<Колонка>) <Колонка> - это либо номер колонки либо идентификатор колонки
- НайтиЗначение(<Знач>,<Строка>,<Колонка>) <Строка> – идентификатор переменной, в которую возвращается номер найденной строки, <Колонка> – идентификатор переменной, в которую возвращается колонка найденной строки.
- ВыбратьСтроки() – открыть выборку строк из таблицы
- ПолучитьСтроку() – получить следующую строку из выборки
- Сортировать(<Колонки>) <Колонки> - это строка, содержащая список идентификаторов или номеров колонок, разделенных запятой, по которым выполняется сортировка строк таблицы значений
- Свернуть(<ГрупКолонки>,<СумКолонки>) <ГрупКолонки> - это строка, содержащая список идентификаторов или номеров колонок, разделенных запятой, по которым выполняется группировка строк таблицы значений (то есть если имеется несколько строк с одинаковыми значениями в группируемых колонках, то в результате свертки останется одна такая строка). <СумКолонки> - это строка, содержащая список идентификаторов или номеров колонок, разделенных запятой, в которых выполняется суммирование значений строк таблицы значений (то есть, если группируется несколько строк, то значения суммируемых колонок будут складываться).

Запуск внешних приложений из 1С.

Для запуска внешнего приложения можно использовать функцию ЗапуститьПриложение(). Эта функция использует средства OLE-Automation и позволяет использовать серверы автоматизации в приложениях 1С. Небольшой пример интеграции 1С и Excel приведен в листинге 37.

Листинг 37. Запуск внешнего приложения из программы 1С

```
Окно = СоздатьОбъект(«Excel.Application»); Окно.Visible = 1; // делаем окно видимым Окно.Caption =
«Отчет»; // задаем имя окну Окно.Workbooks.Add(); // создаем новую рабочую книгу Для Н=1 По 10 Цикл
Ячейка = Окно.Cells(Н,1); Ячейка.Value = Н; КонецЦикла;
```

Лекция 10: Конструкция языка 1С

1С Предприятие является гибкой настраиваемой системой, с помощью которой можно решать широкий круг задач в сфере автоматизации деятельности предприятий.

Специфические алгоритмы конфигурации описываются в системе 1С: Предприятие в режиме запуска Конфигуратор в программных модулях, содержащих тексты на встроенном языке системы 1С:Предприятие. Встроенный язык представляет собой предметно-ориентированный язык программирования, специально разработанный с учетом возможности его применения не только профессиональными программистами.

Программный модуль- это текст на встроенном языке, в котором размещены тексты процедур и функций с необходимыми алгоритмами, вызываемые системой во время работы системы в определенные моменты. Место размещения программного модуля предоставляется конфигуратором в тех точках конфигурации, которые требуют описания специфических алгоритмов функционирования. Эти алгоритмы следует оформлять в виде процедур или функций, которые будут вызваны самой системой в заранее предусмотренных ситуациях. Каждый отдельный программный модуль воспринимается системой как единое целое, поэтому все процедуры и функции программного модуля выполняются в едином контексте. Каждый программный модуль связан с остальной частью конфигурации. Эта связь называется контекстом выполнения модуля.

Следует различать два вида контекста:

Глобальный контекст задачи, который образуется:

- значениями свойств и методов глобального контекста;
- системными перечислениями и системными наборами значений.

Глобальный контекст виден всем программным модулям и определяет общую языковую среду конфигурации.

Локальный контекст модуля образуется тем конкретным местом конфигурации задачи, для которого использован программный модуль. Локальный контекст виден только конкретному модулю и определяет для модуля набор непосредственно доступных модулю объектов, их свойств и методов.

В системе 1С:Предприятие существуют несколько видов программных модулей.

Они различаются по месту размещения и доступному контексту.

- Модуль приложения
- Модуль внешнего соединения
- Общие модули
- Модули прикладных объектов
- Модули форм

Структура программного кода

// Раздел объявления переменных

Перем А ; Перем Пустая строка ; Перем Пустое число;

Перем Пустая дата ;

// Раздел процедур и функций

// Раздел основной программы

Пустая строка = “ “; // Строковая переменная

Пустое число = 0; // Числовая переменная

Пустая дата = ‘ 00010101 ’; //Переменная даты (год, месяц, число)

**А = ложь ; // Булево- переменная логического типа
(или истина)**

Команда вывода информации на экран

Сообщить (“Привет, мир!!”);

S=6 ;

Сообщить (“ сумма S =”+ S);

Дата = ‘198710071015’ ; //дата и время

Сообщить (Дата);

Результат

Привет, мир!!

Сумма S = 6

07.10.1987 10.15.00

Команда ветвления

$A=6$; $B=3$; $C=5$;

$D = B * B - 4 * A * C$;

Если $D < 0$ Тогда Сообщить (“ Корней нет”);

Если Иначе $D = 0$ Тогда Сообщить (“ Один корень”);

Иначе Сообщить (“ Два корня”);

КонецЕсли

Результат

Корней нет

Зарплата Выдана Всем = ложь ;

Если Зарплата Выдана Всем Тогда Сообщить (“ Ура!!”);

Иначе Сообщить (“ Жаль !!”);

КонецЕсли

Результат

Жаль!!

Команды цикла

$D=0$;

Для $K = 1$ По 10

Цикл

$D = D + K$;

КонецЦикла ;

Сообщить (“ $D=$ ”+ D);

Результат

$D=55$

$X=0$;

Пока $X < 5$

Цикл

Сообщить(X);

$X=X+1$;

КонецЦикла;

Результат

0

1

2

3

4

Процедуры и функции

Процедура ПередНачаломРаботыСистемы (Отказ)

$A=3$;

Сообщить (A);

Тест (A);

Сообщить (A);

КонецПроцедуры

Процедура Тест (Входная)

Входная= Входная+ 10;

КонецПроцедуры

Результат

3

13

Процедура ПередНачаломРаботыСистемы (Отказ)

A=3;

Сообщить (A);

B=Тест (A);

Сообщить (B);

КонецПроцедуры

Функция Тест (Входная) Экспорт //Функция будет

Видна всем модулям

Входная= Входная+ 10;

Возврат Входная;

КонецФункции

Результат

3

13

M = Новый Массив ();

M.Добавить(1);

M.Добавить(5);

M.Добавить(15);

M.Добавить(7);

M.Добавить(9);

Сообщить (“Вывод элемента массива”);

Для Каждого Элементмассива Из M

Цикл

Сообщить (Элементмассива);

КонецЦикла;

Перем МасЗнач;

МасЗнач=Новый Массив();

Для л=1 По 10 Цикл

МасЗнач.Добавить(л);

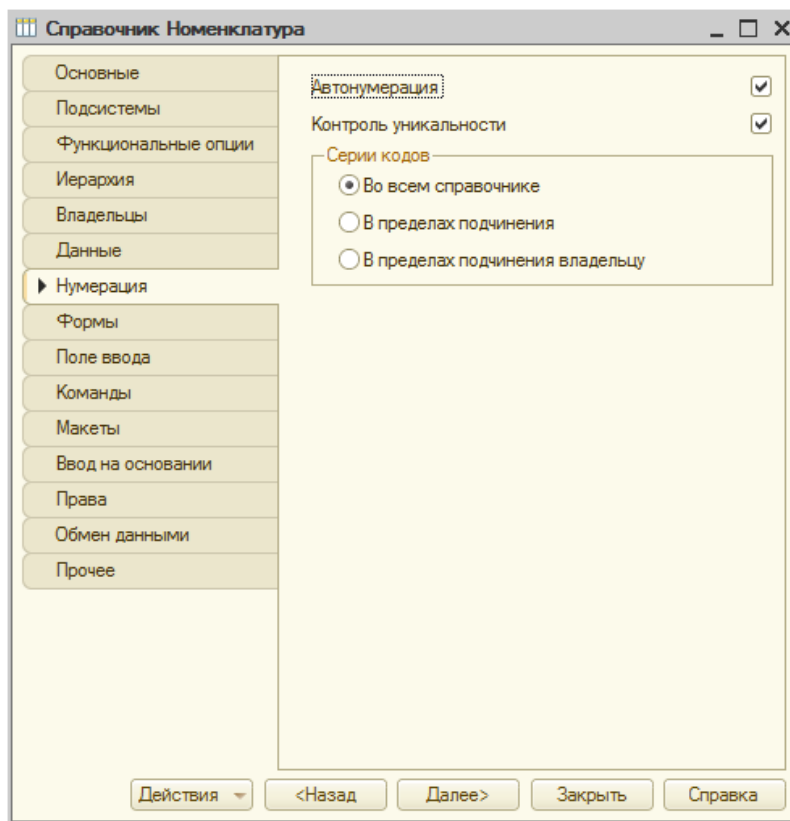
КонецЦикла;

Лекция 11: Справочники

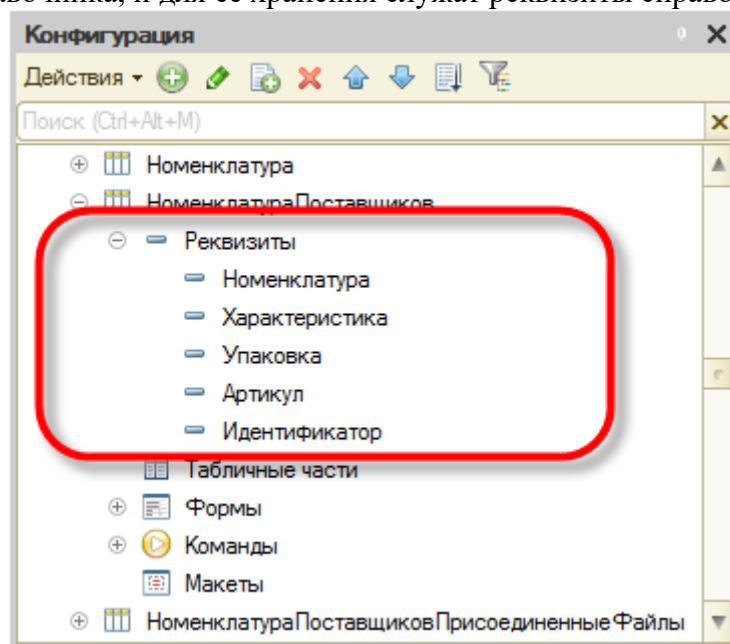
Справочники — это прикладные объекты конфигурации. Они позволяют хранить в информационной базе данные, имеющие одинаковую структуру и списочный характер. Это может быть, например, список сотрудников, перечень товаров, список поставщиков или покупателей.

Структура справочника

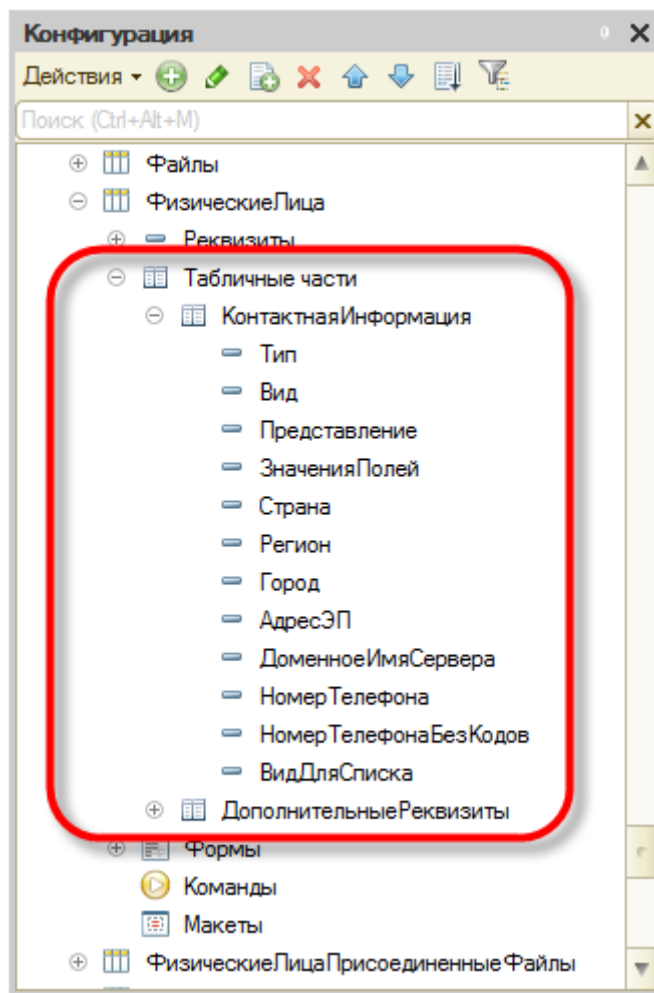
Каждый элемент справочника характеризуется кодом и наименованием. Система поддерживает режим автоматической нумерации элементов, при котором она самостоятельно может генерировать код для нового элемента справочника. Кроме этого система позволяет осуществлять контроль уникальности кодов справочника, не разрешая создавать элементы с одинаковыми кодами:



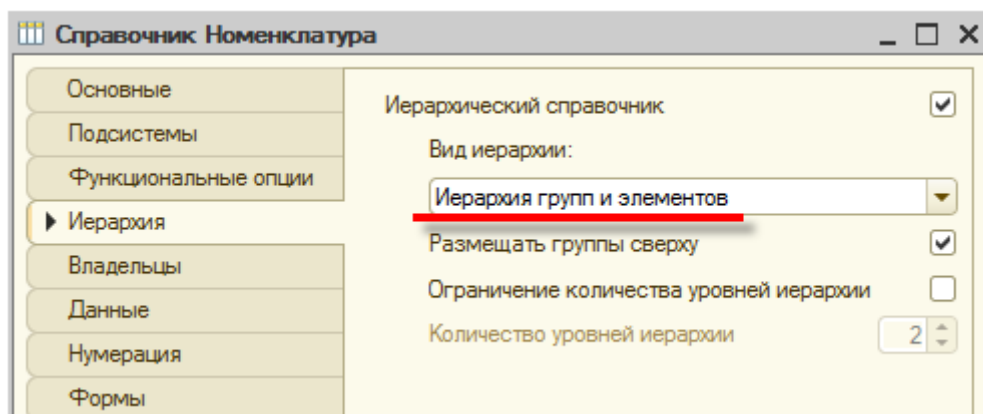
Помимо кода и наименования, каждый элемент справочника, как правило, содержит некоторую дополнительную информацию, которая подробно описывает этот элемент. Например, для товара это может быть информация об артикуле, упаковке и т. п. Набор такой информации является одинаковым для всех элементов конкретного справочника, и для ее хранения служат реквизиты справочника:



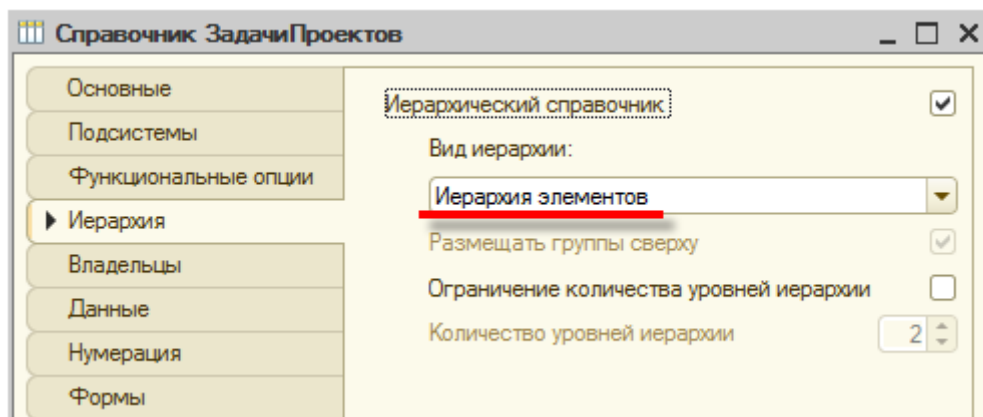
Кроме этого, каждый элемент справочника может содержать некоторый набор информации, которая одинакова по своей структуре, но различна по количеству, для разных элементов справочника. Например, для каждого сотрудника в справочнике **Физические лица** это может быть контактная информация или информация о составе семьи, образовании. Для хранения подобных данных служат табличные части справочника:



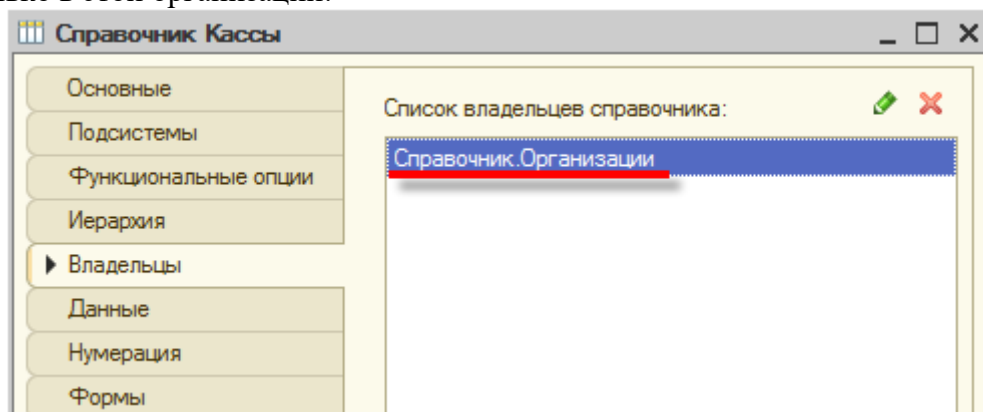
Справочники могут поддерживать иерархическое расположение элементов. Например, в справочнике **Номенклатура** могут быть созданы группы: **Бытовая техника**, **Обувь**, **Продукты** и т. д., в которых располагаются элементы, относящиеся к этим группам. Кроме того группы справочника могут включать в себя и другие группы, создавая тем самым многоуровневую иерархическую структуру.



В справочниках возможен и другой вид иерархии, при котором элементы справочника будут относиться не к группам, а к другим элементам этого же справочника. Такой вид иерархии может использоваться, например, при создании справочника **Задачи**, где одна задача может включать в себя несколько других задач.

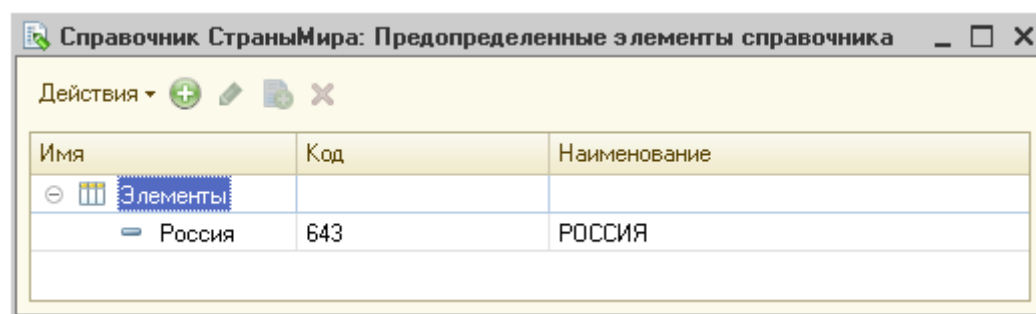


Разные справочники могут находиться в состоянии подчинения, т. е. элементы одного справочника могут быть подчинены элементам или группам другого справочника. Например, справочник **Кассы** может быть подчинен справочнику **Организации**. Тогда при оформлении кассовых документов для некоторой организации можно будет выбрать кассу не среди всех имеющихся в программе касс, а среди касс, существующих только в этой организации:



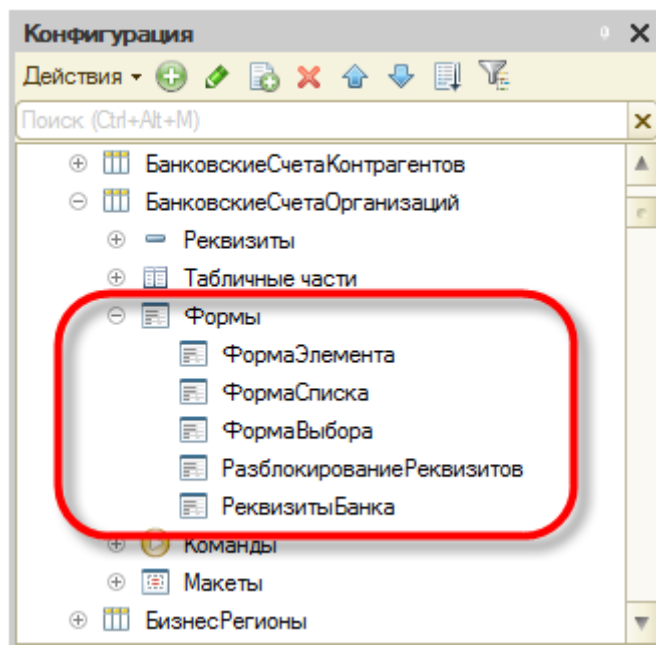
Справочники допускают также создание predetermined элементов, которые существуют в справочнике всегда, вне зависимости от действий пользователя. Такие элементы справочника создаются разработчиком при разработке прикладного решения и не могут быть удалены или перемещены пользователем.

Например, если в основном прикладное решение будет внедряться на предприятиях одного государства, России, в справочник **Страны мира** сразу же можно добавить predetermined элемент **Россия**, поскольку он понадобится подавляющему большинству пользователей.



Формы справочника

Для того чтобы пользователь мог просматривать и изменять данные, содержащиеся в справочнике, система поддерживает несколько форм представления справочника. Система может автоматически генерировать все нужные формы справочника. Наряду с этим разработчик имеет возможность создать собственные формы, которые система будет использовать вместо форм по умолчанию:



Форма списка

Для просмотра данных, содержащихся в справочнике, используется форма списка. Она позволяет выполнять навигацию по справочнику, добавлять, помечать на удаление и удалять элементы и группы справочника, перемещать элементы и группы. Форма списка может представлять данные в иерархическом и не иерархическом виде и позволяет выполнять сортировку и отбор отображаемой информации по нескольким критериям:

Наименование	Код	Артикул	Поставщик
Bosch1234	000000017	Ч-0001	ЭлектроБыт ЗАО
Bosch15	000000018	Ч-0002	ЭлектроБыт ЗАО
Sony K3456P	000000040	H657	ЭлектроБыт ЗАО
Veko345MO	000000028	VEKO00001	ЭлектроБыт ЗАО
Veko67NE	000000022	Ч-0003	ЭлектроБыт ЗАО
Veko876N	000000034	876N	ЭлектроБыт ЗАО
VekoNT02	000000033	NT02	ЭлектроБыт ЗАО

Форма элемента

Для просмотра и изменения данных отдельных элементов справочника используется форма элемента. Как правило, она представляет данные в удобном для восприятия и редактирования виде:

🏠 ← → ☆ **Валенки (Товар)** ✕

Главное Файлы Цены товаров

Записать и закрыть Записать Создать на основании ▾ Печать ▾ Еще ▾ ?

Группа: ▾ 🔗

Код:

Наименование:

Вид: Товар Услуга


Артикул:

Поставщик: ▾ 🔗

Штрихкод:

Изображение Характеристики Описание

Картинка: ... ✕ 🔗 Обновить картинку



Форма группы

Система поддерживает отдельный вид формы для групп справочника. Это связано с тем, что, как правило, состав информации, относящейся к группе справочника, намного меньше, чем состав данных, имеющих отношение к отдельным элементам справочника:

Обувь (Товары)

Записать и закрыть Записать Еще ▾ ?

Код:

Наименование:

Группа: ▾ 🔗

Форма выбора, форма выбора группы

При различных действиях пользователя в прикладном решении возникает необходимость выбора конкретных элементов справочников. Это может потребоваться, например, при создании приходной накладной, в которой должны быть перечислены определенные товары. В этом случае система использует форму выбора справочника, которая, обычно, содержит минимальный набор информации, необходимой для выбора того, или иного элемента:

Товары

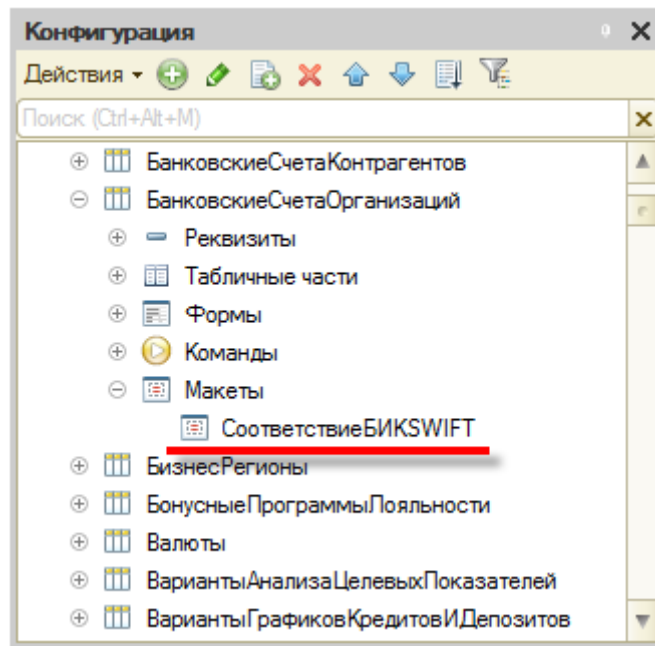
Выбрать Создать Создать группу Поиск (Ctrl+F) × Еще ?

Наименование	Код
Электротовары	000000012
Чайники	000000013
Bosch1234	000000017
Bosch15	000000018
Veko67NE	000000022
VekoNT02	000000033

Кроме всех перечисленных форм, система поддерживает еще форму выбора группы справочника, предназначенную для выбора только среди групп, содержащихся в справочнике.

Макеты

Справочнику могут быть сопоставлены несколько макетов, содержащих данные, необходимые для обеспечения работы справочника:



Макеты могут использоваться для формирования печатных форм справочника или для отображения дополнительной информации, имеющей отношение к справочнику.

Лекция 12: Основы программирования

В современных условиях при автоматизации предприятий приходится сталкиваться с различными и часто диаметрально.

«1С: Предприятие» является универсальной системой автоматизации деятельности предприятия. 1С: Предприятие - это специализированная объектно-ориентированная система управления базами данных (СУБД), предназначенная для автоматизации деятельности предприятия.

Учет – специфическая деятельность по регистрации свершившихся хозяйственных операций с целью хранения факта операции, группировки анализа подготовки отчетности для управления отдельными участками и предприятием в целом.

1С: Предприятие 8.3 представляет собой систему программ и включает в себя: технологическую платформу, созданные на ее основе прикладные решения различного масштаба и различной направленности, методологию создания прикладных решений, информационно-технологическую поддержку пользователей и разработчиков. Сама платформа не является программным продуктом для использования конечными пользователями, которые обычно работают с одним из многих прикладных решений (конфигураций), разработанных на данной платформе.

Такой подход позволяет автоматизировать различные виды деятельности, используя единую технологическую платформу.

1С: Предприятие включает 4 типовые конфигурации:

«1С: Бухгалтерия»,

«Управление производственным предприятием», «Управление торговлей»,

«Зарплата и Управление персоналом».

Помимо этого, существуют различные локализованные прикладные решения, разрабатываемые фирмой 1С на платформе «1С: Предприятие», обеспечивающие ведение учета, формирование первичных документов и отчетности в соответствии с требованиями национального законодательства.

Система «1С:Предприятие 8» может функционировать в двух основных режимах:

- Конфигурирование
- Использование (1С:Предприятие)

Режим «Конфигурирование» предназначен для разработки прикладного решения задачи

- На этапе конфигурирования системы 1С: Предприятие выполняется настройка различных режимов системы в соответствии с особенностями конкретного предприятия. При этом определяются структуры и свойства справочников и документов, описываются формы и алгоритмы построения отчетов, настраивается ведение учета.
- При конфигурировании системы разработчик конфигурации или администратор системы может также создать наборы прав доступа к данным, соответствующие пользователям разного уровня.
- На этапе конфигурирования может быть создано: несколько пользовательских интерфейсов для разных типов пользователей (менеджеров, бухгалтеров, кадровиков и т.д.), а также для конкретной организации создается список пользователей.

Режим «1С:Предприятие» для использования написанного в конфигураторе программного кода и работы пользователя.

- Работа пользователя с информационной базой осуществляется при запуске системы в режиме «1С: Предприятие» и включает: ввод документов, заполнение справочников, формирование различных отчетов, выполнение различных регламентных расчетов и т.д.
- Пользователю предоставляется возможность ввода и обработки информации описанной в конфигурации структуры с использованием алгоритмов, созданных на этапе конфигурирования. Пояснение особенностей работы с конкретными режимами (документами, справочниками, отчетами), которые зависят от конфигурации, могут содержаться в описании самой конфигурации. В этом случае их можно получить, обратившись к режиму пользовательского описания.

Запуская систему, пользователь должен обязательно работать с какой-либо информационной базой (ИБ). ИБ содержит конфигурацию, базу данных (БД), список пользователей и некоторые другие параметры. Для удобства пользователей ИБ сохраняется в одном файле - «1cv8.1CD».

Существует две версии программы «Толстый клиент» и «Тонкий клиент»

Толстый клиент позволяет запускать конфигуратор и работать в пользовательском режиме в рамках локальной сети.

Тонкий клиент – это отдельная программа, предназначенная только для работы в пользовательском режиме. Может работать через Интернет.

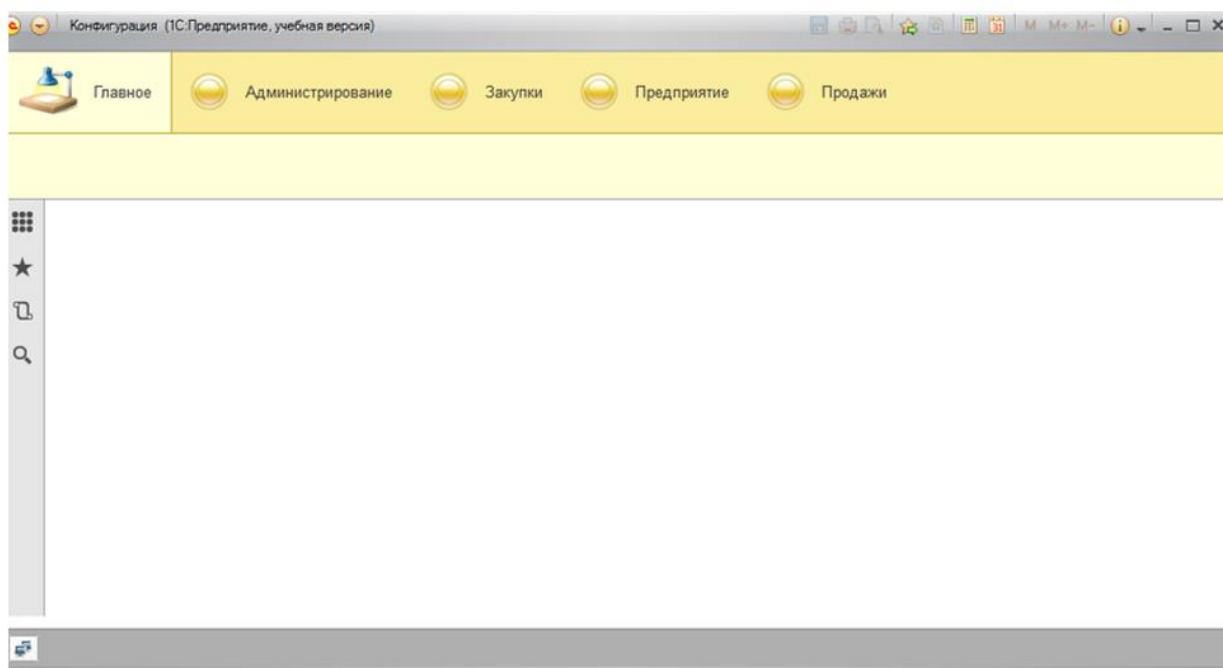
В некоторых случаях на компьютере пользователя можно вообще не устанавливать «1С», а вместо клиентского приложения использовать Интернет-браузер. Такой вариант подключения называется Web-клиент и предназначен для удаленной работы пользователей.

		Программы		
		Толстый	Тонкий	Web
Режимы запуска	1С:Предприятие	Да	Да	Да
	Конфигуратор	Да	Нет	Нет

ЗАДАНИЕ:

Создаем четыре подсистемы:

- Предприятие – включает все общие справочники и механизмы
- Закупки – все, что потребуется пользователю для оформления покупки товаров
- Продажи – все, что потребуется пользователю для оформления продажи
- Администрирование – служебные функции программы.



Закладка «Основные» справочника «Организации»

Справочник: Организации

Основные

Имя:

Синоним:

Комментарий:

Представление объекта:

Расширенное представление объекта:

Представление списка:

Расширенное представление списка:

Пояснение:

Действия

Предопределенные элементы

Справочник: Организации

Предопределенные

Модуль объекта

Модуль менеджера

Предопределенные

Обновление предопределенных данных:

Справочник Организации: Предопределенные элементы справочника

Имя	Код	Наименование
Элементы		

Элемент справочника

Родитель:

Имя:

Код:

Наименование:

Действия

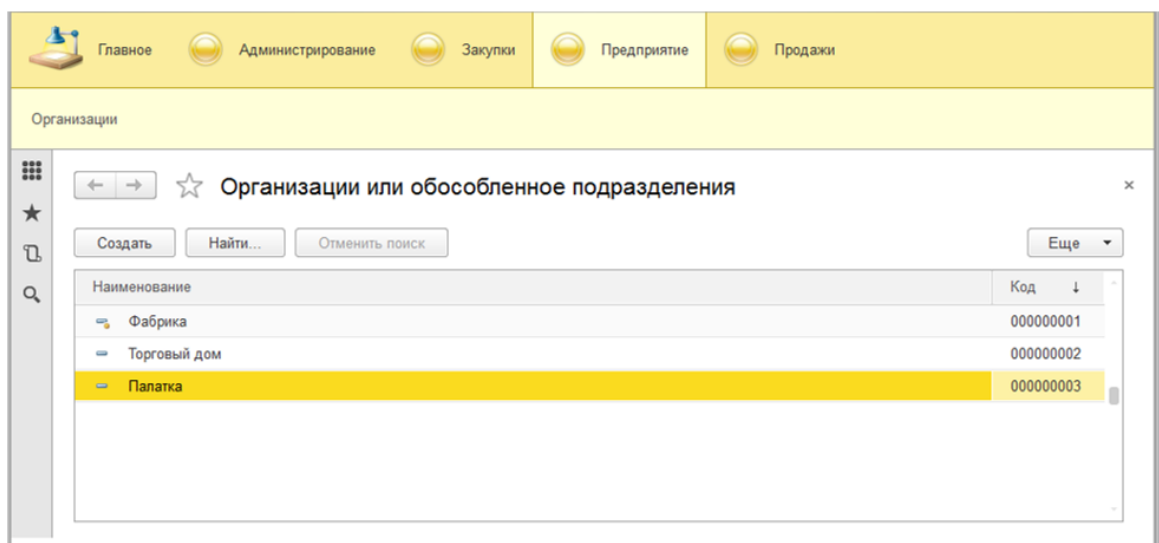
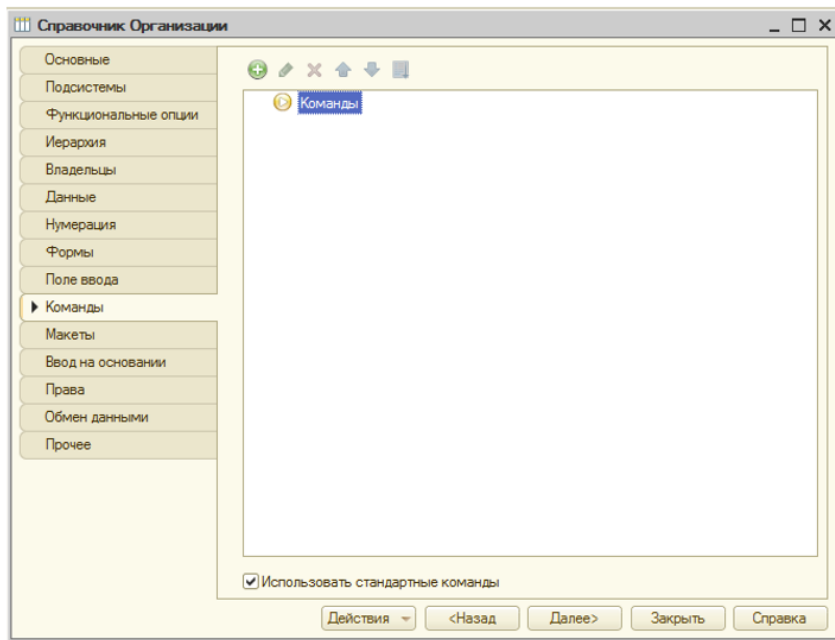
Справочник: Организации

Подсистемы

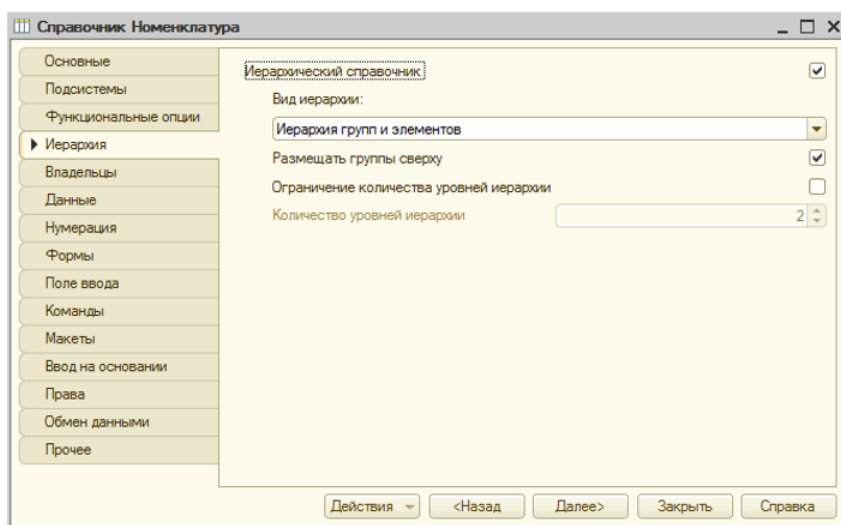
Подсистемы, в которых участвует справочник:

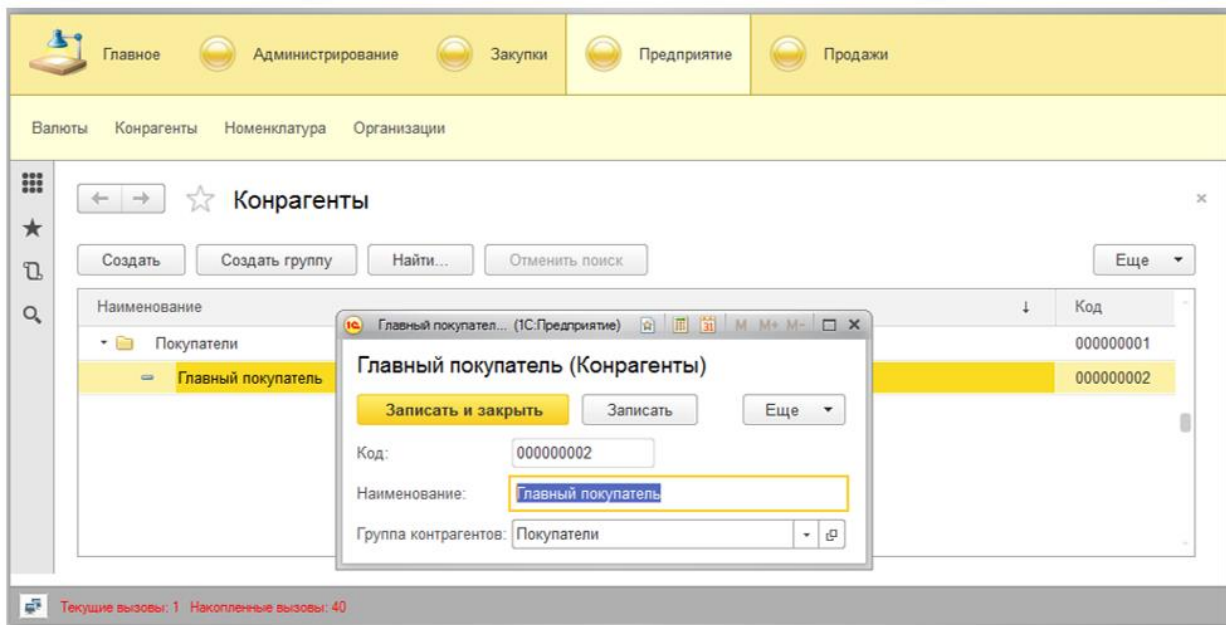
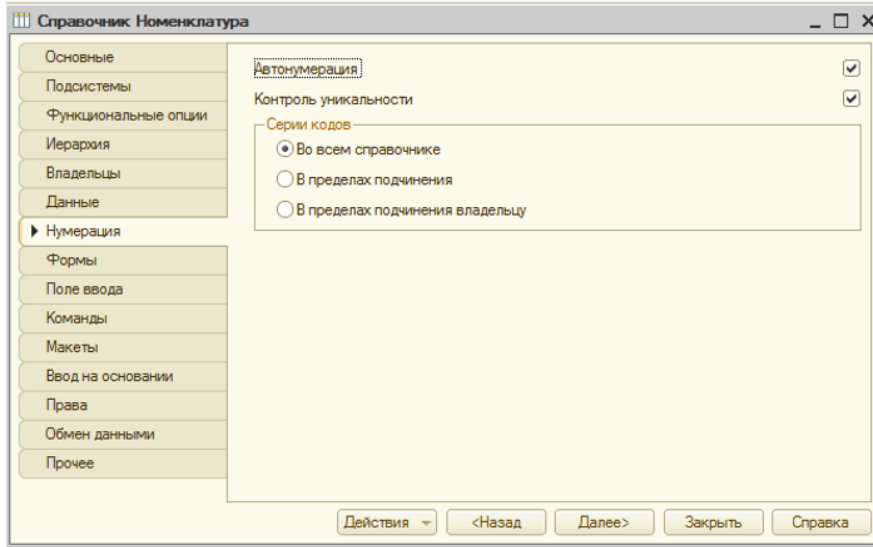
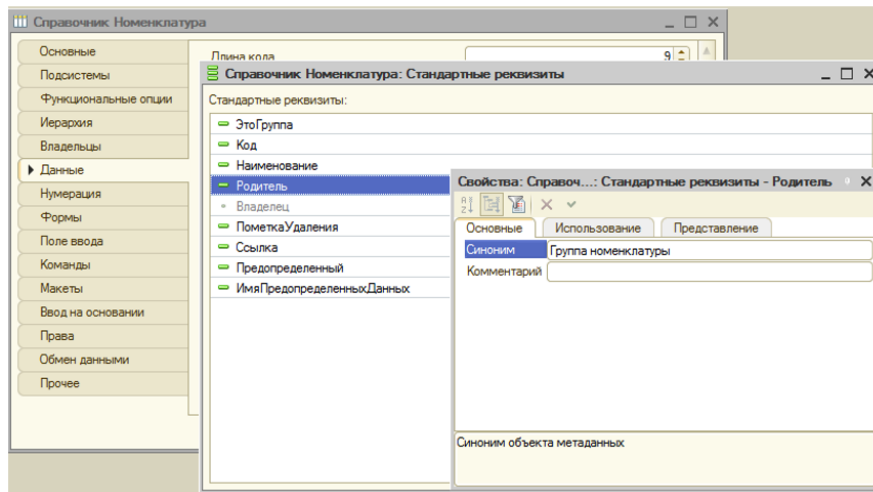
- Предприятие
- Закупки
- Продажи
- Администрирование

Действия

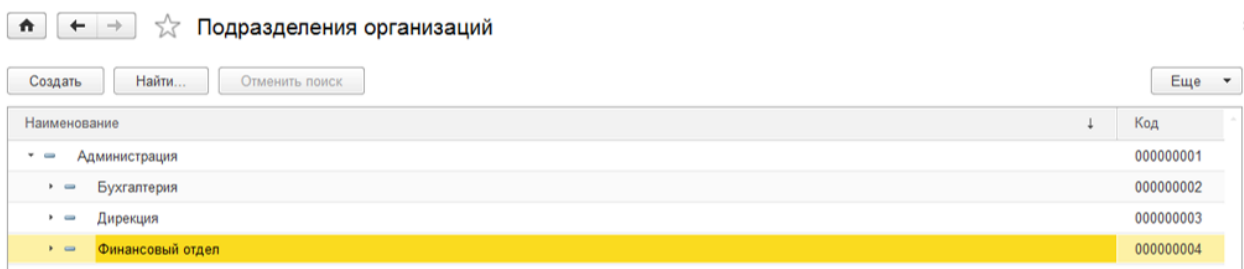
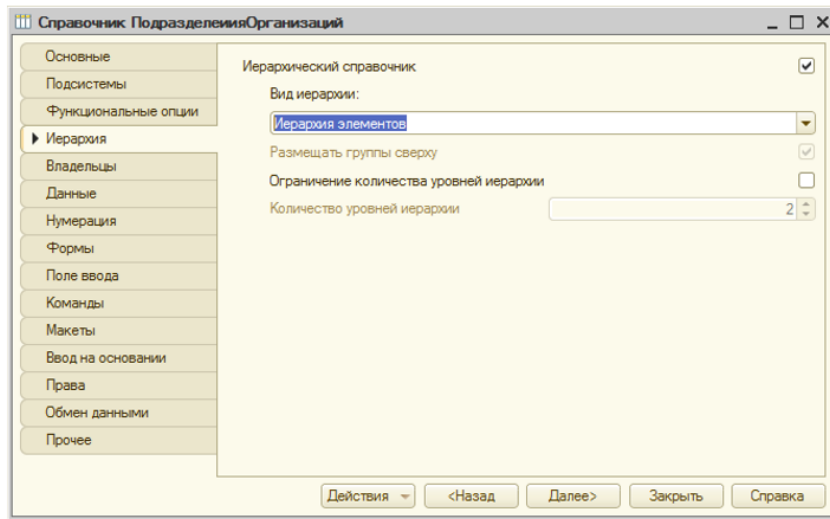


Иерархические справочники

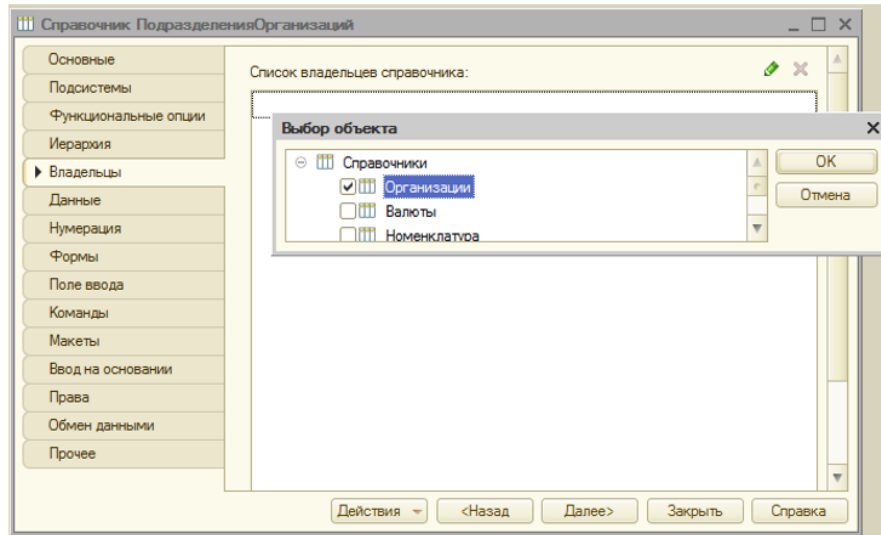


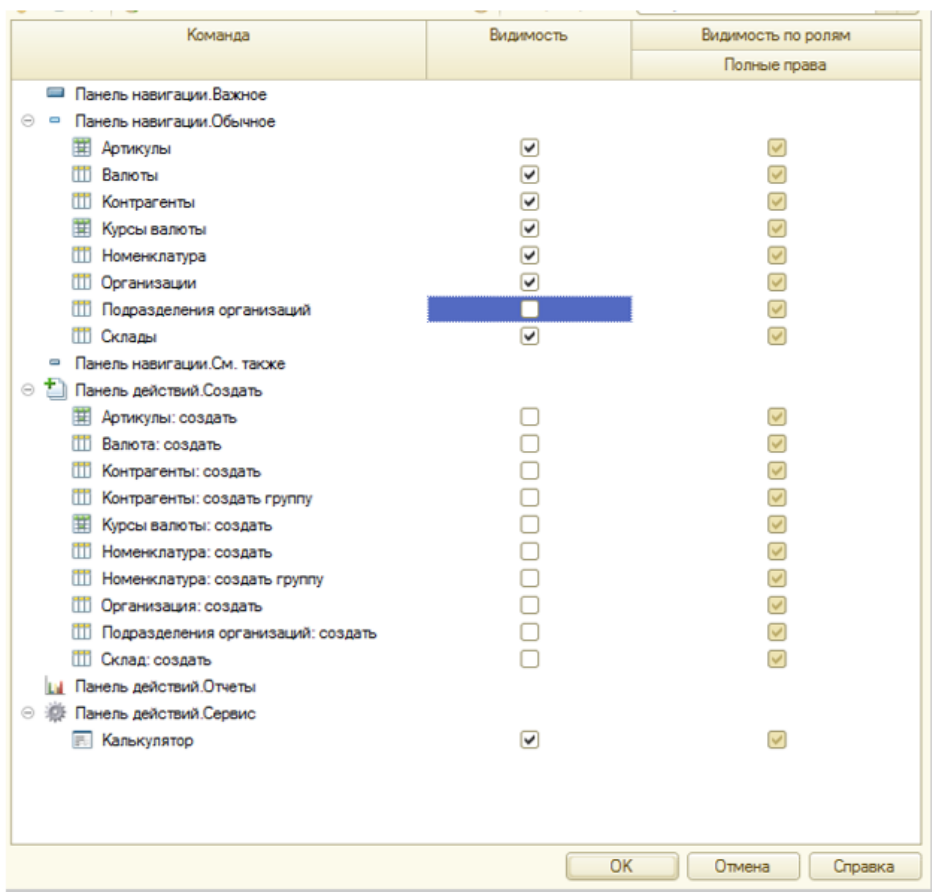
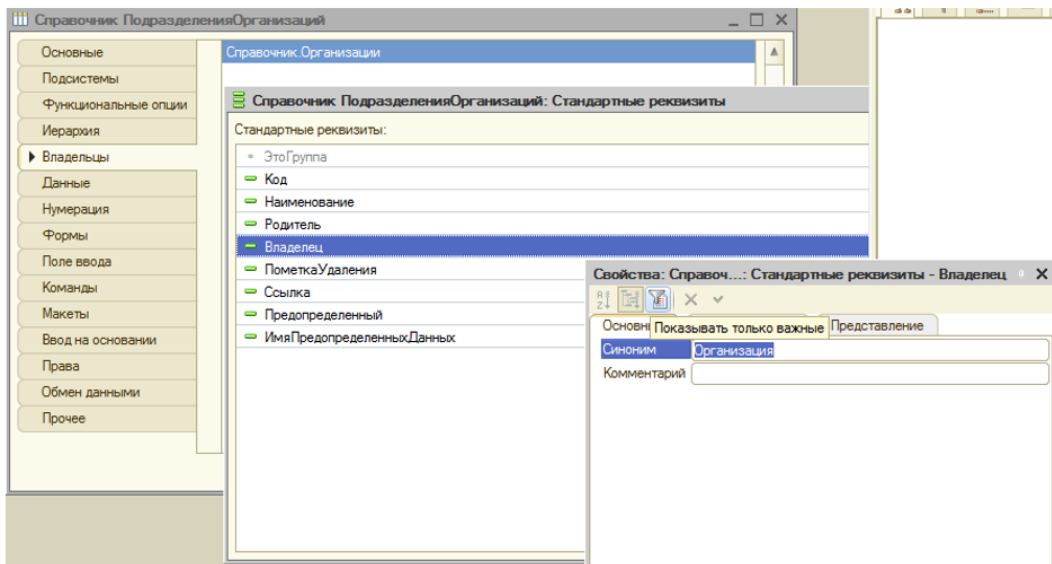


Иерархия элементов



Подчиненные справочники





Лекция 13: Документы

Структура лекции: структура и формы документов, ввод и редактирование данных, элементы журналов, ввод, редактирование и удаление записей в журналах

Документы

Любая операция хозяйственной деятельности предприятия может быть зафиксирована путем ввода информации в документ соответствующего вида и записи этого документа в журнал.

При настройке конфигурации задачи определяется: структура документов, вид диалогов для их ввода, печатные формы, алгоритмы изменения итоговой информации (например, алгоритм формирования бухгалтерских проводок). При описании документов на стадии конфигурирования существуют следующие возможности: документ любого вида имеет номер (который может быть как строковым, так и числовым) и дату, документ может иметь две основные части: реквизиты и табличные части. К реквизитам относятся данные, которые являются общими для документа, то есть имеют один экземпляр на документ. Например, для такого документа, как счет, это название клиента, реквизиты клиента и размер скидки. Табличная часть документа представляет собой список однотипных данных, которых может встречаться более одного на документ. Например, список товаров с количеством и номенклатурой для того же счета. В общем случае, табличная часть представляет совокупность строк определенной структуры. В некоторых документах табличная часть может использоваться как исходные данные для расчета общей суммы (например, общей стоимости выписанных по счету товаров). Документ в системе «1С:Предприятие 8.1» может содержать несколько табличных частей.

Ввод документа может осуществляться различными способами: ввод из журнала, ввод из меню. Если активным является окно какого-либо из существующих в системе журналов документов, то для ввода нового документа следует выполнить команду: «Действия - Добавить». На экране появится список видов документов, которые могут отображаться в текущем журнале. В списке видов необходимо указать наименование вида необходимого документа и нажать кнопку «ОК». На экран будет выведена форма нового документа. Технология ввода документов из меню зависит от конфигурации.

Каждый документ имеет поля: номер, дата, время документа. Эти реквизиты обязательны для заполнения, и программа не позволит закончить ввод документа, если они не заполнены. Если при настройке конфигурации для документа установлен признак автоматической нумерации, то поле для ввода номера документа будет содержать номер, автоматически сформированный системой. Этот номер можно исправить, однако программа проследит, чтобы указанный вами номер не совпадал с номерами документов этого же вида, уже введенных в систему. Если это обусловлено конфигурацией, при вводе документа его номер будет проверяться на уникальность по правилам, определенным в конкретной конфигурации, и, если при сохранении документа система обнаружит нарушение требований уникальности, будет выдано сообщение «Номер не уникальный!», тогда введенный номер следует исправить.

Система 1С: Предприятие предлагает в качестве даты документа рабочую дату, установленную в общих параметрах (пункт «Параметры» меню «Сервис» главного меню программы).

В общем случае рабочая дата равна текущей системной дате компьютера. Предложенную системой дату можно изменить. Для ввода новой строки в табличную часть документа используется клавиша INS. Для копирования строки в табличную часть документа используется клавиша F9. Для удаления строки из табличной части документа используется клавиша Del.


Новый документ может быть введен путем копирования уже существующего документа. Для этого следует в журнале с документами нужного вида поместить курсор на строку с документом, который предполагается копировать, и нажать клавишу F9 или выбрать пункт «Копировать» в меню «Действия». В результате на экран будет выдана форма документа, в которой все реквизиты, за исключением номера и даты документа, будут скопированы из реквизитов документа-образца.

Новому документу, если это обусловлено конфигурацией, присваивается очередной порядковый номер, а в качестве даты будет проставлена рабочая дата, установленная в общих параметрах. Корректировка реквизитов такого документа производится по общим правилам. Удаление документа производится в журнале документов с использованием клавиши Del.

Если в процессе настройки конфигурации задачи для документа была создана, помимо экранной формы (диалога), также печатная форма, то в диалоге для ввода реквизитов документа, как правило, присутствует

кнопка, нажатие на которую вызывает построение этой формы. Обычно такая кнопка имеет надпись «Печать», хотя может иметь и какую-либо другую подобную надпись.


После нажатия кнопки «Печать» (или выполнения каких-либо иных действий, если это указано в описании конфигурации) будет построена печатная форма документа. Созданная печатная форма открывается в стандартном редакторе таблиц системы 1С:Предприятие. Если в окне редактора таблиц видна только часть формы, можно использовать клавиши управления курсором и линейки прокрутки для вывода на экран остальной части формы. При необходимости такая форма может быть отредактирована и сохранена на диск с помощью команды «Файл - Сохранить как». Редактирование печатной формы производится аналогично редактированию в MS Excel.



После того, как заполнена форма документа, его необходимо сохранить. В зависимости от настройки конкретного документа возможны несколько вариантов выполнения этого действия: если диалог для ввода документа содержит соответствующие кнопки, например, «ОК», «Сохранить», «Записать» или аналогичные, сохранение документа выполняется путем нажатия на такую кнопку; если диалог для ввода документа не имеет кнопок, явно предназначенных для сохранения документа, то сохранение документа выполняется путем закрытия окна диалога. Для этого следует нажать на кнопку закрытия окна .

В обоих случаях на экран будет выдан запрос о необходимости сохранения документа.

Вы можете выбрать один из трех вариантов ответа: «Да» - сохранить документ и закрыть окно диалога для ввода документа; «Нет» - закрыть окно диалога для ввода документа, но не сохранять документ, «Отмена» - вернуться в диалог ввода документа.

Проведением документа называется действие, которое выполняет изменения других данных системы на основании информации документа. При нажатии кнопки формы документа, для которой определено проведение документа (обычно это кнопка «ОК» или «Провести»), на экран выдается запрос необходимости проведения документа. Если на запрос ответить «Да» - документ будет проведен.

В журнале документов проведенный документ будет помечен в крайней левой колонке знаком . Ответив «Нет» на запрос, можно отказаться от проведения документа сразу, и выполнить проведение в другое время. Обычно при проведении документа кнопкой «ОК» после проведения документ закрывается. В некоторых случаях при проведении документа возникает ситуация, из-за которой проведение не может быть выполнено. Например, при проведении расходной накладной обнаружено, что на складе не числится нужного количества товара. Будет выдано сообщение, о невозможности выполнить проведение документа.

В этом случае документ не будет закрыт автоматически, а будет предложено закрыть документ. Если отказаться от закрытия документа, то можно внести в него исправления и заново попробовать провести документ. Чтобы сделать документ не проведенным, следует в меню «Действия» выбрать функцию «Отмена проведения». После подтверждения признака  проведенного документа изменится на . При отмене проведения документа отменяются все действия, которые он выполнил в процессе проведения. Не проведенный документ можно откорректировать, записать, провести (или не проводить), а знак в крайней левой колонке журнала будет правильно отражать текущее состояние документа.

Журналы

В системе 1С: Предприятие информация о хозяйственных операциях, совершаемых на предприятии, может вводиться при помощи документов и сохраняться в журналах документов.

При настройке конфигурации производится настройка документов и создаются необходимые журналы для их хранения.

Количество журналов документов, с которыми может работать пользователь, определяется его набором прав и устанавливается в режиме «Конфигуратор». При настройке конфигурации системы «1С: Предприятие» можно определять практически неограниченное число журналов документов. Режим конфигурирования системы 1С: Предприятие предоставляет возможности по настройке состава колонок журнала, его экранного и печатного представления.

Журналы документов предназначены лишь для удобного отображения списков документов - документ не связан «жестко» с каким-либо журналом. Набор документов, отображаемых в том или ином журнале, полностью определяется конфигурацией. Один и тот же вид документа может отражаться в нескольких журналах. В системе 1С: Предприятие нет заранее заданного распределения документов по журналам. В процессе создания конфигурации задачи можно определить для каждого вида документов наименование

журнала, в котором будет осуществляться работа с этими документами. Для конкретных журналов можно указать перечень видов документов, которые должны в нем отображаться.

Могут быть созданы также журналы, в которых отображаются все документы с возможностью их отбора по тому или иному признаку. Таким образом, состав видов документов, входящих в журнал, определяется в конфигурации в соответствии с назначением журнала.

Если в конфигурации не задано иного, по умолчанию журналы документов можно открыть, используя пункт «Операции - Документы» главного меню программы. На экран будет вызван список журналов документов, существующих в системе.

В этом списке клавишами или мышью следует выбрать наименование нужного журнала и затем нажать кнопку «ОК». Окно с выбранным журналом будет выдано на экран. Каждая строка журнала - ссылка на документ. Вид окна того или иного журнала полностью зависит от настройки конфигурации задачи. Вид журнала, выдаваемый по умолчанию, то есть в том случае, когда иное не задано при настройке представляет собой таблицу, имеющую три (колонки): дата, вид документа, номер документа. Внешний вид журнала, его свойства, поведение управляющих элементов определяются при настройке конфигурации задачи и, в общем случае, может быть самым разнообразным. Форма журнала может состоять из совокупности управляющих элементов - полей ввода, кнопок, списков, полей со списками, флажков, являющихся стандартными элементами управления Windows. Для просмотра журналов документов используются общие приемы работы с табличной частью формы. При большом количестве документов, находящихся в журнале (особенно это касается системных журналов), просмотр журнала для поиска нужного документа может занять много времени. В этом случае бывает полезно ограничить объем просмотра записей в журнале. Интервал видимости определяет, за какой период времени будут видны документы при просмотре журнала. Текущий интервал видимости выводится в заголовке окна журнала в скобках после названия журнала. Интервал видимости устанавливается двумя способами - автоматически и вручную. Когда окно с журналом вызывается на экран, интервал видимости автоматически устанавливается таким, как определено в настройке параметров журнала. Для изменения интервала видимости записей журнала необходимо выбрать пункт «Действия - Установить интервал дат» на панели инструментов журнала.

На экран будет выдан запрос «Настройка периода». После выбора интервала необходимо нажать кнопку «ОК». Порядок расположения записей в журнале определяется командой: «Действия - Сортировка». При этом: записи в журнале упорядочены по дате; внутри даты записи упорядочены по времени документа; если в пределах одной даты несколько документов имеют одинаковое время, они упорядочены по порядку их ввода в систему. Если необходимо расположить документы в журнале в порядке их ввода в программу, необходимо отключить сортировку по дате. Набор колонок в журнале можно изменить с помощью команды «Действия - Настройка списка», оставив галочки в колонках, необходимых для отображения. Поиск в журнале осуществляется путем отбора данных. Механизм отбора документов предоставляет пользователю возможность просматривать в журнале документы, отобранные по некоторому значению их реквизитов или по видам документов: например, можно отобразить все документы по конкретному складу или конкретной организации-контрагенту, или отобразить все приходные накладные.

В системе «1С: Предприятие» предусмотрены следующие виды отбора: отбор по текущему значению, отбор по виду, отбор в интервале дат, принудительный отбор.

Выбор вида отбора осуществляется командой: «Действия - Отбор». Принудительный отбор устанавливается на этапе конфигурации для ограничения прав пользователя. Например, при принудительном отборе, пользователь может видеть только те документы, которые он создал сам.

Контрольные вопросы:

- 1. Как связаны журналы документов и документы?
- 2. Опишите структуру и состав журнала документов.
- 3. От чего зависит экранный вид журнала документов?
- 4. Что такое «интервал видимости»?
- 5. Сколько табличных частей может содержать документ в системе 1С: Предприятие 8.1?

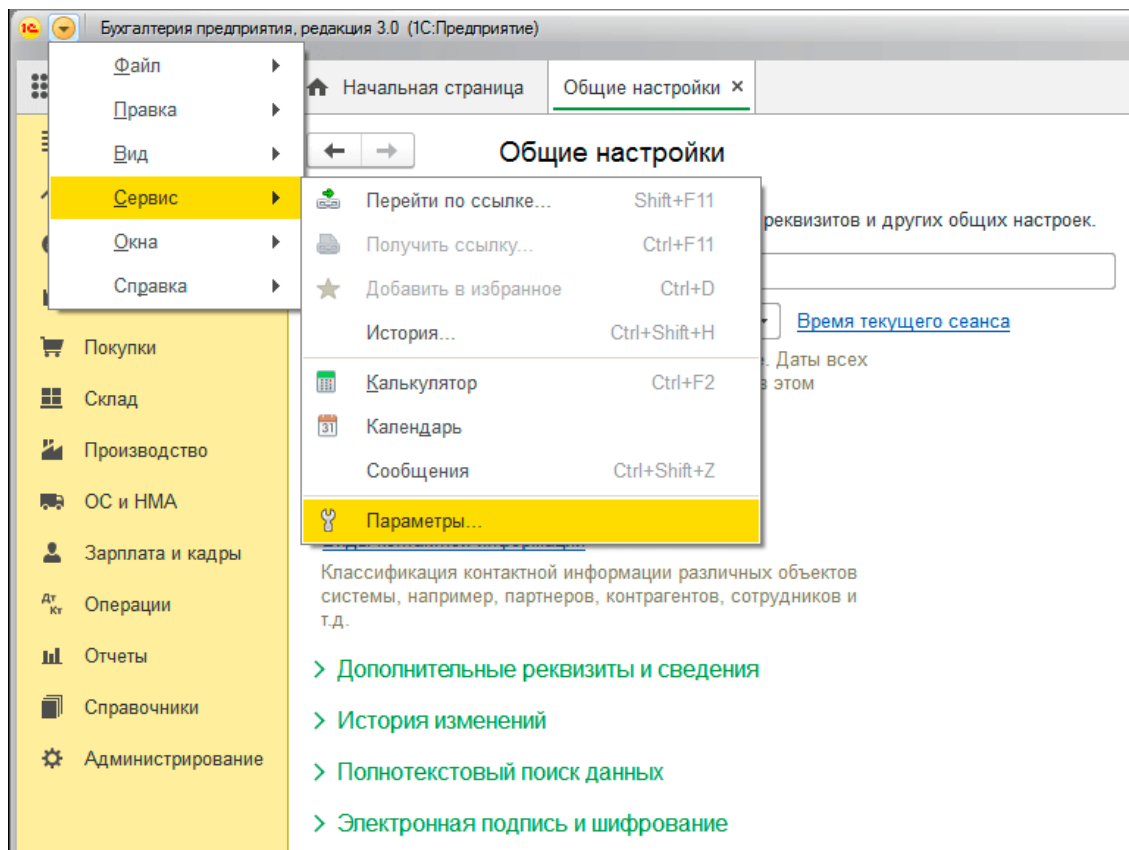
Лекция 14: Составление и проведение документов

Проведение документов, перепроведение документов и восстановление последовательности проведения документов в 1С 8.3 – очень важная часть работы с программами 1С, с которой следует быть знакомым каждому пользователю. Что же это такое и для чего нужно?

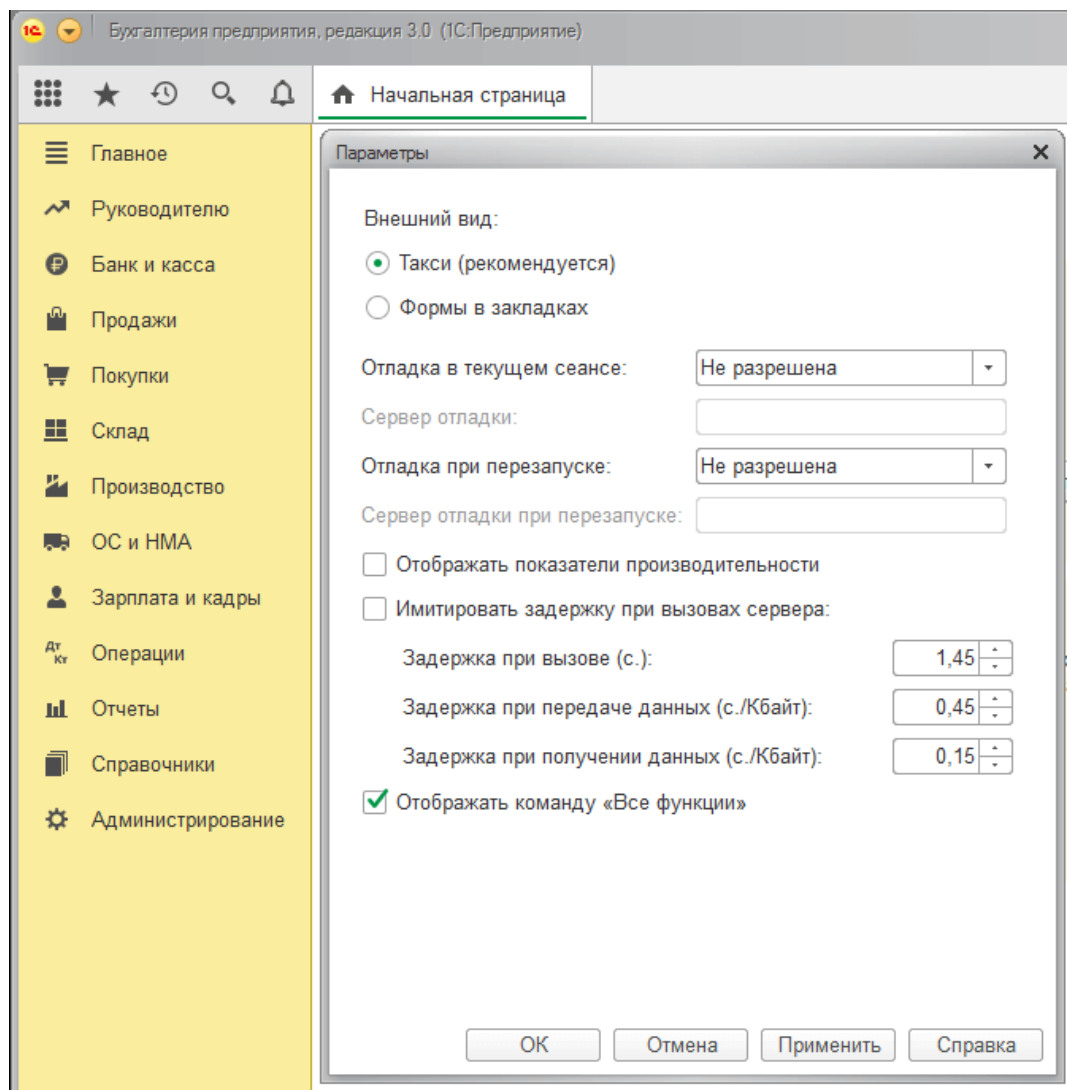
При изменении документов прошедших периодов, либо создании документов в неверной календарной последовательности могут происходить различного рода нарушения и ошибки. Продали позавчера то, что купили завтра. Оплатили аванс три недели назад, хотя выписали нам его только сегодня. Подобных примеров можно привести целое множество, но главное, что их объединяет – непоследовательное проведение документов.

Для исправления этих ситуаций 1С содержит несколько различных механизмов, позволяющих в пакетном режиме последовательно перепровести документы. Рассмотрим самые основные, позволяющие решить эту проблему в большинстве случаев.

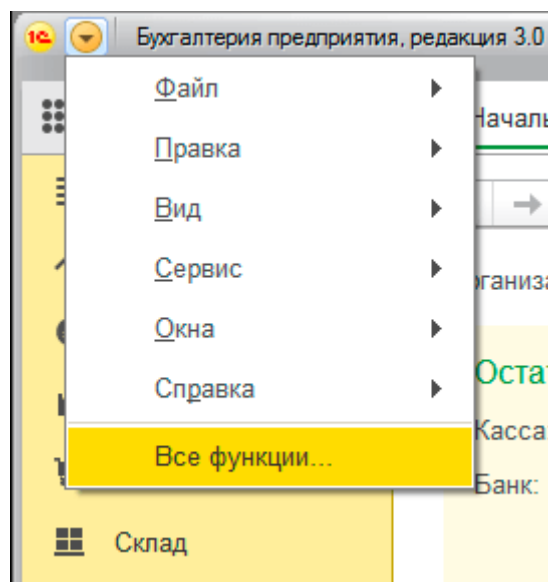
Для доступа к обработкам по проведению документов необходимо включить пункт меню «Все функции». Для этого откройте меню и нажмите Сервис – Параметры.



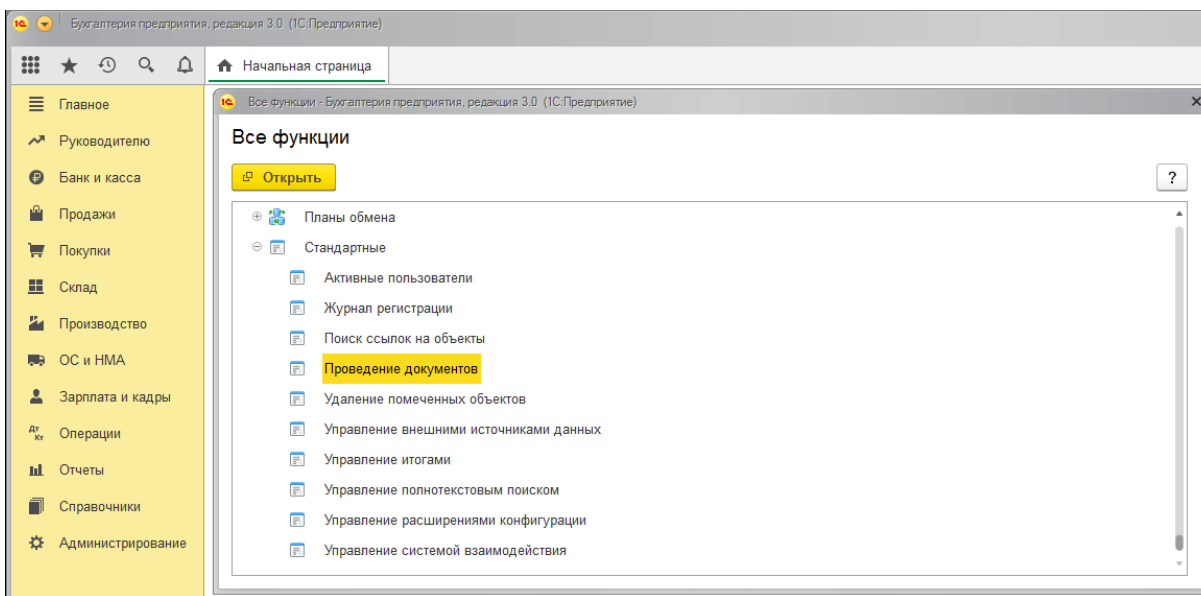
В открывшемся окне поставьте галочку «Отображать команду Все функции»:



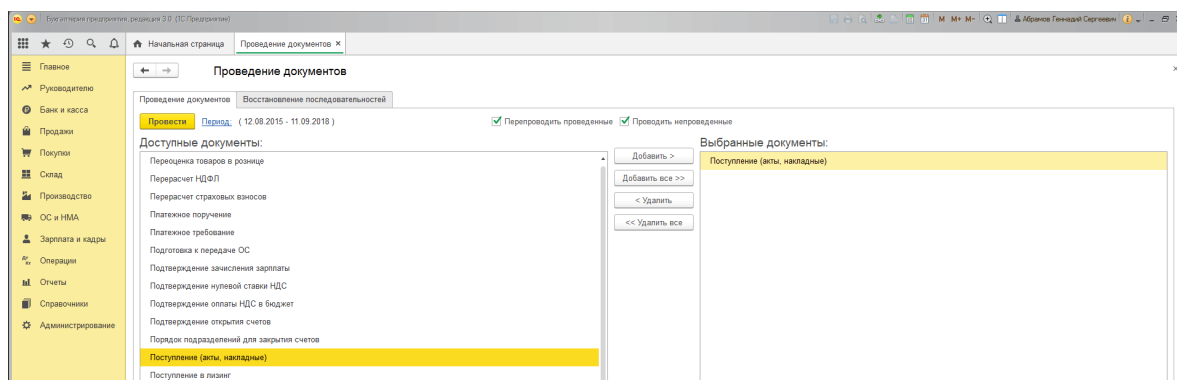
Закройте окно параметров и откройте главное меню ещё раз. В нём должен отдельным пунктом появиться «Все функции...». Нажмите на него:



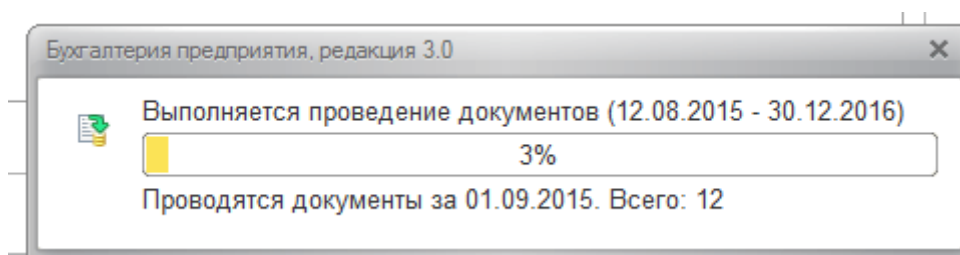
В открывшемся окне найдите список стандартных механизмов, а среди них – Проведение документов. Он будет рядом с Журналом регистрации, работу которого мы рассматривали в недавней статье. Откройте Проведение документов:



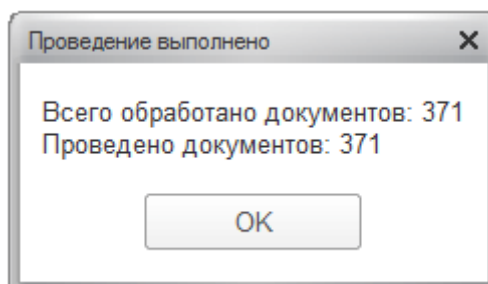
В представленном окне доступен функционал проведения документов. В левой половине отображены все типы документов, доступных к перепроведению. Рядом с кнопкой Провести также доступен фильтр по периоду, в котором можно указать, за какой период проводить документы. Выбирайте необходимые документы и переносите их в правую половину окна двойным нажатием мыши или с помощью кнопки Добавить > посередине. После этого нажмите Провести:



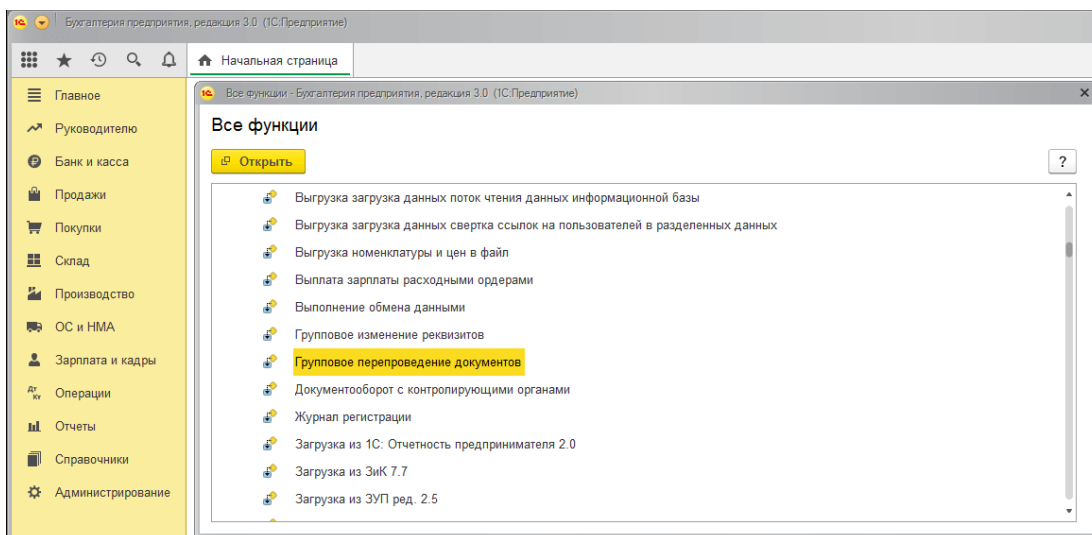
Данное окно будет информировать нас о прогрессе выполняющихся операций:



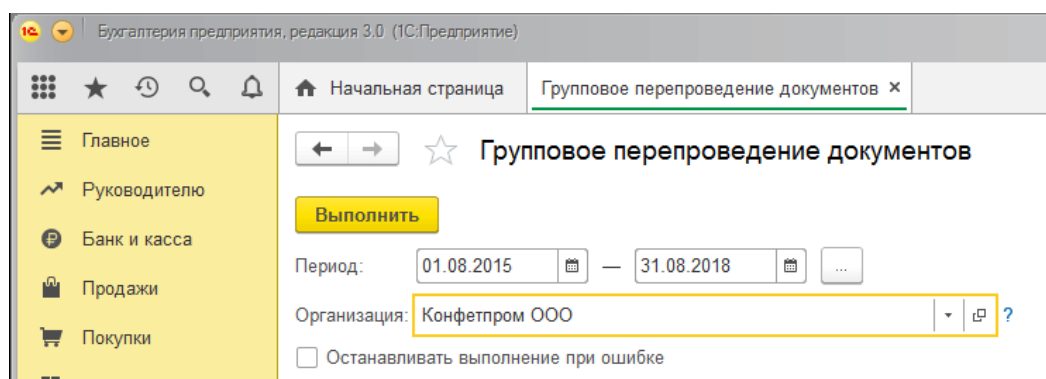
После завершения программа выдаст окно с коротким отчетом:



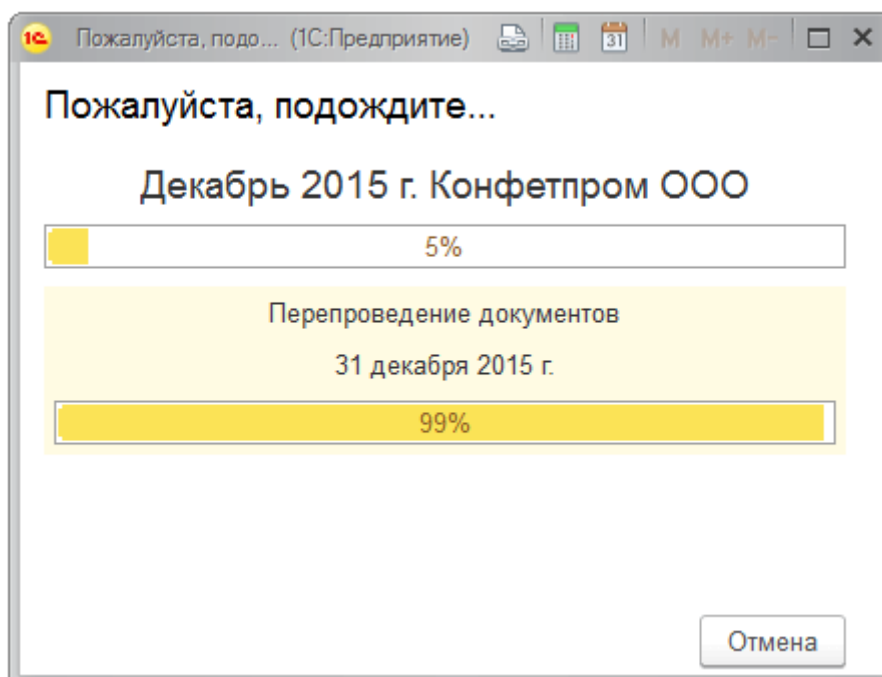
Другой вариант проведения документов – с помощью встроенной обработки. Как и в п. 3, откройте меню Все функции. Выберите в списке обработок «Групповое перепроведение документов»:



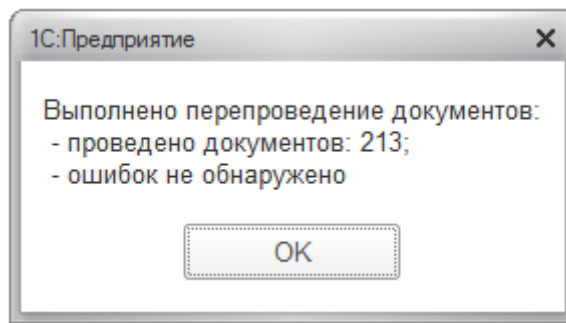
В открывшемся окне укажите период, выберите организацию и нажмите Выполнить:



О прогрессе выполнения нам будет докладывать похожее окно:



После завершения перепроведения документов будет показан аналогичный отчет об операции:



Мы рассмотрели два варианта группового проведения документов в 1С 8.3. Их возможностей достаточно для исправления нарушенной последовательности проведения документов.

Напоминаем, что в рамках механизма Закрытие месяца всегда выполняется контроль последовательности проведения документов, и в случае его нарушения автоматически запускается операция перепроведения документов.

Лекция 15: Отчеты

Возможность пользоваться в программе 1С историей хозяйственных операций для формирования итоговых отчетов и планирования показателей – одно из главных достоинств программы 1С и одно из главных условий современного бизнеса. Ниже приведен список отчетов, благодаря которым каждая коммерческая компания пользующаяся программой 1С "Управление торговлей" держит руку на пульсе происходящих дел. Специалисты нашего проекта решили поделиться практическим опытом, и рассказать пользователям о самых востребованных отчетах 1С и их предназначении. Ниже приведены виды отчетов, краткая характеристика и ссылка на инструкцию с подробным описанием настройки по формированию отчета (ссылка есть на все отчеты). Впрочем, все кто знает для чего нужен тот или иной отчет, но "плавает" в методах построения может сразу ознакомиться с готовыми инструкциями по настройке отчетов в рубрике "Аналитка" нашей Wiki-1С энциклопедии FAQ 1С. 1.

Продажи и результаты деятельности

· отчет по продажам

самый главный и самый востребованный отчет всех коммерческих компаний. Показывает оперативные показатели коммерческой активности компании на любую дату, любой период времени. Показатели и настройки отчета очень разнообразны и гибки, удовлетворят даже самого требовательного управленца. Возможно формирование отчета по: покупателям, по номенклатуре, по менеджерам, по количеству и сумме (убывание/возрастание), с группировками по характеристикам продукции (размер, цвет, производитель, сорт) с отбором по определенному товару, признаку и т.д.

· валовая прибыль

указатели сравнения сумм продаж накладных с их себестоимостью, что позволяет оперативно формировать показатели рентабельности работы компании. Так же как отчет по продажам, отчет по валовой прибыли настраивается с разными параметрами и показателями. Полезен для руководителей подразделений продаж, маркетологов, экономистов.

· рапорт руководителю

предназначен для предоставления руководителю сводной информации о текущем состоянии дел компании. В отчет могут быть включены произвольное количество данных, но как правило используется несколько ключевых показателей для простоты восприятия оперативных показателей: дебиторская (кредиторская) задолженности, продажи, закупки, остатки, валовая прибыль, рентабельность, период оборота товарных запасов. Для удобства наглядности данных предусмотрено графическое отображения показателей в виде графиков, диаграмм. Если отчет по продажам можно назвать основным отчетом руководителей отделов продаж, то рапорт руководителю – основной отчет руководителя компании.

• **авс-анализ продаж**

отчет выполняется на основании отчета о продажах. В отчет заложен принцип Парето (80 % продаж делает 20 % ассортимента продукции). Этот отчет классифицирует весь товар на самую продаваемую продукцию и неходовую продукцию. Универсальность этого отчета позволяет проводить авс-классификацию практически любых показателей деятельности компании. Необходим для маркетологов, руководителей, продавцов.

• **хуз-анализ покупателей**

отчет предназначен для анализа стабильности. Для каждого из 3-х классов (X, Y и Z) задаются свои значения – относительные коэффициенты вариации параметра. Считается что класс X объекты с наиболее стабильными параметрами, а класс Z – с самыми нестабильными параметрами.

• **план-фактный анализ продаж**

любая уважающая себя компания планирует свою деятельность, в том числе и продажи. Планфактный отчет по продажам как нельзя лучше отражает успеваемость фирмы согласно запланированным показателям по продажам. Необходимым условием эффективного использования данного отчета является ведение в базе данных плановых показателей продаж. Отчет необходим руководителям торговых подразделений, ответственным менеджерам, директорам, экономистам. процент выполнения/невыполнения заказов в торговой деятельности существует практика оформления предварительной договоренности с клиентом о покупке продукции. Как правило, такие договоренности оформляются заказом покупателя. Если вас интересует, как формируется воронка продаж вашей компании, то просто необходимо пользоваться отчетом о проценте выполнения/невыполнения заказов покупателей. Причиной невыполнения (потери) клиента могут быть: отказ клиента, отсутствие товара, отсутствие оплаты и т.д. Анализ заказов покупателей ответит на все вопросы по заказам.

• **отчет по товарам, переданным на комиссию**

немногие компании работают по схеме передачи товара на комиссию, но если работают, то часто возникают вопросы о количестве переданного, проданного и оплаченного товара комиссионером. Отчет по работе с комиссией предназначен для формирования ключевых показателей работы с комиссионером, отражает состояние дел по комиссии, задолженность за товар и т.д. отчет по клиентам (контактные данные, экспорт данных) иногда возникает потребность использовать базу данных контактов клиентов для формирования почтовой рассылки, звонков, поздравлений и т.д. Существует возможность извлечения таких данных для использования их в формате Excel. С помощью фильтров и отборов в приложениях Microsoft Office возможна дальнейшая обработка, фильтрация данных для дальнейшей работы.

• **показатели работы менеджеров**

отчет показывает эффективность работы менеджеров по продажам. Например, если у вас за каждым клиентом закреплен ответственный менеджер, вы с легкостью можете проанализировать какую выручку в этом месяце принес конкретный менеджер. По сути этот тот же отчет по продажам, но с возможностью аналитики эффективности торговых сотрудников компании.

2. **Остатки и склад**

• **отчет по срокам хранения продукции**

показывает количество времени хранения товарных по интервалам времени на складах компании. Интервалы могут быть произвольными, но обычно это промежутки: до 30 дней, от 30 до 60 и более. Отчет необходим менеджерам по логистике, закупкам и сбыту. Отчет наглядно показывает самый быстрооборачиваемый товар и товар с большим сроком хранения (угроза неликвида).

• **анализ точки заказа**

этот отчет поистине можно назвать сердцем логистики. Благодаря данному отчету анализируется текущий остаток товаров с учетом невыполненных сделок, сравнивается с точкой минимально допустимого порога остатков номенклатурной позиции на остатке. Если порог точки заказа достигнут, отчет формирует потребность в заказе продукции для пополнения товарных запасов. Точка заказа конкретной продукции для каждой номенклатуры задается индивидуально. Возможен учет сезонности и всплесков.

- анализ закупок. План-факт выполнения закупок

отчет предназначен для понимания выполнения обещанных поставок поставщиками и реальными поставками. Недовыполнения поставок, как вы понимаете, может привести к потере продаж, а излишняя загруженность товарными запасами к наращиванию на остатках неликвидов, порче продукции, увеличению потребности складских площадей и т.д. К тому же будет полезно знать, кто из поставщиков добросовестно выполняет свои обязательства, а кто нет.

- **анализ свободного товара на складе**

как известно весь товарный остаток организации делится на три составляющие: общий, зарезервированный и свободный остаток. Понимание состояния товарных запасов именно в таком разрезе необходимо, если вы намерены вести качественные отношения с клиентами, не досадуя покупателей фразами «товар, который вы заказывали вчера нет, т.к. он отложен для другого». Работа только со свободным остатком через отчет качественно повысит работу с клиентом, сделает ваши отношения с покупателями более доверительными и надежными.

- **анализ складских остатков**

простой отчет для оценки количества остатков товаров на складах компании в разрезе номенклатуры, складов, единиц измерения и т.д. Полезен, например, для отслеживания динамики роста или снижения складских запасов, занимаемым площадям, аналитики размера запасов в разные периоды времени. ведомость по товарам на складах отчет очень похож на предыдущий отчет, с тем отличием, что ведомость по товарам на складах дает остатки, но еще и с причинно-образующей аналитикой. Т.е. отчет отвечает на вопрос почему, каким документом или благодаря какому поставщику увеличились складские остатки. Очень удобен для поиска ошибок управленческого учета и аналитики движений товара в количественных показателях

- **остатки на складах в ценах себестоимости (закупочных ценах)**

отчет интересен тем, кого кроме количественных показателей остатков интересуют стоимостные показатели в закупочных ценах поставщика. С помощью этого отчета также можно узнать себестоимость любого вида продукции и посмотреть партиообразующий документ (каким документов оприходован в программу)

3. НДС и налоги

- **реестр налоговых накладных**

отчет предназначен для формирования реестра налоговых накладных за период и выгрузку в xml формате для дальнейшей загрузки в другие программы налоговой отчетности.

- **проверка суммы обязательств по НДС**

формирует суммы выписанных и не выписанных налоговых обязательств за период. Необходим бухгалтерам для ориентации в выписке налоговых документов, т.к. не всегда понятно какому клиенту, когда именно и была ли выписана налоговая накладная, по предоплате или по отгрузке и т.д.

4. Задолженность, деньги

- **отчет о дебиторской задолженности**

расширенный отчет о дебиторской задолженности по клиентам, договорам и даже по документам движения. Удобен тем, что позволяет формировать взаиморасчеты по клиентам входящим в один холдинг. Т.е. если клиенты скреплены в программе главным юр. или физ. лицом, то можно брать взаиморасчеты по главному лицу, получая таким образом сгруппированные данные о задолженности. Отчет необходим бухгалтерам, финансистам, экономистам. платежный календарь показывает ежедневное, оперативное движение денежных средств. Показывает сумму входящих платежей, а исходя из этого, позволяет планировать расходы, позволяет избежать кассовых разрывов.

- **БДР**

Отчет о доходах и расходах. Выполняется на основании бюджета движения денежных средств (БДДС). Дает возможность сделать аналитику всех видов деятельности компании. Наглядно можно рассмотреть все виды деятельности: хозяйственная, финансовая, инвестиционная. Итогом этого отчета считается отчет о «Прибыли и убытках» показывающий в целом результаты деятельности компании за квартал, год.

• анализ доступности денежных средств

отчет о наличии наличных и безналичных денежных средств на расчетных счетах и кассах компании. Необходим для планирования платежей и среза остатка денежных средств компании. Отчет понадобится для планирования платежей и поступлений денежных средств для финансистов, бухгалтеров, экономистов. Может быть сформирован оперативно на любую дату в валюте управленческого и регламентного учета.

• анализ движения денежных средств

отчет предназначен для аналитики движения денежных средств по статьям. Статьи вносятся в программу произвольно и зависят от внутренней финансовой архитектуры компании. Отчет необходим об аналитике расходных и доходных статей затрат для финансистов, управленцев.

5. Ценообразование

• анализ цен

предназначен для анализа изменения закупочных цен поставщиков, если таких ведется регистрация в программе. Необходим для менеджеров по закупкам, специалистов по ценам, учетникам. Позволяет сравнивать закупочные цены на продукцию между поставщиками для выбора наиболее выгодных.

• прайс-листы

отчет о текущих ценах – инструмент, который есть в наличии практически у каждого менеджера по продажам. Цены – это главное. Невозможно продать что-либо не назвав цены. Возможны настройки прайс-листа с разными группировками и отборами для более компактного вида, т.к. прайс-листы торговых компаний имеют зачастую большой ассортимент продукции

• ценники

самый маленький отчет для розничной торговой точки. Фактически это тот же прайс-лист, но предназначенный для каждой номенклатурной позиции отдельно с возможностью компактного вывода на печать для дальнейшего размещения на торговых стеллажах магазинов, салонов, складов и т.д.

Лекция 16: Проведение сформированного отчета

Конечная цель любой учетной системы – формирование отчетов. 1С:Предприятие 8.2 предоставляет разработчику множество инструментов для *создания отчетов* – от достаточно простых механизмов, позволяющих создавать несложные отчеты, до комплексных средств, таких, как *система компоновки данных*. Сейчас мы рассмотрим пример создания простого отчета. Нам нужно вывести отчет в виде списка контрагентов *по* группам с указанием наименования контрагента, основного контактного лица и телефона этого контактного лица.

Отчет нужного нам вида можно сформировать различными способами. Так, вполне можно реализовать эту функциональность непосредственно в справочнике **Контрагенты**, добавив в него соответствующие программные *механизмы*, выведя кнопку "**Сформировать список контрагентов**" в форму списка справочника. Можно сделать это с помощью специализированного прикладного объекта **Отчет**. Обычно для создания подобного рода отчетов так и поступают.

Отчет, в нашем случае, будет строиться на основе макета, с областями которого работают в программном коде, формируя готовый отчет. Как правило, если речь идет о неких общих отчетах, их создают в виде отдельных объектов, если же, например, нам нужно создать печатную форму для отдельного элемента справочника или отдельного документа – вполне можно "поместить" всю функциональность требуемого отчета внутри прикладного объекта. В частности, прикладные объекты могут иметь в числе подчиненных объектов макеты, которые и используются при создании отчетов. В то же время, внешний отчет вполне можно использовать и для создания *печатной формы* отдельного элемента справочника или документа.

Создадим в ветви *дерева конфигурации* **Отчеты** новый отчет, дадим ему имя **СписокКонтрагентов**, рис. 5.13.

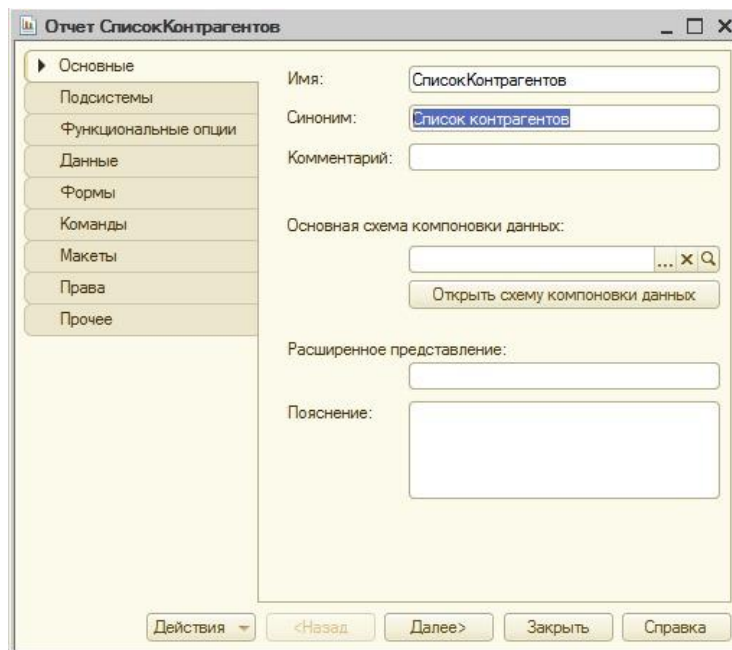


Рис. 5.13. Создание нового отчета

Первым этапом работы над отчетом станет создание макета отчета. *Макет* позволяет заранее определить и оформить "блоки", из которых будет построен отчет.

Следует отметить, что во всех возможных случаях при разработке *прикладных решений* для 1С:Предприятие 8.2. следует создавать их на основе схемы компоновки данных. Однако умение работать с макетами в форме табличных документов может пригодиться в том случае, если вам понадобится отредактировать сторонний отчет, выполненный в таком стиле.

Перейдем на закладку формы редактирования объекта **Макеты** и нажмем на кнопку **Добавить**. Появится окно конструктора макета, где нам предложат задать его имя (оставим имя *по умолчанию* – **Макет**), и тип макета – нас устроит **Табличный документ**, рис. 5.14.

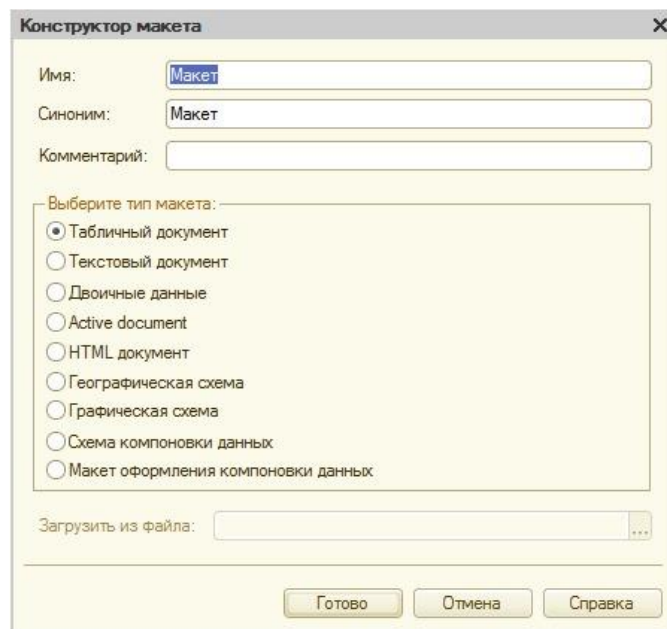


Рис. 5.14. Создание макета для отчета

После нажатия на кнопку **Готово**, мы видим табличный редактор, рис. 5.15., очень напоминающий Microsoft Excel. Работая с ним, мы можем пользоваться стандартной палитрой свойств, а так же – панелями инструментов, в частности – **Форматирование**, **Табличный документ**, **Имена**. Наша задача сейчас – создать и отформатировать области, которые позже будут использованы для формирования готового отчета.

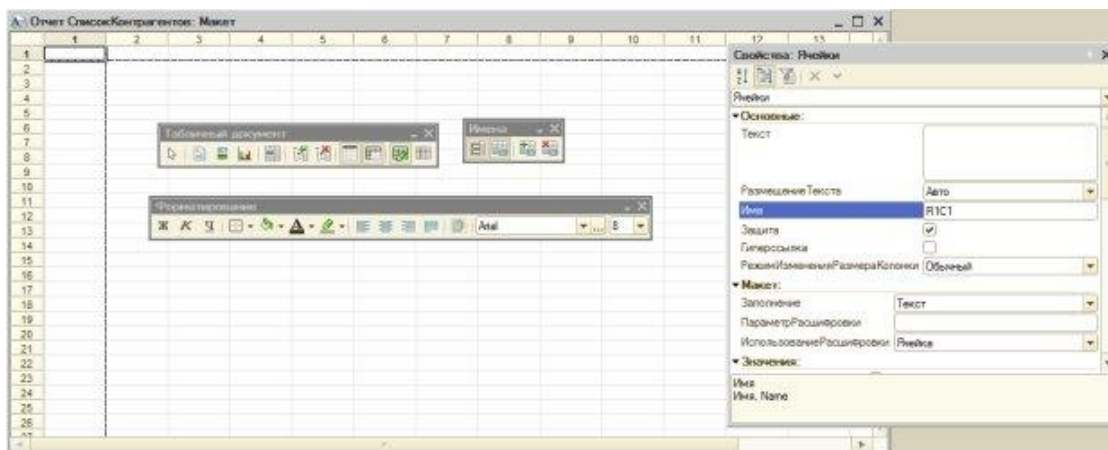


Рис. 5.15. Средства редактирования макета отчета

При создании макета мы можем вводить в ячейки обычный текст – такой текст отображается в ячейке без каких-либо дополнительных знаков. *Ячейка* может содержать именованный *параметр*, который будет заполнен при формировании отчета. Так же ячейки могут содержать шаблоны, состоящие из обычного текста и параметров, которые так же можно заполнить.

На рис. 5.16. показан готовый макет.

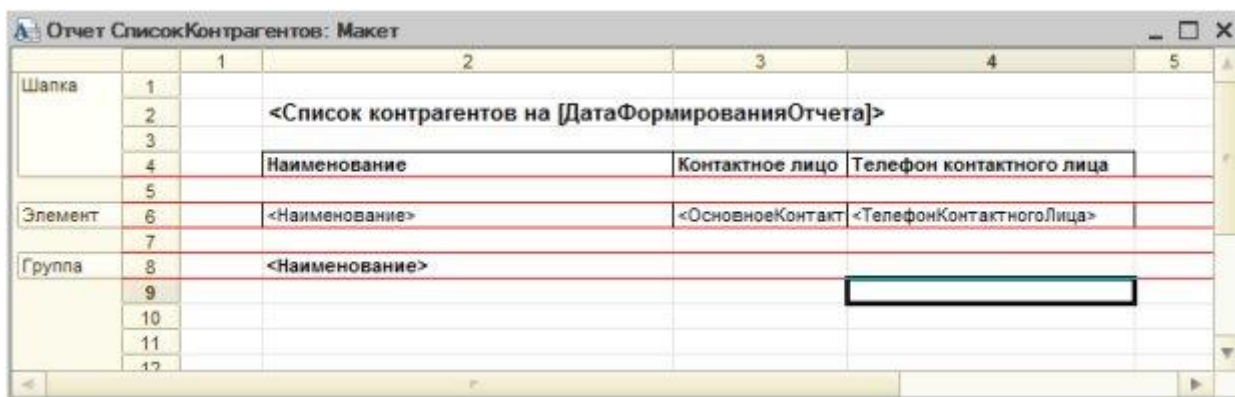


Рис. 5.16. Готовый макет отчета

Ячейка 2,2 заполнена следующим образом: в нее сначала введен текст "Список контрагентов на [ДатаФормированияОтчета]", после чего вызвано окно свойств этой ячейки, в которых, в свойстве **Заполнение** выбрано **Шаблон**, рис. 5.17.

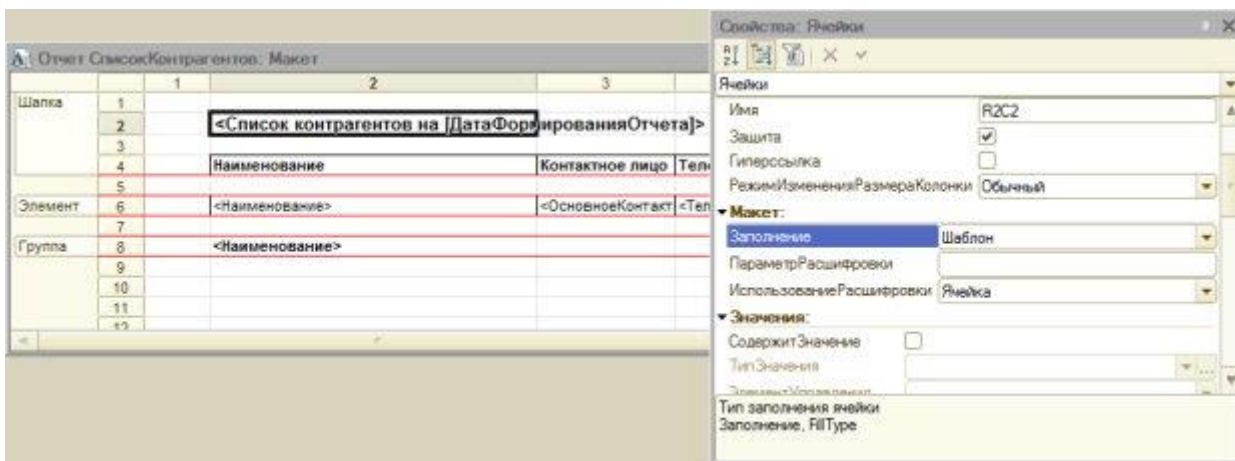


Рис. 5.17. Настройка ячейки, содержащей шаблон

Параметр ДатаФормированияОтчета мы установим в текущую дату программно при формировании отчета.

Ячейки с 4,2 по 4,4 содержат обычный текст – он будет выводиться в качестве шапки таблицы.

И заголовок отчета и шапка таблицы объединены в область с именем **Шапка**. Для задания имени области достаточно выделить нужные ячейки (выделять нужно *по* заголовкам строк) и отредактировать в палитре свойств *параметр Имя выделенного диапазона*, или воспользоваться кнопкой **Назначить имя** панели инструментов **Имена**.

Область **Элемент** содержит три параметра – **Наименование**, **ОсновноеКонтактноеЛицо** и **ТелефонКонтактногоЛица**. После ввода в каждую из ячеек имен параметров, нужно выделить их (все вместе или *по* одной) и в окне свойств в *поле Заполнение* указать **Параметр**. К тексту в ячейках будут автоматически добавлены угловые скобки (<>), что позволяет визуально определить наличие в ячейке параметра.

Область **Группа** содержит лишь *параметр Наименование*.

Обратите внимание на имена параметров – они соответствуют именам реквизитов справочника, которыми мы собираемся их заполнять.

Ячейки в шаблоне можно форматировать – задавать их границы, оформление текста, выравнивание и т.д.

Теперь приступим к созданию формы отчета. Перейдем на вкладку **Формы** окна редактирования объекта, добавим новую форму отчета, оставим все настройки в состоянии *по* умолчанию и нажмем **Готово**. Добавим, на вкладке **Реквизиты** редактора форм новый *реквизит*, назовем его **ТабличныйДокумент**, выберем для него тип **ТабличныйДокумент**. Перетащим созданный *реквизит* в *поле Элементы*. В состав команд формы добавим новую команду, зададим ей имя **СформироватьОтчет** и так же переместим в *поле Элементы*. В итоге у нас получится форма, выглядящая так, как показано на рис. 5.18.

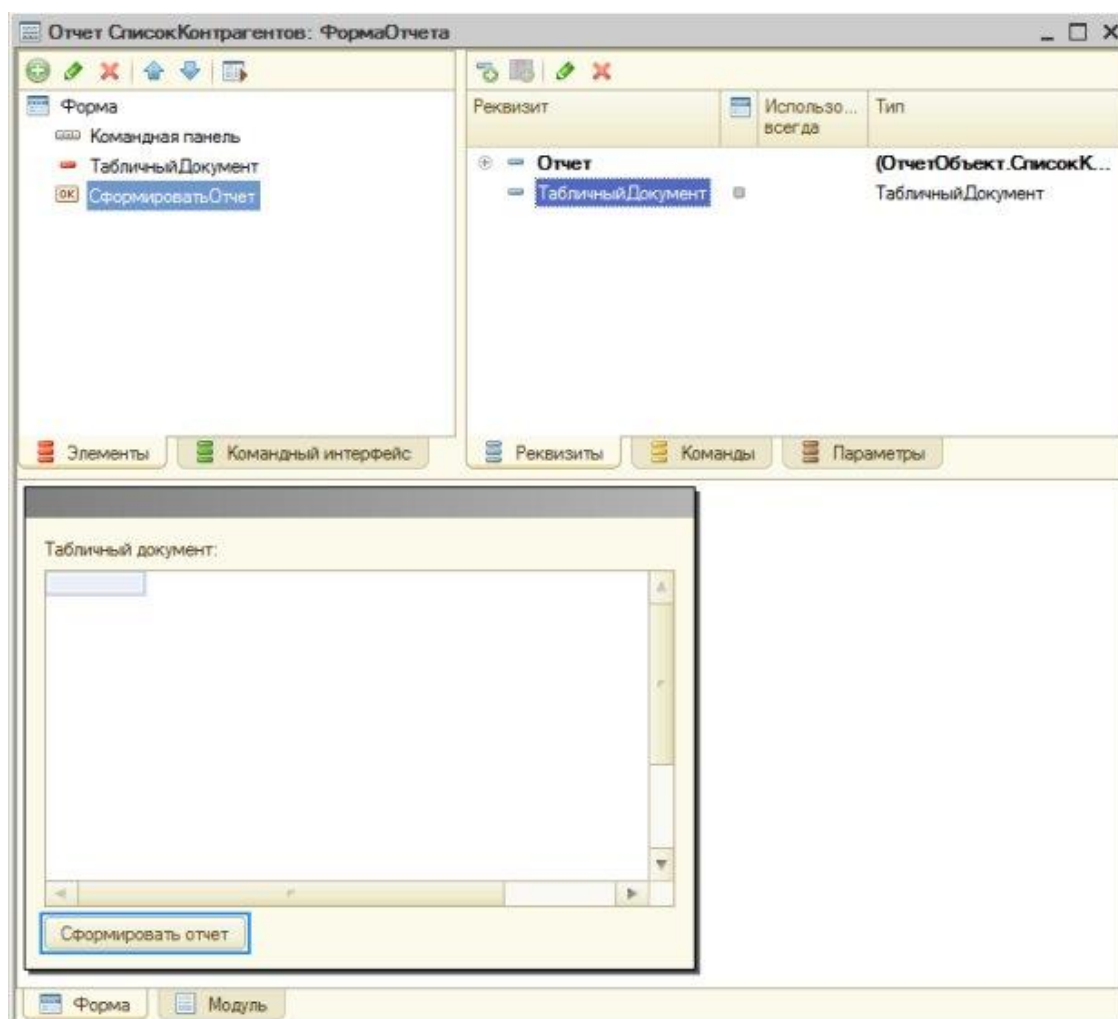


Рис. 5.18. Настройка формы отчета

Для построения отчета мы должны будем получать данные из справочника. Это можно сделать с помощью уже знакомого вам объекта **СправочникВыборка**, можно получить данные с помощью запроса. В любом случае, это предусматривает работу с базой данных, то есть, нам понадобится процедура, выполняемая на сервере. До настоящего времени мы пользовались лишь директивой компиляции **&НаСервере** – при вызове методов, объявленных с этой директивой, мы имеем *доступ* к контексту формы, при этом между клиентом и сервером происходит передача дополнительных данных – как при вызове серверного метода с клиента на сервер, так и в обратном направлении. Сейчас мы воспользуемся серверным внеконтекстным методом для формирования отчета. В этом методе мы собираемся формировать табличный документ, содержащий данные отчета.

При реализации метода в виде процедуры, нам придется передать в него в качестве параметра наш *реквизит* **ТабличныйДокумент**. По умолчанию параметры передаются *по ссылке*, то есть, работать процедура будет непосредственно с нашим реквизитом.

При реализации метода в виде функции мы можем ничего не передавать в него, сформировать внутри функции табличный документ и вернуть уже заполненный документ в точку вызова, присвоив его нашему реквизиту **ТабличныйДокумент**.

Реализуем метод в виде функции. Готовый код формирования отчета (рис. 5.19) будет выглядеть следующим образом:

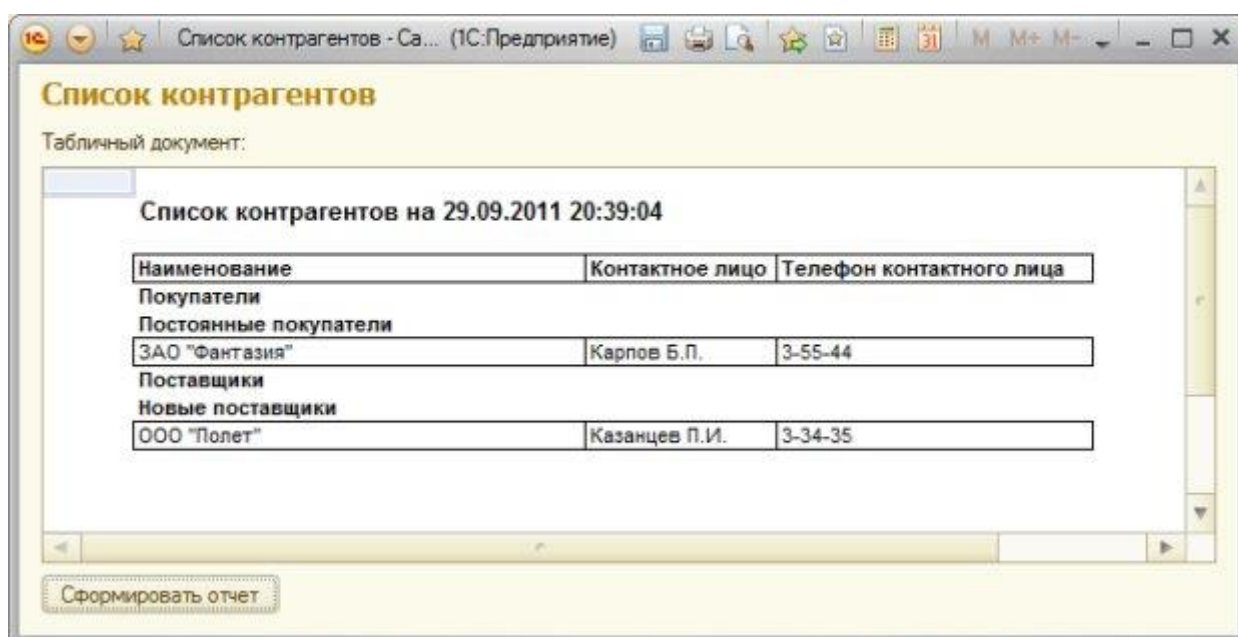


Рис. 5.19. Готовый отчет

&НаКлиенте

Процедура СформироватьОтчет(Команда)

ТабличныйДокумент=СформироватьОтчетНаСервере());

КонецПроцедуры

&НаСервереБезКонтекста

Функция СформироватьОтчетНаСервере()

ТабличныйДокумент=Новый ТабличныйДокумент();

Макет=Отчеты.СписокКонтрагентов.ПолучитьМакет("Макет");

Шапка=Макет.ПолучитьОбласть("Шапка");

Элемент=Макет.ПолучитьОбласть("Элемент");

Группа=Макет.ПолучитьОбласть("Группа");

Шапка.Параметры.ДатаФормированияОтчета=ТекущаяДата();

ТабличныйДокумент.Вывести(Шапка);

Выборка=Справочники.Контрагенты.ВыбратьИерархически();

Пока Выборка.Следующий() Цикл

Если Выборка.ЭтоГруппа Тогда
Область=Группа;
Иначе
Область=Элемент;
КонецЕсли;
Область.Параметры.Заполнить(Выборка);
ТабличныйДокумент.Вывести(Область);
КонецЦикла;
Возврат (ТабличныйДокумент);
КонецФункции

В клиентской процедуре **СформироватьОтчет()** мы вызываем серверную внеконтекстную функцию **СформироватьОтчетНаСервере()**, присваивая возвращаемое ей значение реквизиту **ТабличныйДокумент**.

В серверной функции мы создаем новую переменную с типом **ТабличныйДокумент** и именем **ТабличныйДокумент**. Именно в него мы будем выводить данные и именно его будем возвращать в точку вызова. Несмотря на то, что имя реквизита формы и имя данной переменной совпадают, между ними нет никакой связи. Это – отдельные объекты.

Далее, мы получаем *макет* из нашего отчета, пользуясь методом **ПолучитьМакет()** и задавая имя макета. Отчет может иметь несколько макетов, их выбор осуществляется *по* имени.

В переменную **Шапка** мы записываем область макета **Шапка**, соответственно поступаем с переменными **Элемент** и **Группа**.

Командой **Шапка.Параметры.ДатаФормированияОтчета=ТекущаяДата();** мы заполняем ранее заданный в макете *параметр* **ДатаФормированияОтчета**, записав в него текущую дату. Дата, возвращаемая функцией **ТекущаяДата**, содержит, помимо года, месяца и дня, так же часы, минуты и секунды. При необходимости дату перед выводом можно отформатировать при помощи функции **Формат()**.

После того, как *параметр*, находящийся в шапке, заполнен, мы можем вывести шапку в табличный документ командой **ТабличныйДокумент.Вывести(Шапка);**

Следующим этапом нашей работы будет получение иерархической выборки справочника. Такая *выборка* позволяет получить элементы и группы справочника в иерархическом порядке, учитывая родительские отношения между элементами.

В цикле обхода выборки мы сначала проверяем, является ли текущий элемент справочника группой. Если является, присваиваем переменной **Область** ранее полученную область отчета **Группа**. Если не является – присваиваем ей область **Элемент**.

Благодаря совпадению имен параметров и имен реквизитов справочника, для помещения данных из выборки в область макета, достаточно воспользоваться конструкцией **Область.Параметры.Заполнить(Выборка);**. После заполнения параметров мы выводим сформированную область в табличный документ.

Когда цикл перебора выборки будет завершен, мы возвращаем сформированный табличный документ в точку вызова и *пользователь* видит готовый отчет.

Для формирования простейших отчетов, *пользователь* может воспользоваться стандартной функциональностью, присутствующей в 1С:Предприятие 8. Для этого, открыв, например, *список* справочника, он может выполнить команду **Все действия > Вывести список**. Появится окно **Вывести список**, рис. 5.20, где в *поле Выводить в* можно выбрать либо **Табличный документ** (его обычно и используют), либо – **Текстовый документ**.

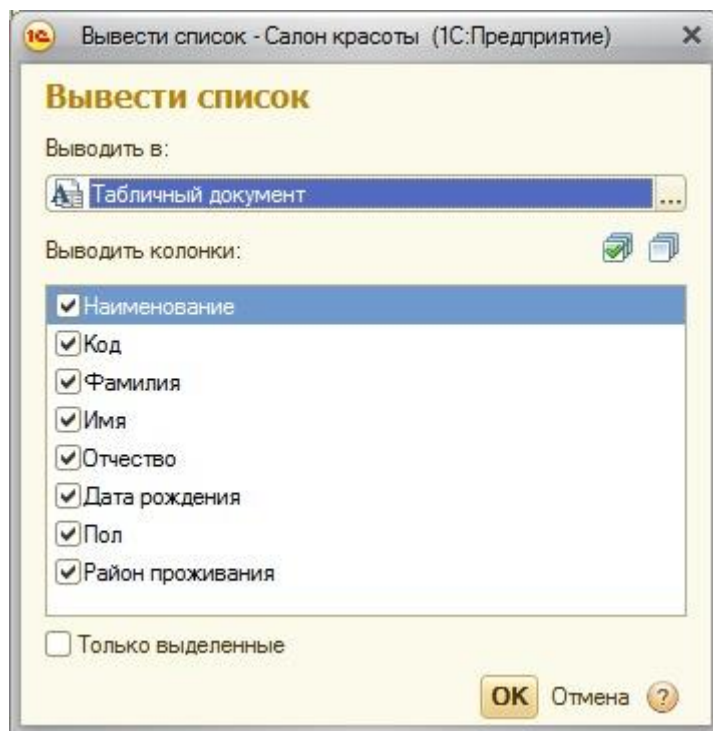


Рис. 5.20. Окно Настройка списка

В поле **Выводить колонки** можно настроить состав выводимых в документ колонок (в нашем случае команда выполнена для справочника **Физические Лица**). После нажатия на **OK** выбранные данные оформляются в виде табличного документа, а с помощью команды **Файл > Сохранить как**, рис. 5.21., этот документ можно сохранить в нужном формате для дальнейшей обработки в других приложениях.

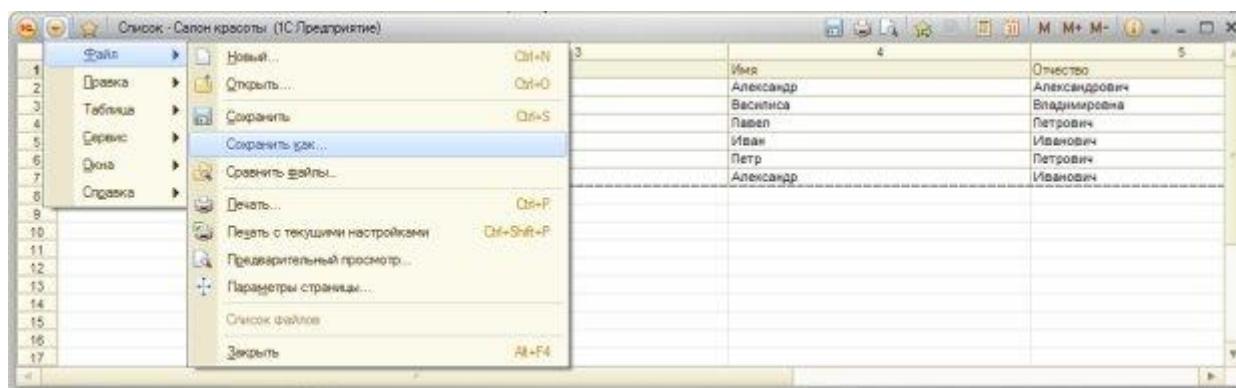


Рис. 5.21. Вывод данных в табличный документ

Лекция 17: Регистры

Регистры предназначены для хранения и обработки различной информации, отражающей хозяйственную или организационную деятельность предприятия. В них могут храниться сведения о курсах валют, котировках акций, остатках и оборотах товаров, бухгалтерские проводки и т.д. Данные в регистры могут записываться как в ручном режиме, так и при помощи процедуры проведения документов – когда поля регистра заполняются значениями реквизитов соответствующих документов.

Благодаря их использованию конфигурации «1С: Предприятие» обладают существенно лучшей производительностью. Это обеспечивается как правильным использованием регистров для хранения данных, так и наличием у них специального механизма – виртуальных таблиц.

Как основные объекты, регистры входят в состав объектной модели данных. Так же, как и другие основные объекты, регистры имеют связанные программные объекты, обеспечивающие доступ как к самим регистрам, как объектам, так и к данным регистров.

Как таблицы БД, регистры входят в состав табличной модели данных. Каждый регистр представляется отдельной таблицей. Строка таблицы регистра называется записью.

Лучше всего назначение и роль регистров видна в объектной модели данных. Пользователь осуществляет первичный ввод данных при помощи соответствующих документов. Документы могут являться регистраторами по отношению к регистрам – то есть вызывать при своем закрытии специальную процедуру проведения, которая изменяет состояние регистра. Процедура проведения добавляет в регистр новую запись, при этом в поля записи переносятся данные из реквизитов документа. Таким образом, данные попадают в регистры. Один документ может проводить осуществлять движения по нескольким регистрам и один регистр может иметь несколько регистраторов.

Измерения предназначены для идентификации записей регистра и анализа данных. Набор значений измерений для каждой записи представляет собой уникальный идентификатор в пределах регистра. Разработчик может обращаться к регистру и осуществлять отбор данных по заданным измерениям (рис.3.8). При этом каждое измерение выступает как плоскость (разрез) данных регистра, а сам регистр можно представить как n-мерный гиперкуб, где n – число измерений. Часто, например, говорят: «анализ остатков в разрезе поставщиков».

Измерения						Ресурс
N	Дата	Номенклатура	Склад	Кладовщик	Ячейка	Количество
C	01.01.01	Товар 1	Склад 1	Иванов	1	16
N	01.01.01	Товар 2	Склад 1	Иванов	4	20
M	04.01.01	Товар 1	Склад 2	Петров	2	10

Рис.3.8. К определению измерений и ресурсов

Ресурсы предназначены для хранения собственно данных. Значения ресурсов есть значимая для пользователя информация. Например, в регистре остатков ресурсы могут содержать количество и сумму.

Реквизиты в регистрах играют такую же роль, как и в других основных объектах.

После ввода данных регистр может использоваться для получения, как первичных данных, так и некоторых результатов вычислений и запросов. При этом существенную роль играют разновидности регистров и их виртуальные таблицы.

Контроль учета и документооборота осуществляется на основе значений показателей трех видов.

Показатели остатков (накопления) используются для контроля переменных данных, анализ которых необходимо проводить как по текущему значению, так и по предыдущим. Пример: количество товаров на складе, сколько денег в кармане и т.д.

Оборотные показатели (накопления) используются для контроля данных, анализ которых нужно проводить за период. При этом данные за различные периоды должны быть независимы. Пример, ВВП, оборот за месяц и т.д.

Показатели состояния используются для контроля данных, имеющих дискретные, независимые состояния. Пример: курс валют, котировки на бирже.

В «1С: Предприятия» существует 4 вида регистров.

Регистры сведений являются просто хранилищами данных – хранят сведения самого различного характера. Это могут быть курсы валют, графики работ, котировки акций, разнообразные расписания и т.д. Это единственный вид регистров, который может изменяться не только регистратором, но и в пользовательском режиме. Например, можно вводить ежедневный курс доллара к рублю документом, а можно вручную.

Регистры сведений могут быть периодическими или непериодическими. В периодических регистрах появляется предопределенное измерение период, в котором содержится дата создания записи. Это измерение должно быть уникальным в пределах периода. Периодом может быть секунда, день, месяц, квартал, год.

Регистры накоплений предназначены для хранения специфической информации об остатках и оборотах; могут изменяться только регистратором.

Регистры бухгалтерии используются для отображения операций в рамках бухгалтерского учета. Это хранилище проводок. С каждым регистром бухгалтерии связан один план счетов, счета которого используются при формировании проводок и заполнении измерений по счетам дебета и кредита.

Регистры расчета предназначены для проведения сложных периодических расчетов. Как правило, на практике это расчеты заработной платы. С каждым регистром расчетов должен быть связан план видов расчета, предназначенный для хранения видов расчета. Регистры расчета являются периодическими, разновидности периодов: день, месяц, квартал, год.

Механизм виртуальных таблиц обеспечивает быстрый доступ к специфической информации в регистре. Каждый регистр имеет собственный набор виртуальных таблиц. Виртуальные таблицы не хранятся в базе данных, а создаются по мере обращения к ним. Разработчик может использовать виртуальные таблицы в запросах или в программном коде при посредстве специальных методов регистров.

Так как виртуальные таблицы используются для повышения производительности, то их использование в соответствующих запросах является обязательным. Обращение к регистру там, где возможно обращение к виртуальной таблице является ошибкой.

Рассмотрим виртуальные таблицы регистров сведений и накопления.

Таблица *СрезПервых* возвращает записи регистра сведений, наиболее близкие в прошлом к заданной в параметре таблицы дате.

Таблица *СрезПоследних* возвращает записи регистра сведений, наиболее близкие в будущем к заданной в параметре таблицы дате.

Таблица *Остатки* возвращает остатки по регистру накоплений на заданную дату. Используется только в регистре накоплений остатков. В качестве параметров передаются границы временного диапазона.

Таблица *Обороты* возвращает обороты по регистру накоплений на заданную дату. Используется только в регистре накоплений оборотов. В качестве параметров передаются границы временного диапазона.

Каждому регистру объектной модели данных соответствует набор программных объектов, при помощи которых разработчик может решать прикладные задачи (рис. 3.9). Рассмотрим их. (В описаниях объектов вид регистра, например «РегистрСведений», заменено на «Регистр»).

Объект *РегистрыМенеджер* представляет собой коллекцию всех регистров конфигурации.

Объект *РегистрМенеджер* предназначен для управления отдельным регистром. При помощи свойств и методов этого объекта разработчик может работать с формами, обращаться к записям.

Объект *РегистрСписок* обеспечивает управление списком записей регистра в элементах управления.

Объект *РегистрВыборка* поддерживает механизм динамического перебора элементов справочника для получения доступа к их реквизитам. Данные доступны только для чтения.

Объект *РегистрНаборЗаписей* представляет собой коллекцию записей регистра. Позволяет добавлять и удалять записи, изменять их данные.

Объект *РегистрЗапись* представляет собой отдельную запись регистра, является элементом коллекции объекта *РегистрНаборЗаписей*.

Объект *РегистрКлючЗаписи* содержит поля записи, представляющие собой уникальный идентификатор записи.

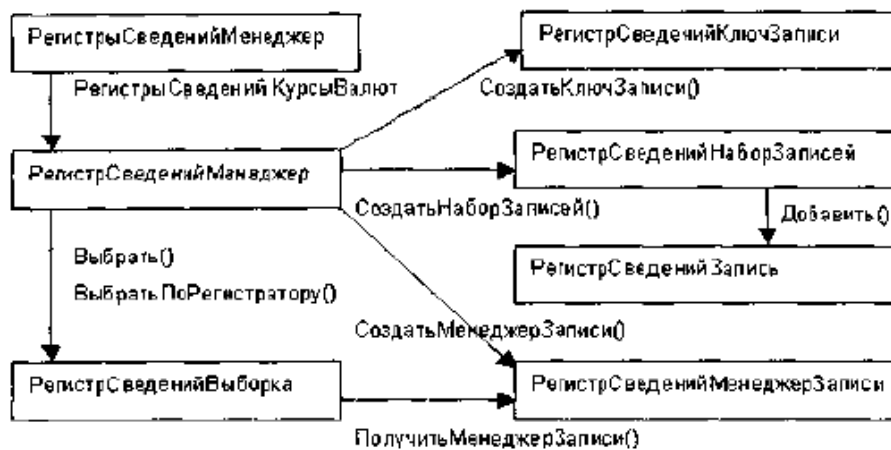


Рис. 3.9. Взаимосвязь объектов для работы с регистрами

Лекция 18: Регистры остатков. Основы программирования

Существует два вида регистров накопления: регистры накопления остатков и регистры накопления оборотов. Регистр накопления остатков позволяет хранить как итоговые значения ресурсов — остатки, так и изменения этих ресурсов — обороты. Регистр накопления оборотов является более «специализированным» видом регистра накопления и позволяет хранить только изменения ресурсов — обороты.

Существование регистра накопления оборотов связано с тем, что при автоматизации экономической деятельности существует большое количество ситуаций, когда требуется накапливать только обороты, а значения остатков не имеют смысла. Типичным примером использования регистра накопления оборотов является регистр Выручка и себестоимость продаж, хранящий информацию об объемах продаж:

Регистратор	Период ↓	Организация	Клиент	Заказ клиента	Номенклатура	Тип запасов или
	Номер строки	Подразделение	Контрагент	Соглашение	Характеристика	Вид запасов
		Склад /	Менеджер	Договор	Серия	
• Реализация товаров и услуг ТД00-000053 от 05.06.2015 12:00:00	05.06.2015 12:00:00	Торговый дом "Ко...	Саймон и Шустер	Реализация товар...	Комбайн MOULIN...	Собственный тое
	1	Отдел продаж тор...	Саймон и Шустер	Оптовое (кредит)		Собственный тое
		Склад бытовой тех...	Федоров Борис М...	Договор продажи ...		Поставщик: Эле
• Реализация товаров и услуг ТД00-000053 от 05.06.2015 12:00:00	05.06.2015 12:00:00	Торговый дом "Ко...	Саймон и Шустер	Реализация товар...	Кофеварка JACOB...	Собственный тое
	2	Отдел продаж тор...	Саймон и Шустер	Оптовое (кредит)		Собственный тое
		Склад бытовой тех...	Федоров Борис М...	Договор продажи ...		Менеджер: ...
• Реализация товаров и услуг ТД00-000053 от 05.06.2015 12:00:00	05.06.2015 12:00:00	Торговый дом "Ко...	Саймон и Шустер	Реализация товар...	Кофеварка JACOB...	Собственный тое
	3	Отдел продаж тор...	Саймон и Шустер	Оптовое (кредит)		Собственный тое
		Склад бытовой тех...	Федоров Борис М...	Договор продажи ...		Поставщик: Эле
• Реализация товаров и услуг ТД00-000053 от 05.06.2015 12:00:00	05.06.2015 12:00:00	Торговый дом "Ко...	Саймон и Шустер	Реализация товар...	Мясорубка MOULI...	Собственный тое
	4	Отдел продаж тор...	Саймон и Шустер	Оптовое (кредит)		Собственный тое
		Склад бытовой тех...	Федоров Борис М...	Договор продажи ...		Поставщик: Эле

Поскольку регистр накопления оборотов не накапливает остатки ресурсов, для него не имеет смысла «направление» движения ресурсов (приход или расход); накапливается только величина изменения ресурсов. Поэтому все записи регистра накопления оборотов отмечены одинаковыми пиктограммами.

Агрегаты

Для оборотных регистров накопления платформа поддерживает специальный механизм агрегатов, который позволяет значительно ускорить получение данных из регистров, содержащих большое количество записей — сотни тысяч и миллионы записей.

Форма списка и форма набора записей

Для того чтобы пользователь мог просматривать данные, содержащиеся в регистре накопления, система поддерживает форму представления регистра накопления — форму списка. Она позволяет выполнять сортировку и отбор отображаемой информации по нескольким критериям.

Система может автоматически генерировать эту форму. Наряду с этим разработчик имеет возможность создать собственные формы, которые система будет использовать вместо формы умолчания, в том числе и форму набора записей, которая позволяет добавлять, изменять и удалять записи регистра сведений.

Функциональные возможности регистра накопления

Основными функциональными возможностями, которые предоставляет регистр накопления разработчику, являются:

- выбор записей в заданном интервале по заданным критериям;
- выбор записей по регистратору;
- получение остатков и оборотов на указанный момент времени по заданным значениям измерений;
- режим работы с разделением итогов, который обеспечивает более высокую параллельность записи в регистр;
- отключение использования текущих итогов;
- расчет итогов на указанную дату;
- чтение, изменение и запись набора записей в регистр;
- возможность записи в регистр без пересчета итогов;
- полный пересчет итогов и пересчет итогов за указанный период.

Лекция 19: Запросы

Создание запроса и использование его результатов

Аналогом SQL в «1С:Предприятии» является язык запросов. Синтаксис и основные конструкции языка почти идентичны стандарту ANSI SQL 92. Возможна запись в русскоязычной и англоязычной нотациях.

Назначение запросов – извлечение данных, структура которых отличается от структуры таблиц и анализ данных. Для того, чтобы использовать запрос, необходимо выполнить следующие действия:

1. Создать объект запроса.
2. Разработать запрос. (Это можно делать вручную в редакторе кода или при помощи специального инструмента – Конструктора запросов)
3. Задать параметры запроса
4. Исполнить запрос
5. Обработать результаты запроса
6. Отобразить результаты запроса



Рис.4.1. Алгоритм выполнения запроса

Для использования запросов имеется объект встроенного языка:

НекоторыйЗапрос = Новый Запрос(текст);

Текст запроса также можно задавать не в конструкторе, но и при помощи свойства

Запрос.Текст = "ВЫБРАТЬ";

Параметры запроса задаются методом

Запрос.УстановитьПараметр(Имя, Значение);

Выполнение запроса осуществляется методом

Запрос.Выполнить();

Результаты запроса необходимо представить пользователю или использовать в программном коде для дальнейшей работы. Метод Выполнить() возвращает коллекцию РезультатЗапроса, которая содержит набор записей запроса. Структура записей определяется свойством Колонки

Обращаться к коллекции можно так:

РезультатЗапроса[1][1]

РезультатЗапроса.Колонки[1]

Также возможно обращение по именам колонок. При этом согласование типа данных ложится на разработчика.

Перебор записей результата запроса осуществляется при помощи объекта Выборка:

Выборка = РезультатЗапроса.Выбрать();

И его метода

Выборка.Следующий();

Перебор может быть организован в цикле:

Пока Выборка.Следующий() Цикл

...

КонецЦикла;

4.4. Структура и описание запроса

Текст запроса состоит из следующих логических частей:

<Описание запроса>

[<Объединение запросов>]

[<Упорядочивание результатов>]

[АВТОУПОРЯДОЧИВАНИЕ]

[<Описание итогов>]

Описание запроса является единственной обязательной частью запроса. Описание запроса определяется следующей конструкцией

ВЫБРАТЬ [РАЗРЕШЕННЫЕ] [РАЗЛИЧНЫЕ] [ПЕРВЫЕ <Количество>]

<Список полей выборки>

[ИЗ <Список источников>]

[ГДЕ <Условие отбора>]

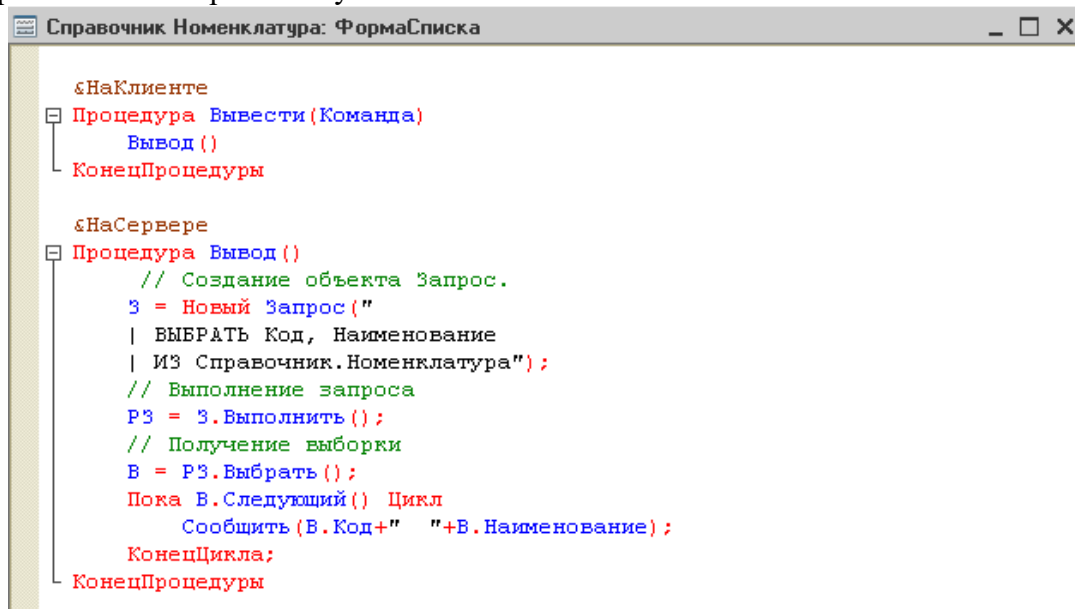
[СГРУППИРОВАТЬ ПО <Поля группировки>]

[ИМЕЮЩИЕ <Условие отбора>]

[ДЛЯ ИЗМЕНЕНИЯ [<Список таблиц верхнего уровня>]]

Команда ВЫБРАТЬ обеспечивает отбор данных в соответствии с заданным критерием. После нее приводится перечень имен полей, по которым требуется выполнить поиск. Поля разделяются запятой.

После оператора ИЗ задается имя таблицы, которой адресован запрос, имя виртуальной таблицы или вложенный запрос. В одном запросе можно обращаться к полям нескольких таблиц, в этом случае имена таблиц перечисляются через запятую.



```
«НаКлиенте
[ Процедура Вывести (Команда)
  Вывод ()
  КонечПроцедуры

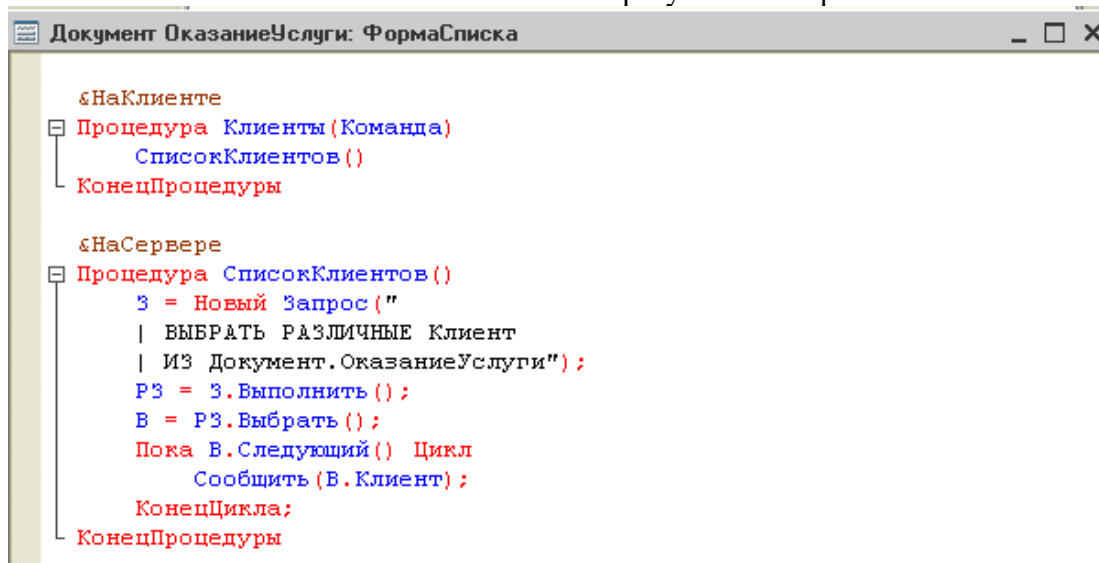
«НаСервере
[ Процедура Вывод ()
  // Создание объекта Запрос.
  З = Новый Запрос ("
  | ВЫБРАТЬ Код, Наименование
  | ИЗ Справочник.Номенклатура");
  // Выполнение запроса
  РЗ = З.Выполнить ();
  // Получение выборки
  В = РЗ.Выбрать ();
  Пока В.Следующий() Цикл
    Сообщить (В.Код+" "+В.Наименование);
  КонечЦикла;
  КонечПроцедуры
```

Вертикальная линия «|» в

языке запросов означает перенос выражения из предыдущей строки.

Ключевое слово РАЗРЕШЕННЫЕ означает, что запрос выберет только те записи, на которые у текущего пользователя есть права.

Ключевое слово РАЗЛИЧНЫЕ позволяет оставить в результате запроса только отличающиеся строки.



```
«НаКлиенте
[ Процедура Клиенты (Команда)
  СписокКлиентов ()
  КонечПроцедуры

«НаСервере
[ Процедура СписокКлиентов ()
  З = Новый Запрос ("
  | ВЫБРАТЬ РАЗЛИЧНЫЕ Клиент
  | ИЗ Документ.ОказаниеУслуги");
  РЗ = З.Выполнить ();
  В = РЗ.Выбрать ();
  Пока В.Следующий() Цикл
    Сообщить (В.Клиент);
  КонечЦикла;
  КонечПроцедуры
```

Конструкция ПЕРВЫЕ <Количество> позволяет задать предельное количество строк в результате запроса.

В запросе можно использовать дополнительные условия отбора. Для этого применяется оператор ГДЕ, после которого можно задавать ограничения на отбираемые поля.

В дополнительных условиях можно применять не только запрашиваемые поля, но и любые другие из данной таблицы. Допускается использование арифметических операторов сравнения, логических операторов AND, OR, NOT. В качестве параметров можно вводить числа, строки, даты, логические переменные.

```

«НаКлиенте
  Процедура Вывести(Команда) ...
«НаСервере
  Процедура Вывод()
    З = Новый Запрос("
    | ВЫБРАТЬ Наименование, ЭтоГруппа, ПометкаУдаления
    | ИЗ Справочник.Номенклатура
    | ГДЕ ЭтоГруппа=ЛОЖЬ И ПометкаУдаления=ЛОЖЬ");
    //Отбор только непомяченных на удаление элементов номенклатуры
    РЗ = З.Выполнить();
    В = РЗ.Выбрать();
    Пока В.Следующий() Цикл
      Сообщить(В.Наименование+" "+В.ЭтоГруппа+" "+В.ПометкаУдаления);
    КонечЦикла;
  КонечПроцедуры
  
```

Если источником данных является табличная часть объекта, то обращение к реквизитам самого объекта может происходить только по ссылке. Табличным частям и полям могут назначаться псевдонимы. Для этого используется ключевое слово КАК. Псевдонимы обязательно использовать при обращении к полям табличной части.

```

«НаКлиенте
  Процедура Номенклатура(Команда)
    СписокНоменклатуры()
  КонечПроцедуры
«НаСервере
  Процедура СписокНоменклатуры()
    З = Новый Запрос("
    | ВЫБРАТЬ Ссылка.Дата, Ссылка.Номер,
    |           Номенклатура КАК Ном, Количество КАК Кол
    | ИЗ Документ.ОказаниеУслуги.ПереченьНоменклатуры");
    //Обращение к реквизитам шапки документа и к табличной части
    РЗ = З.Выполнить();
    В = РЗ.Выбрать();
    Пока В.Следующий() Цикл
      Сообщить(В.Номер+" "+В.Дата+" "+В.Ном+" "+В.Кол);
    КонечЦикла;
  КонечПроцедуры
  
```

4.5. Взаимосвязь таблиц

Механизм соединения таблиц, участвующих в запросе, позволяет дополнительно ограничивать состав данных, получаемых в результате выполнения запроса. Если в операторе ИЗ перечислить несколько таблиц, а в операторе ВЫБРАТЬ задать некоторые поля из этих таблиц, то результатом будут все возможные сочетания значений полей таблиц. Для ограничения выборки используют операторы соединения таблиц.

Различают четыре вида соединения:

- [ВНУТРЕННЕЕ] СОЕДИНЕНИЕ
- ЛЕВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ
- ПРАВОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ
- ПОЛНОЕ [ВНЕШНЕЕ] СОЕДИНЕНИЕ

Рассмотрим различие вариантов соединения на следующем примере. Пусть есть две таблицы:

Таблица 1

Номенкл.	Номер1
Ручка	1
Карандаш	2

Вилка	3
-------	---

Таблица 2

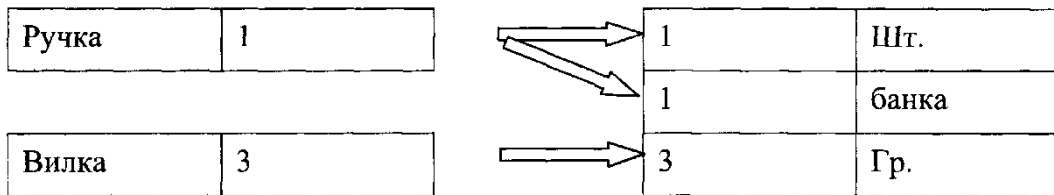
ЕдИзм	Номер2
Шт	1
Гр	3
Кг	4
Банка	1

Пусть условием связи будет:

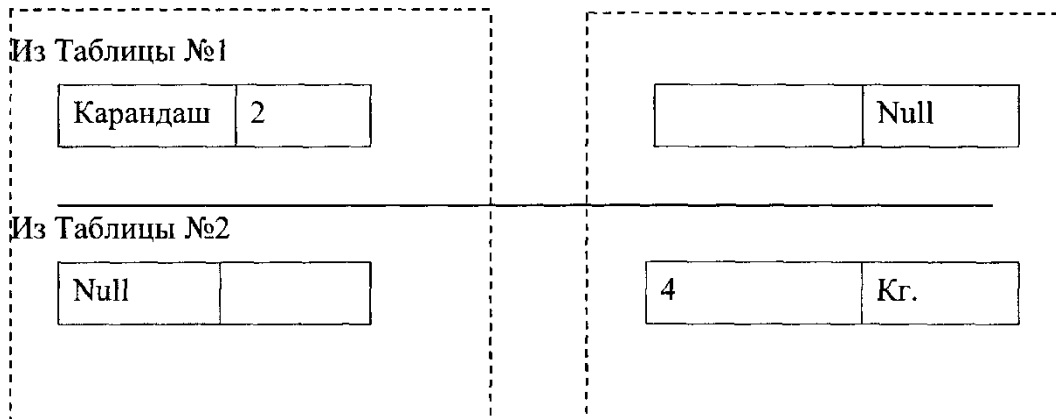
Таблица1.Номер1=Таблица2.Номер2

В качестве полей запроса определим две колонки: «Номенкл.» из первой таблицы и «ЕдИзм» из второй таблицы.

В соответствии с условием можно выделить записи, для которых условие выполняется:



Записи, неудовлетворяющие условию соединения:



Теперь рассмотрим варианты соединения:

Внутреннее соединение: в результат выполнения запроса войдут только данные записей из обеих таблиц, для которых выполняется условие соединения т.е.

Ручка	Шт.
Ручка	банка
Вилка	Гр.

Левое внешнее соединение: в результат выполнения запроса войдут данные из записей, для которых выполняется условие соединения и «не вошедшие» из Таблицы №1.

Ручка	Шт.
Ручка	банка
Вилка	Гр.
Карандаш	Null

Правое внешнее соединение обратно левому.

Ручка	Шт.
-------	-----

Ручка	банка
Вилка	Гр.
Null	Кг.

Полное внешнее соединение. В результате запроса войдут как записи, для которых выполнялось условие соединения, так и записи, полученные из «не вошедших» данных из обеих таблиц.

Ручка	Шт.
Ручка	банка
Вилка	Гр.
Карандаш	Null
Null	Кг.

В любом случае, даже в результате использования полного внешнего соединения не получается полная комбинация значений. (В случае полного соединения добавляются записи, не удовлетворяющие условию, а не все возможные их комбинации).

```

НаКлиенте
Процедура КупленныеМатериалы(Команда)
    КМ()
КонецПроцедуры

НаСервере
Процедура КМ()
    // Выбрать все приобретенные материалы и вывести их с указанием группы
    З = Новый Запрос ("
    | ВЫБРАТЬ Док.Материал КАК Мат,
    |         Спр.Родитель КАК Группа
    | ИЗ Документ.ПриходнаяНакладная.Материалы КАК Док
    |     СОЕДИНЕНИЕ Справочник.Номенклатура КАК Спр
    |         ПО Док.Материал=Спр.Ссылка");
    РЗ = З.Выполнить();
    В = РЗ.Выбрать();
    Пока В.Следующий() Цикл
        Сообщить (" "+В.Мат+" "+В.Группа);
    КонецЦикла;
КонецПроцедуры

```

```

// Показать курсы всех валют, которые хранятся в регистре сведений КурсыВалют.
// Возможно, что для некоторой валюты не будет найдено соответствующей записи в регистре
// сведений, но она также должна попасть в отчет.
| ВЫБРАТЬ Спр.Наименование, Рег.Курс
| ИЗ Справочник.Валюты КАК Спр
| ЛЕВОЕ СОЕДИНЕНИЕ
|     РегистрСведений.КурсыВалют.СрезПоследних() КАК Рег
| ПО Спр.Ссылка = Рег.Валюта

```

// Или то же самое можно сделать так:

```

| ВЫБРАТЬ Спр.Наименование, Рег.Курс
| ИЗ РегистрСведений.КурсыВалют.СрезПоследних() КАК Рег
| ПРАВОЕ СОЕДИНЕНИЕ
|     Справочник.Валюты КАК Спр

```

4.6. Упорядочивание

Часто требуется представить результат выполнения запроса упорядоченным по какому-то полю или группе полей. Для этого используется оператор

УПОРЯДОЧИТЬ ПО СписокПолей

Для каждого поля из списка можно указать порядок сортировки: ВОЗР, УБЫВ, ИЕРАРХИЯ. Если порядок не указан то сортировка происходит по возрастанию.

Последний порядок (ИЕРАРХИЯ) работает только для таблиц с иерархией, например таблиц иерархических справочников. Упорядочивание происходит сначала по первому полю списка, затем в рамках уже существующей сортировки упорядочивание по второму полю и т.д.

```
| ВЫБРАТЬ Наименование КАК Товар
| ИЗ Справочник.Номенклатура
| УПОРЯДОЧИТЬ ПО Товар ИЕРАРХИЯ");
```

Ключевое слово ПЕРВЫЕ ограничивает выборку несколькими первыми записями. Часто это слово используется в комбинации с упорядочиванием.

```
// Найти 3 самых дорогих товара
| ВЫБРАТЬ ПЕРВЫЕ 3 Наименование КАК Товар, ЦенаПродажи КАК Цена
| ИЗ Справочник.Номенклатура
| УПОРЯДОЧИТЬ ПО Цена УБЫВ");
```

4.7. Группировка и итоги

Часто при анализе данных необходимо получать суммирующую информацию для нескольких полей и всего отчета в целом или просто объединять записи в группы по какому-то признаку. Для этого используется механизм группировки и применяется оператор

СГРУППИРОВАТЬ ПО СписокПолей

Порядок следования полей в списке определяет и порядок вложенности (иерархию) групп. Для получения обобщенных данных по группам применяются агрегатные функции СУММА, МИНИМУМ, МАКСИМУМ, СРЕДНЕЕ, КОЛИЧЕСТВО. Эти функции можно использовать и в операторе ВЫБРАТЬ, но при этом в список полей группировки нужно добавлять все поля из ВЫБРАТЬ, не охваченные агрегатными функциями. Ключевое слово АВТОУПОРЯДОЧИВАНИЕ позволяет включить режим автоматического формирования полей для упорядочивания результата запроса.

```
Документ ОказаниеУслуги: ФормаСписка
«НаКлиенте
Процедура ОбъемПродаж(Команда)
    ОП()
КонецПроцедуры

«НаСервере
Процедура ОП()
    // Найти объем реализации каждой номенклатуры через агрегатную функцию СУММА
    З = Новый Запрос("
    | ВЫБРАТЬ Номенклатура,
    |           СУММА(Сумма) КАК ОбъемПродаж
    | ИЗ Документ.ОказаниеУслуги.ПереченьНоменклатуры
    | СГРУППИРОВАТЬ ПО Номенклатура
    | АВТОУПОРЯДОЧИВАНИЕ");
    РЗ = З.Выполнить();
    В = РЗ.Выбрать();
    Пока В.Следующий() Цикл
        Сообщить(" "+В.Номенклатура+" "+В.ОбъемПродаж);
    КонецЦикла;
КонецПроцедуры
```

В группировке можно указывать дополнительные условия отбора по полям группировки и агрегатным функциям. Для этого используется оператор ИМЕЮЩИЕ. Он применяется только с группировками. Он существенно замедляет работу сервера по выполнению запроса.

// Выбрать товары, которых продали не менее 100 штук

```
| ВЫБРАТЬ Номенклатура,  
|           СУММА(Количество) КАК Продано  
| ИЗ Документ.ПродажаТоваров.Товары  
| СГРУППИРОВАТЬ ПО Номенклатура  
| ИМЕЮЩИЕ Сумма(Количество) >= 100
```

Логическим завершением группировки является получение общих итогов по полям запроса. Для этого применяется оператор

ИТОГИ СписокПолейСАгрегатнымиФункциями ПО СписокПолей

В списке полей можно использовать оператор ОБЩИЕ.

```
// Итоги по группировкам  
// Объем продаж по каждой комбинации контрагента и номенклатуры
```

```
| ВЫБРАТЬ Контрагент, Номенклатура, Сумма  
| ИЗ РегистрНакопления.Продажи  
| ИТОГИ СУММА(Сумма)  
|       ПО Контрагент, Номенклатура  
| АВТОУПОРЯДОЧИВАНИЕ
```

4.8. Параметры

Для включения в текст запроса переменных значений непосредственно перед выполнением используется механизм параметров. Параметр – это аналог переменной любого языка программирования в языке запроса. Каждый параметр в тексте запроса обозначается знаком &, например &Параметр1. Параметры могут использоваться в операторе ГДЕ для сравнения с полями запроса. Заполнение значений параметров выполняется средствами встроенного языка перед выполнением запроса.

```
// Использование параметров и ключевого слова МЕЖДУ  
З = Новый Запрос ("  
| ВЫБРАТЬ  Наименование КАК Товар,  
|           ЦенаПокупки  
| ИЗ Справочник.Номенклатура  
| ГДЕ ЦенаПокупки МЕЖДУ &МинЦена И &МаксЦена");  
З.УстановитьПараметр("МинЦена", 100);  
З.УстановитьПараметр("МаксЦена", 1000);  
РЗ = З.Выполнить();
```

Лекция 20: Функции в запросе

Программирование 1С состоит не только из написания программы. 1С это слиток действий пользователя и данных с которыми он работает.

Данные хранятся в базе данных. Запросы 1С – это способ доставать данные из базы данных для того, чтобы показать пользователю в форме или чтобы обработать их.

Основополагающая часть отчета – это запрос 1С. В случае отчета СКД – это большая часть отчета.

```
Запрос = Новый Запрос();
```

```
Запрос.Текст = "ВЫБРАТЬ  
| Номенклатура.Ссылка  
| ИЗ  
| Справочник.Номенклатура КАК Номенклатура  
| ГДЕ  
| Номенклатура.Услуга";
```

Чтобы программировать в 1С недостаточно знать язык программирования 1С. Нужно еще знать язык запросов 1С.

Язык запросов 1С – это совершенно отдельный язык, который позволяет указать какие данные нам нужно достать из базы данных.

Он тоже двуязычен – то есть можно писать на русском или на английском. Он исключительно похож на язык запросов SQL и тем, кто знает такой – можно расслабиться.

Как используются Запросы 1С

Когда пользователь запускает 1С в режиме Предприятие – в запущенном клиенте нет ни грамма данных. Поэтому когда нужно открыть справочник – 1С запрашивает данные из базы данных, то есть делает запрос 1С.

Запросы 1С бывают:

- *Автоматические запросы 1С.* Формируются автоматически системой. Вы создали форму списка документов. Добавили колонку. Это значит, что при открытии этой формы в режиме Предприятие будет запрос и будут запрошены данные по этой колонке.

- *Полуавтоматические запросы 1С.* Есть множество методов (функций) в языке 1С, при обращении к которым происходит запрос к базе данных. Например .ПолучитьОбъект()

- *Ручные запросы 1С* (написанные программистом специально как запрос)

Вы можете написать запрос 1С самостоятельно в коде и выполнить его.

Создание и выполнение запросов 1С

Запрос 1С – это собственно текст запроса на языке запросов 1С. Текст можно написать ручками. То есть взять и написать (если знаешь этот язык).

Так как 1С продвигает концепцию визуального программирования, где многое или почти все можно сделать без написания кода ручками – есть специальный объект Конструктор запроса, который позволяет без знания языка запросов нарисовать текст запроса. Однако чудес не бывает – для этого нужно знать как работать с конструктором.

После того как готов текст запроса 1С – его нужно выполнить. Для этого есть объект в коде 1С Запрос(). Вот пример:

```
Запрос = Новый Запрос();
Запрос.Текст = "ВЫБРАТЬ
| Номенклатура.Ссылка
| ИЗ
| Справочник.Номенклатура КАК Номенклатура
| ГДЕ
| Номенклатура.Услуга";
Выборка = Запрос.Выполнить().Выбрать();
Пока Выборка.Следующий() Цикл
    Сообщить(Выборка.Ссылка);
КонецЦикла;
```

Как Вы видите в примере – после выполнения запроса 1С к нам приходит результат и мы должны его обработать. Результат – это одна или несколько строчек таблицы (в специальном виде).

Результат можно выгрузить в обычную таблицу:

```
Выборка = Запрос.Выполнить().Выгрузить(); //Результат – таблица значений
Или просто обойти по строчкам.
Выборка = Запрос.Выполнить().Выбрать();
Пока Выборка.Следующий() Цикл
    //Что-то делаем с результатами запроса
КонецЦикла;
```

Работа с запросами 1С

Основные принципы запросов 1С

Основные принципы построения запроса 1С –

ВЫБРАТЬ СписокПолей ИЗ НазваниеТаблицы ГДЕ Условия

Пример построения такого запроса 1С:

ВЫБРАТЬ

//список полей, которые надо выбрать

Ссылка,

Наименование,

Код

ИЗ

//наименование таблицы, откуда выбираем данные

//список таблиц – это список объектов в окне configurатора

Справочник.Номенклатура

ГДЕ

//указываем отбор

ВидТовара = &Услуга //отбор по внешнему значению

Или Услуга // «Услуга» реквизит типа Булево, отбор по значению Истина

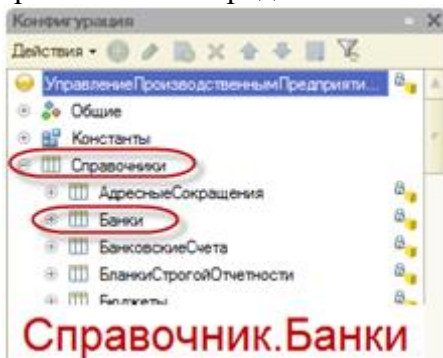
УПОРЯДОЧИТЬ ПО

//Сортировка

Наименование

Список таблиц 1С

Названия таблиц Вы смотрите в окне configurатора. Только необходимо вместо «Справочники» писать «Справочник», например «Справочник.Номенклатура» или «Документ.РеализацияТоваровУслуг» или «РегистрНакопления.Продажи».



Для регистров есть дополнительные таблицы (виртуальные), которые позволяют получить итоговые цифры.

РегистрСведений.ИмяРегистра.СрезПоследних(&Дата) – запрос 1С из регистра сведений, если он является периодическим, на определенную дату

РегистрНакопления.ИмяРегистра.Остатки(&Дата) – запрос 1С из регистра остатков на определенную дату

РегистрНакопления.ИмяРегистра.Обороты(&ДатаНачала, &ДатаОкончания) – запрос 1С из регистра оборотов за период с даты начала по дату окончания.

Дополнительные принципы

Когда мы запрашиваем список каких то данных – работают основные принципы. Но мы можем запросить также цифры и запрос может нам их посчитать (сложить например).

Например мы хотим посчитать с помощью запроса 1С количество документов в базе 1С:

ВЫБРАТЬ

//Количество(ИмяПоля) – считает количество

//Поле КАК ДругоеИмя – переименовывает поле

Количество(Ссылка) КАК КоличествоПроведенныхДокументов

ИЗ

Документ.РеализацияТоваровУслуг

ГДЕ

Проведен

Этот запрос 1С вернет нам общее количество документов. Однако в каждом документе есть поле Организация. Допустим мы хотим посчитать с помощью запроса 1С количество документов по каждой организации.

ВЫБРАТЬ

//просто поле документа

Организация,

//считаем количество

Количество(Ссылка) КАК КоличествоПоОрганизациям

ИЗ

Документ.РеализацияТоваровУслуг

ГДЕ

Проведен

СГРУППИРОВАТЬ ПО

//необходимо использовать, если в списке полей есть функция подсчета() и одновременно одно или несколько полей – тогда нужно группировать по этим полям

Организация

Этот запрос 1С вернет нам количество документов по каждой организации (также говорят «в разрезе организаций»).

Организация	Колво
МебельСтройКомплект завод	5
ЮФО МебельСтройКомплект зав...	5
Торговый дом "Комплексный"	27

Посчитаем дополнительно с помощью запроса 1С сумму этих документов:

ВЫБРАТЬ

//просто поле документа

Организация,

//считаем количество

Количество(Ссылка) КАК КоличествоПоОрганизациям,

//считаем сумму

Сумма(СуммаДокумента) КАК Сумма

ИЗ

Документ.РеализацияТоваровУслуг

ГДЕ

Проведен

СГРУППИРОВАТЬ ПО

//необходимо использовать, если в списке полей есть функция подсчета() и одновременно одно или несколько полей – тогда нужно группировать по этим полям

Организация

Этот запрос 1С вернет нам также сумму документов.

Организация	Колво	Сумма
МебельСтройКомпле...	5	36 6
ЮФО МебельСтройКо...	5	2 401 7
Торговый дом "Комп...	27	569 2

Если мы хотим посчитать с помощью запроса 1С итоговые значения (строка с указанием сколько всего) – нужно дополнительно указать требования получить итоги.

ВЫБРАТЬ

//просто поле документа

Организация,

//считаем количество

Количество(Ссылка) КАК КоличествоПоОрганизациям,

//считаем сумму

Сумма(СуммаДокумента) КАК Сумма

ИЗ

Документ.РеализацияТоваровУслуг

ГДЕ

Проведен

СГРУППИРОВАТЬ ПО

//необходимо использовать, если в списке полей есть функция подсчета() и одновременно одно или несколько полей – тогда нужно группировать по этим полям

Организация

ИТОГИ ПО Общ

Организация	Колво	Сумма
	3	3 007 8
МебельСтройКомпле...	5	36 6
ЮФО МебельСтройКо...	5	2 401 7
Торговый дом "Комп...	27	569 2

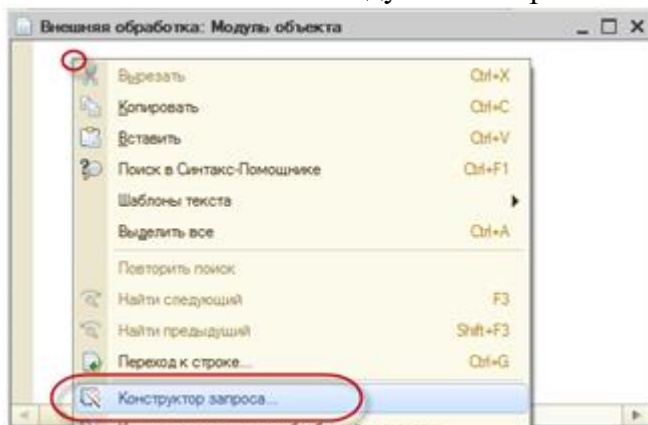
Язык запросов 1С обширен и сложен и мы не будем рассматривать в одном уроке все его возможности – читайте наши следующие уроки.

Кратко о дополнительных возможностях языка запросов 1С:

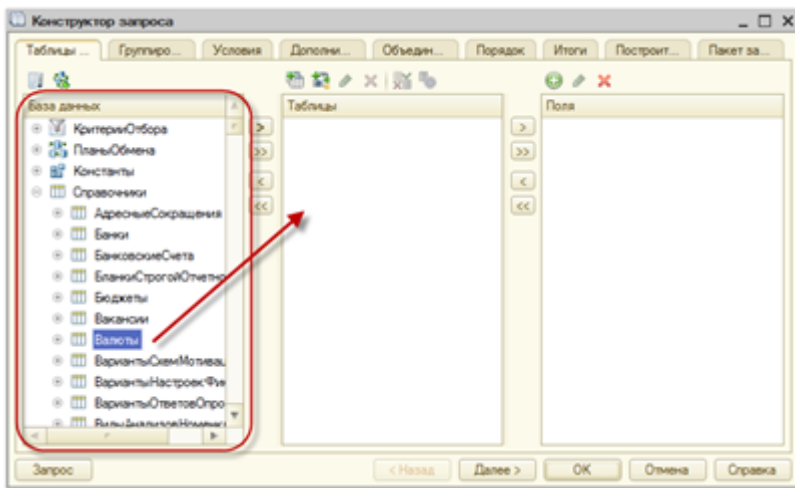
- Соединение данных из нескольких таблиц
- Вложенные запросы
- Пакетный запрос
- Создание собственных виртуальных таблиц
- Запрос из таблицы значений
- Использование встроенных функций получения значения и манипулирования значениями.

Конструктор запросов 1С

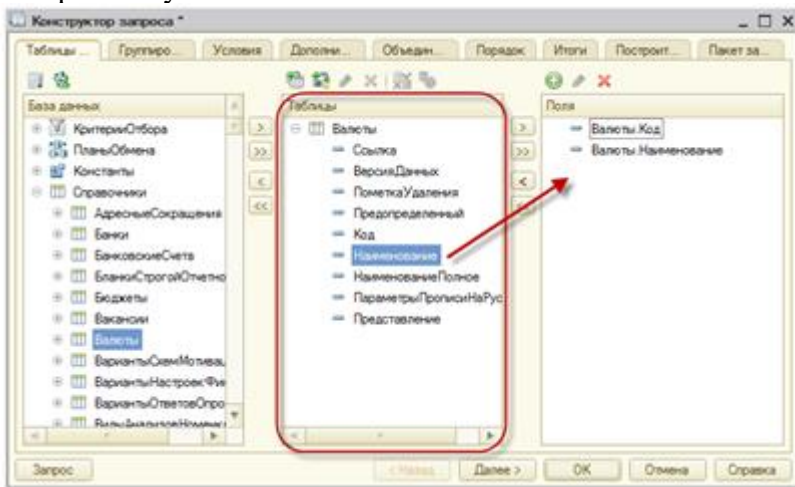
Чтобы не писать текст запроса руками – существует конструктор запросов 1С. Просто нажмите правой кнопкой мыши в любом месте модуля и выберите Конструктор запроса 1С.



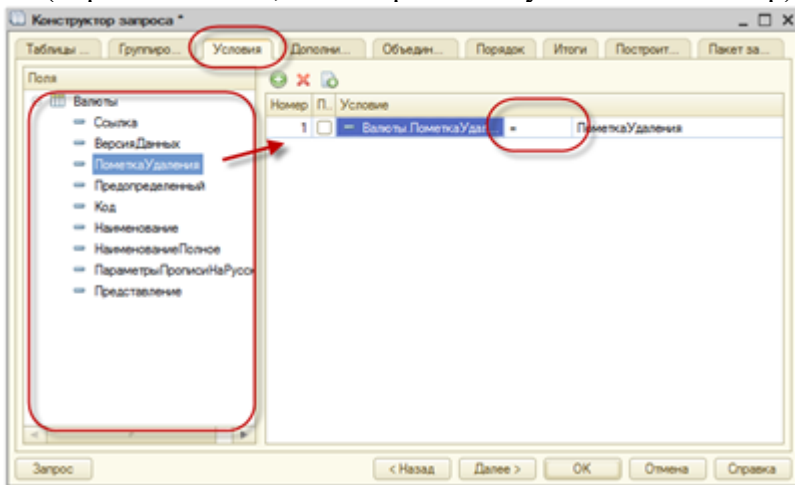
Выберите в конструкторе запросов 1С нужную таблицу слева и перетащите правее.



Выберите в конструкторе запросов 1С из таблицы нужные поля и перетащите правее. Если Вы хотели бы не просто выбрать поле, а применить к нему какую либо функцию суммирования – после перетаскивания щелкните по полю два раза мышкой. На закладке Группировка после этого нужно будет выбрать (перетащить) для группировки нужные поля.



На закладке Условия в конструкторе запросов 1С Вы можете выбрать нужные отборы таким же способом (перетаскивая поля, по которым Вы будете делать отбор). Не забудьте выбрать верное условие.



На закладке Порядок – указывается сортировка. На закладке Итоги – суммирование итогов.

С помощью конструктора запросов 1С Вы можете изучить любой существующий запрос. Для этого нажмите правой кнопкой на текст существующего запроса и выберите также конструктор запросов 1С – и запрос будет открыт в конструкторе запросов 1С.

Контрольные вопросы:

1. Какие бывают запросы в 1С?

2. Где можно посмотреть Названия таблиц?
3. Для чего нужен Конструктор запросов?

Лекция 21: Таблица значений

Посмотреть видео <https://www.youtube.com/watch?v=IJPTgZqCtEc>

Таблица значений – это специфический универсальный объект, предназначенный для хранения данных в табличном представлении. Ключевое отличие таблицы от прикладных объектов – это отсутствие привязки к физическим таблицам базы данных. Таблица значений существует только в оперативной памяти, что с одной стороны дает уникальные возможности, а с другой – накладывает определенные ограничения. Тем не менее возможности взаимодействия с таблицей сопоставимы со взаимодействием с реально существующими в базе данных объектами.

Исторически таблица значений в 1С имеет двойное назначение, являясь виртуальным аналогом существующих таблиц, но при этом – и элементом управления. С переходом на управляемое приложение большая часть этой функциональности устарела, но и в настоящее время она может быть также элементом пользовательского интерфейса, но с рядом существенных ограничений.

Структура таблицы значений как объекта

Свойства таблицы значений определяются комбинациями двух предопределенных коллекций: ее колонок и строк.

Таблица значений Колонки

Колонка таблицы значений – ее определяющее свойство. Именно набор колонок таблицы определяет ее структуру. Колонки соответствуют полям физических таблиц или знакомым по пользовательскому интерфейсу колонкам табличной части или журнала документов. Колонка может иметь внутреннее имя, тип значения и заголовок, отображаемый при интерактивной работе с таблицей.

Так как колонки являются коллекцией объектов, то вы можете добавлять, удалять и редактировать колонки.

Строка таблицы значений

С точки зрения программного интерфейса, строки – это отдельная коллекция, встроенная в таблицу значений. Они являются аналогом записей физических таблиц, то есть привычных для пользователя строк табличной части или журнала документов. Каждая отдельно взятая строка является объектом с набором именованных свойств, имена которых соответствуют именам колонок таблицы.

Таким образом, взаимодействие со строкой очень похоже на взаимодействие с прочими объектами. Можно считывать и записывать ее свойства, в том числе с использованием предопределенной функции «ЗаполнитьЗначенияСвойств()». Так как строки являются основной коллекцией таблицы значений, то чтобы удалить все строки таблицы, применяется метод «Очистить()».

Создать таблицу значений

Существует множество способов получить таблицу значений готовую к использованию. Рассмотрим некоторые из них. Каждый пример будет приведен в качестве листингов кода с комментариями.

Создание таблицы конструктором

Основной способ, позволяющий создать именно такую таблицу, которая нужна разработчику, к сожалению, наиболее трудоемкий, так как требует указать вручную все необходимые свойства таблицы.

```
ДемоТаблица = Новый ТаблицаЗначений;
```

```
// Первым делом инициализируем ТЗ
```

```
// Далее определяем необходимые параметры для новых колонок и добавляем их в коллекцию
```

```
// Создание колонки "Номенклатура"
```

```
Имя = "Номенклатура";
```

```
ТипЗначения = Новый ОписаниеТипов("СправочникСсылка.Номенклатура");
```

```
Заголовок = "Номенклатура (товар)";
```

```
ДемоТаблица.Колонки.Добавить(Имя, ТипЗначения, Заголовок);
```

```
// Создание колонки "Количество"
```

```
Имя = "Количество";
```

```
ТипЗначения = Новый ОписаниеТипов("Число");
```

```
ДемоТаблица.Колонки.Добавить(Имя, ТипЗначения);
```

```
// В результате данных манипуляций мы создали пустую таблицу с типизированными колонками
```

```
// Если вам необходимо использовать более точную типизацию примитивных типов, то следует использовать расширенный синтаксис конструктора "ОписаниеТипов"
```

Создание таблицы копированием

Если у вас под рукой есть эталон с подходящей структурой и/или составом, вы можете скопировать или выгрузить эталонную таблицу значений. Если эталоном является другая таблица, то необходимо применить метод «Скопировать таблицы-эталона». Если вы имеете дело с табличной частью или набором записей регистра, необходимо использовать метод «Выгрузить таблицу значений». Если вам необходима только структура, то можно воспользоваться методом «СкопироватьКолонки».

```
// Вариант с копированием из ТЗ-эталона всех строк но с сохранением лишь двух указанных колонок
```

```
КолонкиЭталона = "Номенклатура,Количество";
```

```
ДемоТаблица = ТаблицаЭталон.Скопировать(, КолонкиЭталона);
```

```
// Вариант с копированием из ТЗ-эталона предварительно отобранных строк,с сохранением двух указанных колонок
```

```
СтрокиЭталона = ОтобратитьВМассивНужныеНамСтрокиИзТаблицыЭталона();
```

```
КолонкиЭталона = "Номенклатура,Количество";
```

```
ДемоТаблица = ТаблицаЭталон.Скопировать(СтрокиЭталона, КолонкиЭталона);
```

```
// Вариант с копированием из ТЗ-эталона строк по указанному фильтру,с сохранением одной колонки "Номенклатура"
```

```
// Будут отобраны все строки где значение в колонке Количество равно 0, в результирующую таблицу попадет только колонка Номенклатура
```

```
ОтборСтрок = Новый Структура("Количество", 0);
```

```
КолонкиЭталона = "Номенклатура";
```

```
ДемоТаблица = ТаблицаЭталон.Скопировать(СтрокиЭталона, КолонкиЭталона);
```

```
// Вариант с полным копированием таблицы и последующем удалении одной строки со значением поля количество равным нулю и удалением целой колонки "Количество"
```

```
ОтборСтрок = Новый Структура("Количество", 0);
```

```
КолонкиЭталона = "Номенклатура";
```

```
ДемоТаблица = ТаблицаЭталон.Скопировать(СтрокиЭталона, КолонкиЭталона);
```

```
СтрокаТаблицы = ДемоТаблица.Найти(0, "Количество");
```

```
ДемоТаблица.Удалить(СтрокаТаблицы);
```

```
ДемоТаблица.Колонки.Удалить("Количество");
```

```
// Аналогичные варианты и их модификации можно применять к табличным частям и наборам записей регистров
```

Создание таблицы запросом

Если в базе данных существует эталон необходимой вам таблицы, то можно воспользоваться запросом для быстрого создания таблицы с нужной структурой.

```
// Пример с созданием пустой таблицы по образцу структуры регистра накопления
```

```
// Нетрудно догадаться, что таким образом можно получить и заполненную таблицу
```

```
Запрос = Новый Запрос("ВЫБРАТЬ ПЕРВЫЕ 0 * Из РегистрНакопления.ТоварынаСкладе");
```

```
РезультатЗапроса = Запрос.Выполнить();
```

```
ДемоТаблица = РезультатЗапроса.Выгрузить();
```

```
// Пример с созданием пустой таблицы по явно заданным типам и именам полей
```

```
Запрос = Новый Запрос;
```

```
Запрос.Текст =
```

```
"ВЫБРАТЬ ПЕРВЫЕ 0
```

```
| Значение(Справочник.Номенклатура.ПустаяССылка) КАК Номенклатура,
```

```
| ВЫРАЗИТЬ(0 КАК ЧИСЛО(15, 3)) КАК Количество";
```

```
РезультатЗапроса = Запрос.Выполнить();
```

ДемоТаблица = РезультатЗапроса.Выгрузить();

// ВАЖНО! Не стоит забывать, что в типах значений колонок, полученных из запроса всегда присутствует тип Null

// Таким образом, ТЗ созданная запросом, всегда имеет составные типы колонок

Лекция 22: Сортировка таблиц значений

Сортировка таблицы значений 1С осуществляется при помощи метода *Сортировать*, этот метод является процедурой, выполнение которой изменяет порядок строк в таблице значений. Данный метод имеет следующий синтаксис.

Сортировать(*Колонки, ОбъектСравнения*)

Где:

Колонки – обязательный параметр, который имеет тип Строка, в этой строке должны быть перечислены колонки таблицы значений, по которым осуществляется сортировка. После названия колонки можно указывать направление сортировки: или «Убыв», тогда сортировка осуществляется по убыванию, или «Возр», в этом случае сортировка будет по возрастанию. Если не указать направление сортировки, то сортировка будет **по возрастанию**. Можно перечислять несколько колонок, тогда таблица значений будет отсортирована сначала по первой указанной колонке, потом строки с одинаковым значением в первой колонке будут отсортированы по второй колонке и т.д.

ОбъектСравнения — необязательный параметр, в него передается переменная с типом *СравнениеЗначений*. Этот параметр необходимо применять, когда в сортируемых колонках содержатся значения не примитивных типов. Если этот параметр не указан, то элементы не примитивных типов сравниваются по своему строковому представлению. Если же он указан, то они сравниваются по следующим правилам:

- Объекты сравниваются по идентификатору
- Моменты времени сравниваются по дате и идентификатору объекта
- Если есть элементы с разными типами, то они сравниваются по коду типа
- Элементы остальных типов сравниваются по строковому представлению

Рассмотрим работу этого метода без второго параметра.

ТЗ = Новый ТаблицаЗначений;

ТЗ.Колонки.Добавить(«ФИО»);

ТЗ.Колонки.Добавить(«ДатаРождения»);

ТЗ.Колонки.Добавить(«Оклад»);

//первая строка

НСтр = ТЗ.Добавить();

НСтр.ФИО = «Иванов И.И.»;

НСтр.ДатаРождения = Дата(1985,1,15);

НСтр.Оклад = 2000;

//вторая строка

НСтр = ТЗ.Добавить();

НСтр.ФИО = «Петров П.С.»;

НСтр.ДатаРождения = Дата(1984,5,25);

НСтр.Оклад = 1000;

//третья строка

НСтр = ТЗ.Добавить();

НСтр.ФИО = «Тарасов К.В.»;

НСтр.ДатаРождения = Дата(1987,6,11);

НСтр.Оклад = 2000;

//сортируем по окладу и дате рождения (убывание)

ТЗ.Сортировать («Оклад,ДатаРождения УБЫВ»);

В этом коде мы отсортировали таблицу по окладу, поскольку название колонки было без указания направления, то сортировка осуществлялась по возрастанию. И вторая колонка для сортировки это дата рождения, здесь мы указали направление сортировки – по убыванию.

Посмотрим в отладке, какая таблица значений была до сортировки.

```
//третья строка
НСтр = ТЗ.Добавить ();
НСтр.ФИО = "Тарасов К.В";
НСтр.ДатаРождения = Дата (1987, 6, 11);
НСтр.Оклад = 2000;
//сортируем по окладу и дате рождения (убывание)
ТЗ.Сортировать ("Оклад,ДатаРождения УБЫВ");
```

Индекс	Значение элем...	Тип элемента	ФИО	ДатаРождения	Оклад
0	Строка Таблиц...	Строка Таблиц...	"Иванов И.И"	15.01.1985 0:00...	2 000
1	Строка Таблиц...	Строка Таблиц...	"Петров П.С"	25.05.1984 0:00...	1 000
2	Строка Таблиц...	Строка Таблиц...	"Тарасов К.В"	11.06.1987 0:00...	2 000

И после сортировки.

```
//третья строка
НСтр = ТЗ.Добавить ();
НСтр.ФИО = "Тарасов К.В";
НСтр.ДатаРождения = Дата (1987, 6, 11);
НСтр.Оклад = 2000;
//сортируем по окладу и дате рождения (убывание)
ТЗ.Сортировать ("Оклад,ДатаРождения УБЫВ");
```

Индекс	Значение элем...	Тип элемента	ФИО	ДатаРождения	Оклад
0	Строка Таблиц...	Строка Таблиц...	"Петров П.С"	25.05.1984 0:00...	1 000
1	Строка Таблиц...	Строка Таблиц...	"Тарасов К.В"	11.06.1987 0:00...	2 000
2	Строка Таблиц...	Строка Таблиц...	"Иванов И.И"	15.01.1985 0:00...	2 000

ИТОГИ ТАБЛИЦЫ ЗНАЧЕНИЙ IC

При работе с таблицей значений можно получить общий итог по какой-то колонке. Это осуществляется при помощи метода *Итог*. Этот метод является функцией, которая возвращает сумму значений определенной колонки всех строк таблицы. У этого метода следующий синтаксис:

Итог(Колонка)

Где: *Колонка* – это колонка, по которой будут просуммированы значения всех строк. Обычно данный метод применяется к колонкам, в которых содержатся значения с типом число.

Рассмотрим пример:

ТЗ = Новый ТаблицаЗначений;

ТЗ.Колонки.Добавить («ФИО»);

ТЗ.Колонки.Добавить («ДатаРождения»);

ТЗ.Колонки.Добавить («Зарплата»);

//первая строка

НСтр = ТЗ.Добавить();

НСтр.ФИО = «Иванов И.И»;

НСтр.ДатаРождения = Дата(1985,1,15);

НСтр.Зарплата = 2000;

//вторая строка

НСтр = ТЗ.Добавить();

НСтр.ФИО = «Петров П.С»;

НСтр.ДатаРождения = Дата(1984,5,25);

НСтр.Зарплата = 5000;

```

//третья строка
НСтр = ТЗ.Добавить();
НСтр.ФИО = «Тарасов К.В»;
НСтр.ДатаРождения = Дата(1987,6,11);
НСтр.Зарплата = 3000;
//итог по зарплатам
Общая = ТЗ.Итог(«Зарплата»);
Посмотрим, какой итог получился по указанной колонке.

```

```

//третья строка
НСтр = ТЗ.Добавить();
НСтр.ФИО = "Тарасов К.В";
НСтр.ДатаРождения = Дата(1987,6,11);
НСтр.Зарплата = 3000;
//итог по зарплатам
Общая = ТЗ.Итог("Зарплата");

```

КонецПроцедуры

Табло - 1		
Выражение	Значение	Тип
Общая	10 000	Число

СВЕРНУТЬ ТАБЛИЦУ ЗНАЧЕНИЙ 1С

Рассмотрим еще один метод, который очень полезен при работе с таблицами значений.

Метод *Свернуть* сворачивает используемую таблицу значений по значениям одной или нескольких колонок. Данный метод является процедурой, которая изменяет текущую таблицу значений.

Рассмотрим синтаксис этого метода

Свернуть(КолонкиГруппы, КолонкиСуммы)

Где:

КолонкиГруппы – колонки, по которым будет свернута таблица значений. Это обязательный параметр типа строка, в котором нужные колонки должны быть перечислены через запятую.

КолонкиСуммы – колонки, значения по которым будут просуммированы (необязательный параметр).

Причем, обратите внимание, после применения метода, в измененной таблице значений останутся только колонки группы и колонки суммы, остальные колонки, те, что были до свертки, будут удалены и информация в них потеряется!

Данный метод производит свертку таблицы значений: значения в колонках, которые перечисленные в первом параметре (*КолонкиГруппы*) будут сгруппированы, а значения в колонках, которые перечислены во втором параметре (*КолонкиСуммы*) будут просуммированы. Одна и та же колонка не может быть указана и в первом параметр, и во втором.

Рассмотрим пример: в таблице значений будут следующие колонки: ФИО, Вид работ, сумма выданных средств и дополнительная информация (аванс, «окончало» и т.п.), свернем эту таблицу значений по колонкам ФИО и Вид работ, и просуммируем колонку «Сумма».

```

ТабВыплат = Новый ТаблицаЗначений;
ТабВыплат.Колонки.Добавить(«ФИО»);
ТабВыплат.Колонки.Добавить(«ВидРабот»);
ТабВыплат.Колонки.Добавить(«Сумма»);
ТабВыплат.Колонки.Добавить(«Причина»);
//первая строка
НовВыплата = ТабВыплат.Добавить();
НовВыплата.ФИО = «Иванов»;
НовВыплата.ВидРабот = «Отделка»;
НовВыплата.Сумма = 1000;
НовВыплата.Причина = «Аванс»;
//вторая строка
НовВыплата = ТабВыплат.Добавить();

```

```

НовВыплата.ФИО = «Иванов»;
НовВыплата.ВидРабот = «Монтаж»;
НовВыплата.Сумма = 1500;
НовВыплата.Причина = «Расчет»;
//третья строка
НовВыплата = ТабВыплат.Добавить();
НовВыплата.ФИО = «Иванов»;
НовВыплата.ВидРабот = «Монтаж»;
НовВыплата.Сумма = 500;
НовВыплата.Причина = «аванс»;
//четверта строка
НовВыплата = ТабВыплат.Добавить();
НовВыплата.ФИО = «Петров»;
НовВыплата.ВидРабот = «Монтаж»;
НовВыплата.Сумма = 500;
НовВыплата.Причина = «аванс»;
//пятая строка
НовВыплата = ТабВыплат.Добавить();
НовВыплата.ФИО = «Петров»;
НовВыплата.ВидРабот = «Монтаж»;
НовВыплата.Сумма = 1500;
НовВыплата.Причина = «Расчет»;
//свертка
ТабВыплат.Свернуть («ФИО,ВидРабот»,«Сумма»);
Посмотрим в отладке, что было до свертки.

```

```

//пятая строка
НовВыплата = ТабВыплат.Добавить();
НовВыплата.ФИО = "Петров";
НовВыплата.ВидРабот = "Монтаж";
НовВыплата.Сумма = 1500;
НовВыплата.Причина = "Расчет";
//свертка
ТабВыплат.Свернуть ("ФИО, ВидРабот", "Сумма");

```

Индекс	Значение эл...	Тип элемента	ФИО	ВидРабот	Сумма	Причина
0	Строка Табл...	Строка Табл...	"Иванов"	"Отделка"	1 000	"Аванс"
1	Строка Табл...	Строка Табл...	"Иванов"	"Монтаж"	1 500	"Расчет"
2	Строка Табл...	Строка Табл...	"Иванов"	"Монтаж"	500	"аванс"
3	Строка Табл...	Строка Табл...	"Петров"	"Монтаж"	500	"аванс"
4	Строка Табл...	Строка Табл...	"Петров"	"Монтаж"	1 500	"Расчет"

И после свертки

```

//свертка
ТабВыплат.Свернуть ("ФИО, ВидРабот", "Сумма");

```

Индекс	Значение элем...	Тип элемента	ФИО	ВидРабот	Сумма
0	Строка Таблиц...	Строка Таблиц...	"Иванов"	"Отделка"	1 000
1	Строка Таблиц...	Строка Таблиц...	"Иванов"	"Монтаж"	2 000
2	Строка Таблиц...	Строка Таблиц...	"Петров"	"Монтаж"	2 000

Как видите, произошла группировка по колонкам *ФИО* и *ВидРабот*, а значения в колонке *Сумма* просуммировались, причем колонка *Причина* была удалена и информация в ней пропала.

Лекция 23: Список значений. Текст.

<https://www.youtube.com/watch?v=lr3ltdGfhe0>

Список значений — это программная коллекция (объект встроенного языка), содержащая пронумерованную последовательность элементов. Каждому элементу присваивается последовательный целочисленный номер — **индекс**, начинающийся с нуля. По индексу можно получить или установить элемент списка значений.

Элемент списка значений

Каждый элемент списка является программным объектом `ЭлементСпискаЗначений` встроенного языка и содержит следующие свойства:

- **Значение** (`Произвольный` тип) — хранимое значение произвольного типа
- **Представление** (`Строка`) — представление элемента списка для пользователя
- **Пометка** (`Булево`) — значение пометки элемента в списке
- **Картинка** (`Картинка`) — картинка элемента списка для пользователя

Создание списка значений

Создание пустого списка значений выполняется с помощью оператора `Новый`:

```
сз = Новый СписокЗначений;
```

Для добавления элементов в список значений используется одноименный метод `СписокЗначений.Добавить()`:

```
сз.Добавить(10);  
сз.Добавить(НачалоДня(ТекущаяДата()), "Сегодня");
```

Каждый новый элемент помещается в конец списка. Для доступа к отдельным элементам списка значений применяется **операция разыменования**: указывается имя переменной, объявленной как список значений, а затем, в квадратных скобках, указывается индекс элемента. Свойства полученного элемента можно использовать как слева от оператора присваивания:

```
сз[0].Значение = 10;  
сз[1].Значение = НачалоДня(ТекущаяДата());  
сз[1].Представление = "Сегодня";
```

Так и справа: в выражениях и в качестве параметров процедур и функций:

```
Сообщить("сз[0] = " + сз[0].Значение);
```

Границы списка значений

Как было сказано выше, левая граница списка значений постоянна и всегда равно нулю (0). Верхняя граница напрямую связана с количеством элементов в списке значений. Получить ее можно с использованием метода `СписокЗначений.Количество()` с последующим вычитанием 1.

Давайте рассмотрим распространенные операции со списком значений.

Перебор элементов списка значений

Для перебора (обхода) всех элементов списка значений обычно используется оператор цикла `Для`:

```
Для Инд = 1 По сз.Количество() Цикл  
    Сообщить("сз[" + Инд + "] = " + сз[Инд-1].Представление);  
КонецЦикла;
```

Либо оператор цикла `Для Каждого`:

```
Для Каждого Элт Из сз Цикл
```

```
Сообщить("сз[" + сз.Индекс(Элт) + "]" = " + Элт.Представление);  
КонецЦикла;
```

Второй способ больше подходит для ситуации, когда нужно только последовательно перебрать все значения списка. Его использование повышает «читабельность кода».

Поиск в списке значений

Для поиска значений в списке предназначен метод СписокЗначений.НайтиПоЗначению(). Он возвращает найденный элемент списка значений, либо Неопределено, если указанного значения в списке нет.

```
НайдЭлт = сз.НайтиПоЗначению(10);
```

```
Если НЕ НайдЭлт = Неопределено Тогда
```

```
    Сообщить("сз[" + сз.Индекс(НайдЭлт) + "]" = " + НайдЭлт.Представление);
```

```
КонецЕсли;
```

Сортировка списка значений

Сортировка элементов в списке значений может производиться двумя способами:

- По значению — реализуется методом СписокЗначений.СортироватьПоЗначению()
- По представлению — реализуется методом СписокЗначений.СортироватьПоПредставлению()

```
сз = Новый СписокЗначений
```

```
сз.Добавить(1, "Первый");
```

```
сз.Добавить(3, "Третий");
```

```
сз.Добавить(2, "Второй");
```

```
сз.СортироватьПоЗначению();
```

Методы объекта СписокЗначений

<u>Вставить()</u>	Добавляет элемент в произвольное место списка
<u>ВыбратьЭлемент()</u>	Открывает диалог выбора элемента списка значений
<u>ВыгрузитьЗначения()</u>	Выгружает значения списка в массив
<u>Добавить()</u>	Добавляет элемент в конец списка
<u>ЗагрузитьЗначения()</u>	Загружает значения из массива
<u>ЗаполнитьПометки()</u>	Устанавливает значение пометок элементов списка
<u>Индекс()</u>	Возвращает индекс элемента списка значений
<u>Количество()</u>	Возвращает количество элементов в списке
<u>НайтиПоЗначению()</u>	Выполняет поиск элемента в списке по значению
<u>НайтиПоИдентификатору()</u>	Выполняет поиск элемента в списке по его идентификатору
<u>ОтметитьЭлементы()</u>	Открывает диалог отметки элементов списка значений
<u>Очистить()</u>	Очищает список значений
<u>Получить()</u>	Возвращает элемент списка по индексу
<u>Сдвинуть()</u>	Сдвигает элемент в списке вверх или вниз

<u>Скопировать()</u>	Возвращает копию текущего списка значений
<u>СортироватьПоЗначению()</u>	Сортирует элементы списка значений по значению
<u>СортироватьПоПредставлению()</u>	Сортирует элементы списка значений по представлению
<u>Удалить()</u>	Удаляет указанный элемент из списка

Лекция 24: Выгрузка списка значений

При формировании пользовательского интерфейса прикладной задачи довольно часто возникают ситуации, когда для удобства работы пользователя необходимо дать ему возможность выбирать какое-либо значение из заранее подготовленного списка. Возможности работы со справочниками, журналами и перечислениями предоставляют достаточно мощные средства выбора этих объектов. Однако, существует также необходимость в выборе значений из списков, сформированных и отсортированных нестандартным образом, которые невозможно или неудобно получать непосредственно при работе со стандартными визуальными представлениями журналов, справочников или перечислений. Для этого в системе 1С:Предприятие используется специальный агрегатный тип данных — «СписокЗначений» и специальные элементы форм диалога: «Список» и «Поле со списком».

Объект типа «СписокЗначений» — это средство языка (не сохраняемый в БД объект), которое позволяет строить «динамические массивы» и манипулировать ими (добавлять, редактировать, удалять элементы, сортировать). Список значений может быть наполнен значениями любого типа, т. е. в одном списке типы хранимых значений могут быть разными. Одним из примеров использования данного объекта может служить организация выбора конкретного документа из списка возможных документов, сформированного по сложному алгоритму.

Контекст работы со Списком Значений

Во всех программных модулях вызов соответствующих методов может выполняться при помощи переменной со ссылкой на объект типа «СписокЗначений». Такие объекты можно создавать при помощи функции СоздатьОбъект, ссылка на который присваивается переменной. Кроме того, если в форму диалога при помощи визуальных средств configurатора вставлены специальные элементы форм диалога «Список» и «Поле со списком», то идентификаторы этих элементов доступны в контексте программного модуля этой формы как уже существующие объекты типа «СписокЗначений».

Чтобы вызвать метод объекта, имя метода (с указанием необходимых параметров) пишется через точку после имени объекта.

При создании объекта типа «СписокЗначений» при помощи функции СоздатьОбъект, в качестве названия агрегатного типа данных обязательно Должно выступать ключевое слово СписокЗначений.

Англоязычный синоним ключевого слова СписокЗначений — ValueList.

Пример:

```
МойСписок = СоздатьОбъект("СписокЗначений");
```

Методы объекта Список Значений

ДобавитьЗначение

Добавить значение в список.

Синтаксис:

ДобавитьЗначение(<Значение>, <Строка>)

Англоязычный синоним:

AddValue

Параметры:

<Значение> Выражение со значением, которое добавляется в список.
<Строка> Необязательный параметр. Строковое выражение, содержащее задаваемое символьное представление добавляемого значения. По умолчанию принимает стандартное в системе 1С:Предприятие символьное представление объекта.

Описание:

Метод ДобавитьЗначение добавляет значение и его символьное представление в конец списка. Представление используется в дальнейшем для удобства сортировки и выбора значений из списка (отображается в диалоговом окне выбора).

Пример:

```
// Объявляем переменные
```

```
Перем Список;
```

```
Перем ВР;
```

```
// процедура выбора значения
```

```
Процедура ВыбратьВР()
```

```
Если Список.ВыбратьЗначение(ВР, "Выбор дог: паты") = 1 Тогда
```

```
    Расчет = ВР;
```

```
    НазваниеВР = Расчет.Код;
```

```
КонецЕсли;
```

```
КонецПроцедуры
```

```
// инициализация переменных
```

```
Список = СоздатьОбъект("СписокЗначений");
```

```
Список.ДобавитьЗначение(ВидРасчета.ДоплатаПроцентом, "Процентом");
```

```
Список.ДобавитьЗначение(ВидРасчета.ДоплатаСуммой, "Суммой");
```

```
// первоначальное значение
```

НазваниеВР = Расчет.Код;

ВставитьЗначение

Вставить значение в указанную позицию списка.

Синтаксис:

ВставитьЗначение(<Позиция>, <Знач>, <Строка>, <Колич>)

АНГЛОЯЗЫЧНЫЙ СИНОНИМ:

InsertValue

Параметры:

- <Позиция> Числовое выражение — номер позиции в списке, начиная с которого будут вставлены новые значения. Номер позиции может быть от 1 до РазмерСписка() + 1.
- <Знач> Выражение со значением, которое добавляется в список.
- <Строка> Необязательный параметр. Строковое выражение, содержащее задаваемое символьное представление добавляемого значения. По умолчанию принимает стандартное в системе 1С:Предприятие символьное представление объекта.
- <Колич> Необязательный параметр. Числовое выражение — количество повторов. По умолчанию — 1.

Описание:

Метод ВставитьЗначение добавляет значение и его символьное представление в указанную позицию списка <Позиция> заданное число раз <Колич>. Представление используется в дальнейшем для удобства сортировки и выбора значений из списка (отображается в диалоговом окне выбора).

Пример:

```
// Объявляем переменные
```

```
Перем Список;
```

```
Перем ВР;
```

```
// процедура выбора значения
```

```
Процедура ВыбратьВР()
```

```
Если Список.ВыбратьЗначение(ВР, "Выбор доплаты") = 1 Тогда
```

```
    Расчет = ВР;
```

```
    НазваниеВР = Расчет.Код;
```

```
КонецЕсли;
```

```
КонецПроцедуры
```

// инициализация переменных

Список = СоздатьОбъект("СписокЗначений");

Список.ВставитьЗначение(1, Вид Расчета.ДоплатаПроцентом, "Процентом");

Список.ВставитьЗначение(1, Вид Расчета.ДоплатаСуммой, "Суммой");

// первоначальное значение

НазваниеВР = Расчет.Код;

РазмерСписка

Определить размер списка.

Синтаксис:

РазмерСписка()

Лекция 25: Экспорт-импорт данных

Способы выгрузки информации

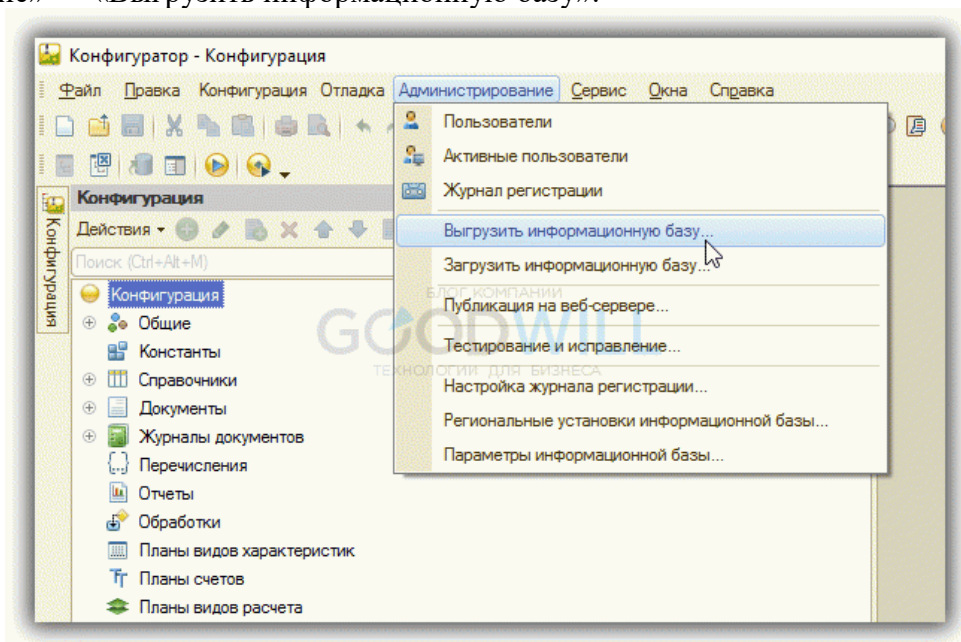
В 1С есть несколько методов, с помощью которых Вы можете получить информацию без привлечения специалистов:

- Выгрузка базы из 1С 8.3 целиком — для последующей загрузки на другом компьютере
- Сохранение отчетов и печатных форм в Excel, Word или PDF
- Выгрузка таблиц из 1С
- Получение информации с помощью консоли запросов

Подробнее о них:

Как выгрузить базу из 1с 8.3 целиком (в файл .dt)

Для этого понадобится зайти в режим configurатора, затем в главном меню выбрать пункт «Администрирование» — «Выгрузить информационную базу»:



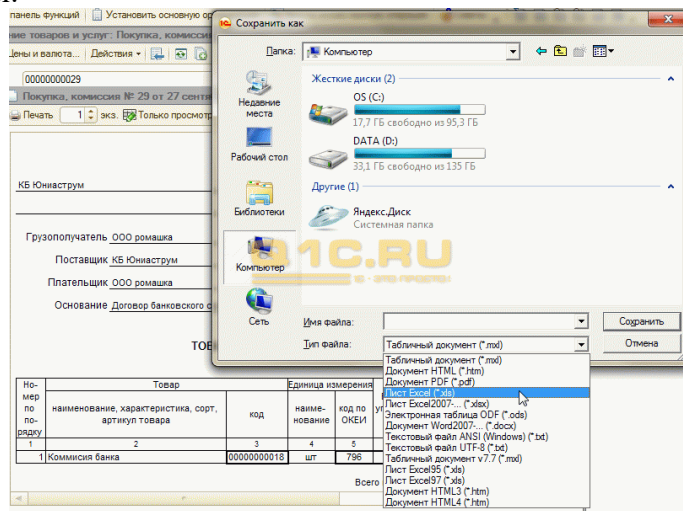
Затем потребуется только указать название будущего файла и его расположение на диске.

Для последующей загрузки в другую базу нажмите на кнопку «Загрузить информационную базу» и выберите предварительно выгруженный файл.

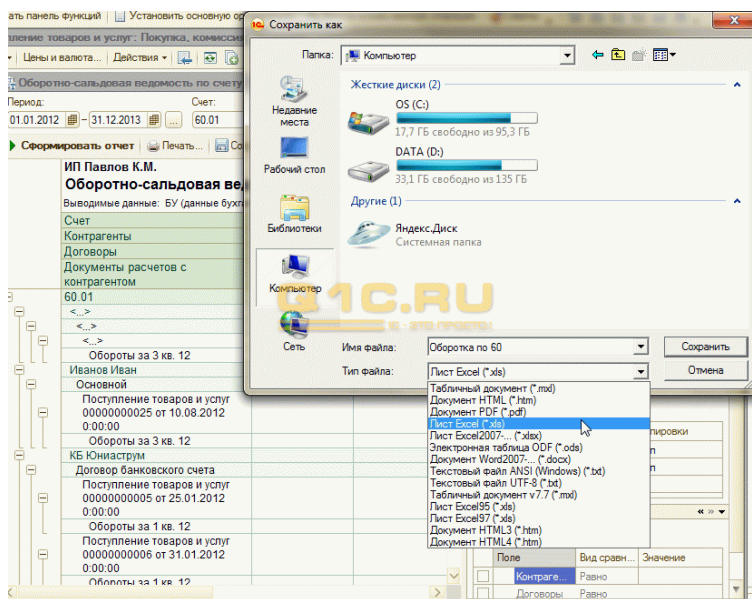
Выгрузка из 1С в Excel отчетов и печатных форм

Этот способ универсальный и очень простой. Для этого достаточно при открытии печатной формы или отчета.

Например, открываем печатную форму ТОРГ 12, нажимаем ctrl+S (или в меню Файл – Сохранить как) и выбираем тип файла и его имя:



Для отчетов действует та же система – отчет выгружается в два клика:

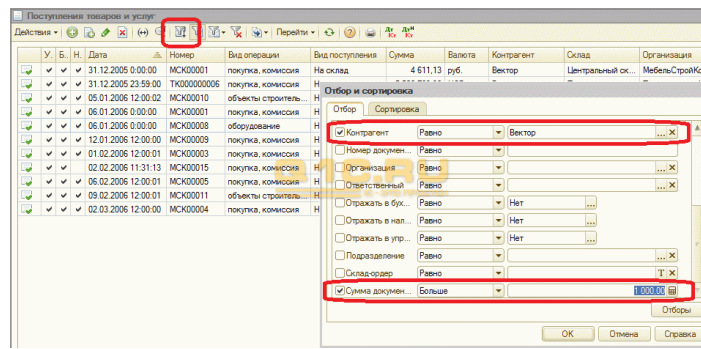


Доступно сохранение в основные форматы – excel, word, pdf, html, txt и тд.

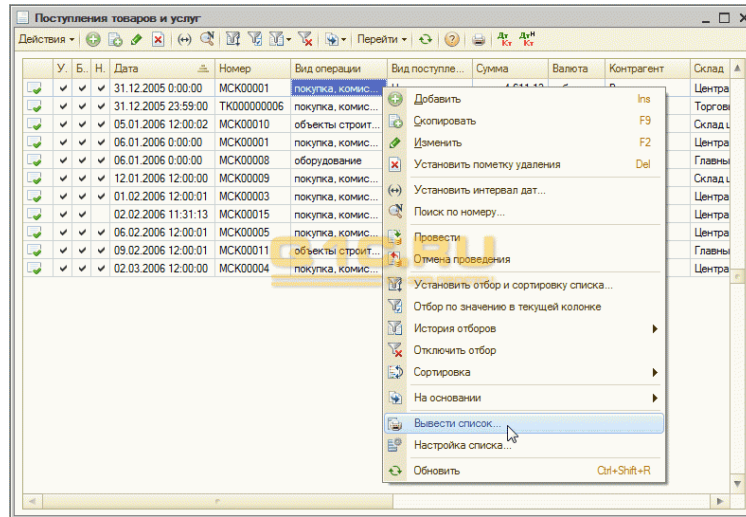
Выгрузка таблиц из 1С 8.3

Часто встречаются такие банальные задачи, как «получить список платежей за текущий год», «выгрузить книгу покупок или продаж» или «выгрузить товары из накладной №256». Решить такие задачи достаточно просто.

Например, мне необходимо получить все поступления от контрагента «Вектор» с суммой более 1000. Откроем форму списка и установим нужные отборы в списке:



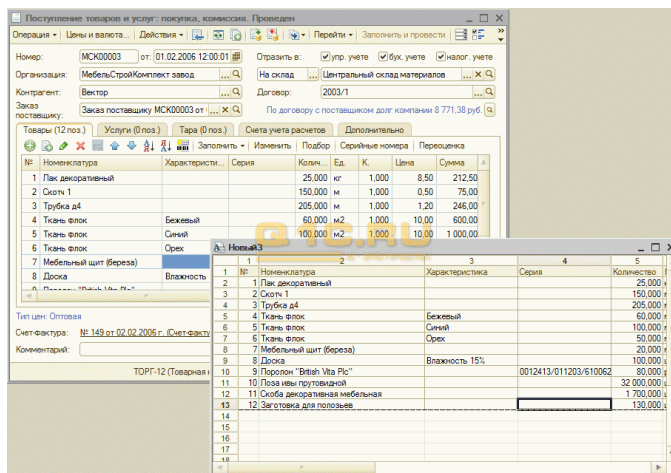
Далее, когда мы получили список документов, необходимо вызвать контекстное меню (правая кнопка на таблице или меню «Действия») и выбрать в списке команд «Вывести список»:



Система предложит вариант вывода – в табличный документ или текстовый, выберем табличный. Программа 1С выведет информацию в такой табличный документ, который, подобно отчетам, можно сохранить в нужный нам формат:

1	2	3	4	5	6	7	8	9	10
У	Б.	Н.	Дата	Номер	Вид операции	Вид поступления	Сумма	Валюта	Конт
✓	✓	✓	31.12.2005 0:00:00	МСК00001	покупка, комиссия	На склад	4 611,13	руб.	Вект
✓	✓	✓	31.12.2005 23:59:00	ТК000000006	покупка, комиссия	На склад	5 502 763,96	USD	Вект
✓	✓	✓	05.01.2006 12:00:02	МСК00010	объекты строительства	На склад	177 000,00	руб.	Вект
✓	✓	✓	06.01.2006 0:00:00	МСК00001	покупка, комиссия	На склад	12 460,83	руб.	Вект
✓	✓	✓	06.01.2006 0:00:00	МСК00008	оборудование	На склад	2 655 000,00	руб.	Вект
✓	✓	✓	12.01.2006 12:00:00	МСК00009	покупка, комиссия	На склад	2 655,00	руб.	Вект
✓	✓	✓	01.02.2006 12:00:01	МСК00003	покупка, комиссия	На склад	24 984,73	руб.	Вект
✓	✓	✓	02.02.2006 11:31:13	МСК00015	покупка, комиссия	На склад	9 027,50	руб.	Вект
✓	✓	✓	06.02.2006 12:00:01	МСК00005	покупка, комиссия	На склад	3 540,00	руб.	Вект
✓	✓	✓	09.02.2006 12:00:01	МСК00011	объекты строительства	На склад	2 773,00	руб.	Вект
✓	✓	✓	02.03.2006 12:00:00	МСК00004	покупка, комиссия	На склад	9 244,12	руб.	Вект

Подобные действия можно произвести практически для любой табличной части в 1С 8.2. Например, нам потребовалось выгрузить список товаров из поступления № МСК00003, не проблема. Делается это очень просто – открываем документ, вызываем контекстное меню и через пару кликов получаем нужную выгрузку в нужном формате:



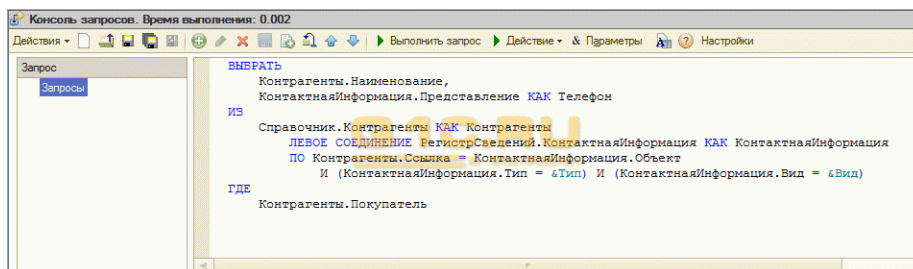
Выгрузка в Excel журнала проводок из 1С:

Получение информации с помощью консоли запросов

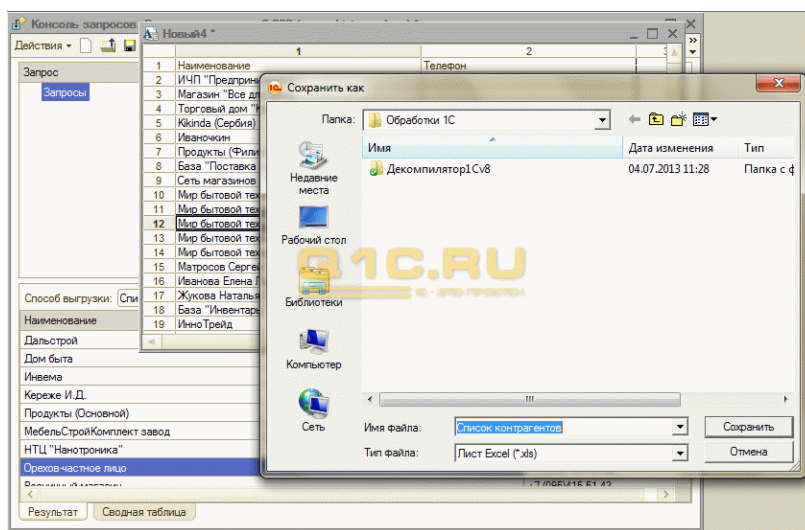
Однако не вся информация может быть найдена в простой табличной части 1С 8.2. Иногда требуется собрать данные из нескольких источников данных, соединить их по некоторым полям.

Этот способ подходит для технически подкованных людей, которые знают основы языка запросов. Заключается он в том, чтобы получить результат запроса с помощью консоли запросов и сохранить его в нужный нам формат.

Например, требуется получить всех контрагентов-поставщиков и их телефонный номер. Для этого напишем следующий запрос:



Полученную таблицу можно также выгрузить в нужный формат файла:



Задание 1. Просмотрите видео ролик на тему «Экспорт-импорт данных» <https://www.youtube.com/watch?v=bxi2O9uro1k>

Лекция 26: Работа со структурой файла DBF

Одним из самых распространенных форматов баз данных до сих пор остается формат DBF. И неумение работать с ним из 1С – серьезное ограничение профессиональных навыков для любого специалиста. Тем более изучение теории и практика по этому вопросу потребует совсем немного времени. К тому же работа с DBF файлом никаких дополнительных библиотек не требует – все инструменты встроены в платформу 1С 8.3. Чтение, создание DBF и запись в данный формат

Вся работа с DBF в 1С происходит с помощью специального объекта – xBase. Рассмотрим основные действия с файлами, начав с чтения из конкретного DBF. В первую очередь необходимо расположить файл базы данных в каталоге, куда есть доступ у пользователя. Код для считывания данных в 1С достаточно прост: Указываем путь к базе в формате DBF;

```
ПутьКФайлу = "D:\Nom.DBF";
```

Создаем объект для взаимодействия с файлом, открываем его и устанавливаем указатель на первую строку с данными;

```
ФайлDBF = Новый XBase;
```

```
ФайлDBF.ОткрытьФайл(ПутьКФайлу,,Истина);
```

В цикле обходим всю базу и производим действия с элементами. В данном случае просто отражаем их на экране пользователя;

```
Пока НЕ ФайлDBF.ВКонце() Цикл
```

```
Сообщить(" " + ФайлDBF.НАИМ + " - "+ФайлDBF.ARTI);
```

```
ФайлDBF.Следующая();
```

```
КонецЦикла;
```

```
Закрываем файл.
```

```
ФайлDBF.ЗакрытьФайл();
```

Вторая часто встречающаяся задача по взаимодействию с объектом – выгрузка DBF из 1С 8.3. Здесь намного легче создать конечный файл нужного формата программно, чем использовать дополнительно программное обеспечение. Этот алгоритм в 1С тоже достаточно прост и состоит из нескольких строк кода:

```
Создаем объект XBase;
```

```
НовыйФайл = Новый XBase;
```

Определяем кодировку нового файла. Существует кодировка для Windows – ANSI и OEM предназначенная для DOS;

```
НовыйФайл.Кодировка = КодировкаXBase.ANSI;
```

```
Необходимо описать все колонки будущего файла DBF, указав их тип;
```

```
НовыйФайл.Поля.Добавить("Naim", "S", 100);
```

```
НовыйФайл.Поля.Добавить("KOD", "N", 10);
```

Укажем каталог, в котором будет происходить создание DBF и запишем итоговый файл.

```
Путь = "D:\Clients.DBF";
```

```
НовыйФайл.СоздатьФайл(Путь);
```

```
НовыйФайл.Записать();
```

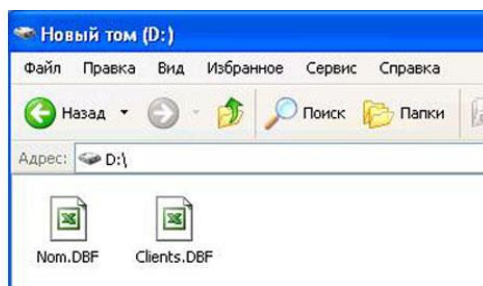


Рис.1

Расположение файлов DBF

После того как мы создали файл нужного формата, потребуется выгрузка в DBF данных. Программируя этот процесс, помните о том, что данные ИБ находятся на сервере. Чтобы записать имеющиеся данные из 1С необходимо к предыдущему коду добавить следующий алгоритм:

Получаем информацию из справочника;

Контрагенты = Справочники.Контрагенты.Выбрать();

В цикле последовательно добавляем по 1строчке и записываем;

Пока Контрагенты.Следующий() Цикл

НовыйФайл.Добавить();

НовыйФайл. Naim = Контрагенты.Наименование;

НовыйФайл. KOD = Контрагенты.Код;

НовыйФайл.Записать();

КонецЦикла;

Закрываем файл и проверяем, что запись прошла удачно.

НовыйФайл.ЗакретьФайл();

Это основы работы с данным форматом. Существуют и другие способы открыть из 1С DBF, но хBase остается самым простым по синтаксису и пониманию. Поняв, как осуществляется простейшая выгрузка и загрузка из DBF, можно приступать к более сложным элементам, например, использованию индексов для поиска, удалению или изменению конкретных записей в DBF, а также применению технологии ADO.

Составить презентацию по теме «Работа со структурой файла DBF»

Лекция 27: Файловая система

Файловый вариант работы — один из вариантов работы системы «1С:Предприятие 8». Файловый вариант работы рассчитан на персональную работу одного пользователя или работу небольшого количества пользователей в локальной сети.

В этом варианте все данные информационной базы (конфигурация, база данных, административная информация) располагаются в одном файле — файловой базе данных. Работу с этой базой данных осуществляет файловая СУБД, разработанная фирмой «1С» и являющаяся частью платформы.



Такой вариант работы обеспечивает легкость установки и эксплуатации системы. При этом для работы с информационной базой не требуются дополнительные программные средства, достаточно иметь операционную систему и «1С:Предприятие 8».

Файловый вариант работы обеспечивает целостность информационной базы и простое создание резервных копий. Исключена ситуация, когда пользователь может по ошибке (например, при копировании информационной базы) перепутать различные файлы информационной базы и привести, таким образом, систему в неработоспособное состояние.

Резервное копирование может осуществляться на файловом уровне, путем простого копирования файла информационной базы.

При работе в файловом варианте, за счет использования механизма транзакций, платформа «1С:Предприятия 8» минимизирует риск нарушения целостности данных при сбоях компьютеров и локальной сети. Однако, разумеется, использование клиент-серверного варианта обеспечивает большую надежность, так как в нем обеспечивается полная независимость записи данных на сервере от сбоев клиентских компьютеров и локальной сети.

Утилита тестирования и исправления файловой базы данных

В дистрибутив платформы входит утилита, предназначенная для тестирования и исправления файловой базы данных после сбоев. Подробнее...

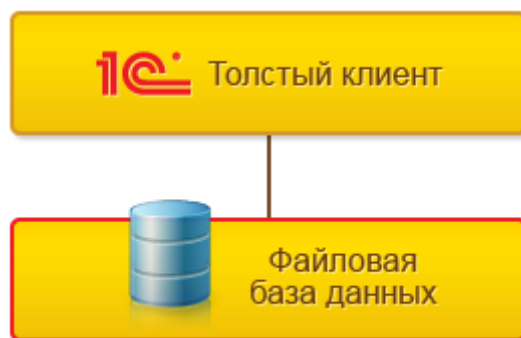
Техническая архитектура файлового варианта работы

Работа в файловом варианте возможна как напрямую, непосредственно с файлом базы данных, так и через веб-сервер, если используются клиентские подключения по протоколу HTTP или HTTPS.



Работа с файловой базой данных напрямую

Работа с файловой базой данных напрямую возможна с помощью тонкого клиента или толстого клиента. Если используется толстый клиент, то он сам реализует всю функциональность файловой СУБД.



Если используется тонкий клиент, то на компьютере, где запущен сам тонкий клиент, организуется специализированная серверная среда.



В рамках этой специализированной среды выполняются:

- загрузка необходимых для работы системы серверных компонентов,
- загрузка конфигурации,
- другие действия, необходимые для организации нормальной работы системы с информационной базой.

С точки зрения тонкого клиента, данная среда выступает в роли сервера. С точки зрения операционной системы, данная специализированная среда не выделена в отдельный процесс и выполняется в рамках процесса тонкого клиента.

Работа с файловой базой данных через веб-сервер

Работа с файловой базой данных через веб-сервер возможна с помощью тонкого клиента или веб-клиента.

В этом случае модуль расширения веб-сервера создает аналогичную серверную среду на веб-сервере для каждой информационной базы.



Прямо в адресное пространство веб-сервера загружается компонент для работы с файловой базой данных и сами данные информационной базы. При этом нагрузка на веб-сервер значительно возрастает, а пользователи одной информационной базы не имеют возможности работать параллельно. Все их запросы к базе данных выстраиваются в одну очередь.

По этой причине такой вариант работы является скорее тестовым. Чтобы, например, попробовать, как информационная база работает через веб-сервер, с веб-клиентом. В качестве рабочего такой вариант, наверное, можно использовать в особенных случаях для очень небольших рабочих групп.

Задание 1. Распишите достоинства и недостатки файловой системы

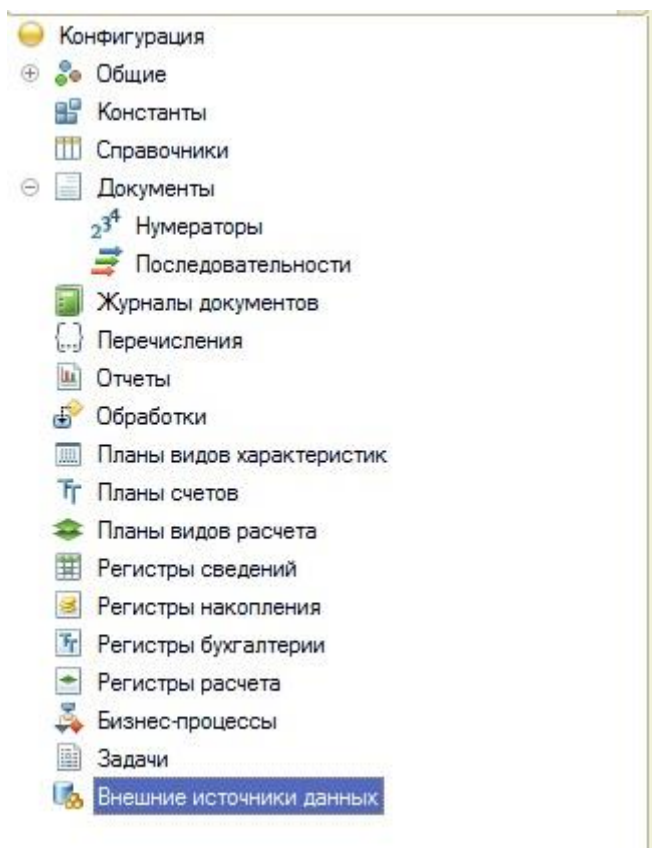
название	достоинства	недостатки
Файловая система		

Лекция 28: Методы, используемые объектом

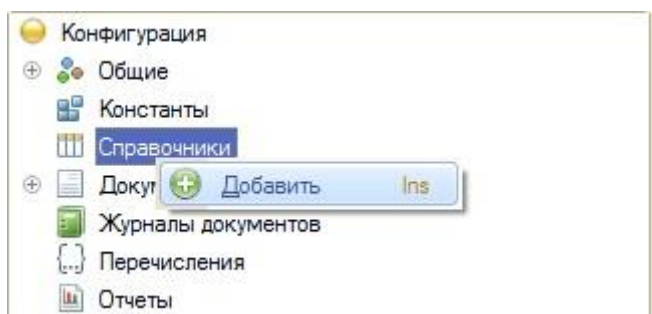
Объекты конфигурации 1С или объекты метаданных создаются на этапе конфигурирования разработчиком — это справочники, документы, константы и т.п. Пользователь может непосредственно работать с некоторыми объектам — создавать экземпляры объектов и манипулировать ими. Или использовать их — отчеты, обработки и тп.

В платформе 1С существуют различные прототипы объектов метаданных. Это справочники, документы и прочие метаданные.

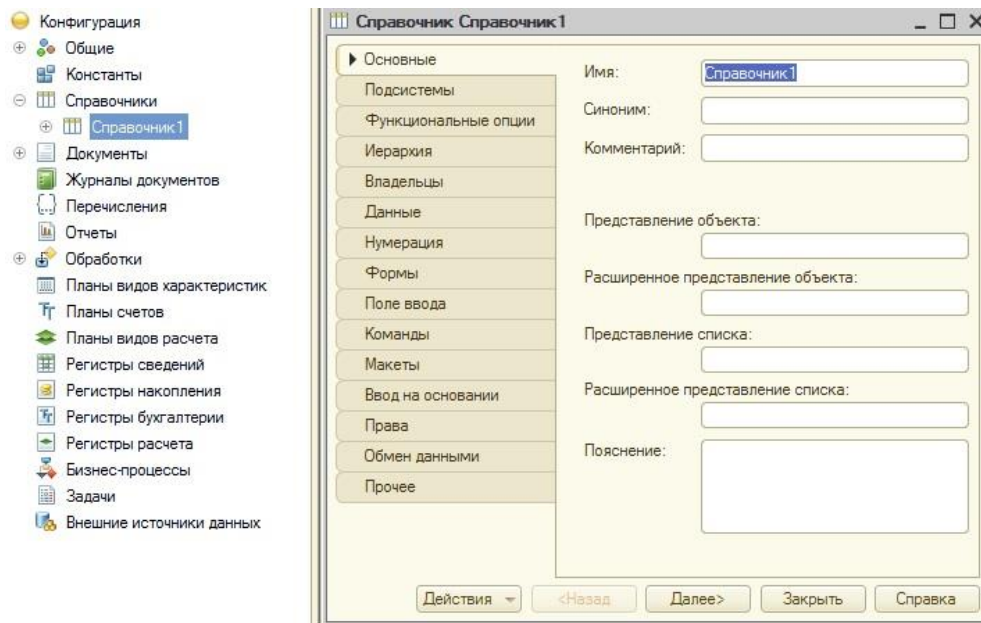
Все прототипы объектов метаданных представлены в дереве метаданных.



Разработчик может самостоятельно создать объект того или иного прототипа. Например, мы можем создать Справочник. Для этого, достаточно выделить ветку Справочники дерева конфигурации, вызвать правой клавишей мышки контекстное меню, и выполнить команду «Добавить» этого меню.



После выполнения команды, будет создан новый объект, с заранее определенным разработчиками платформы 1С набором методов и свойств.

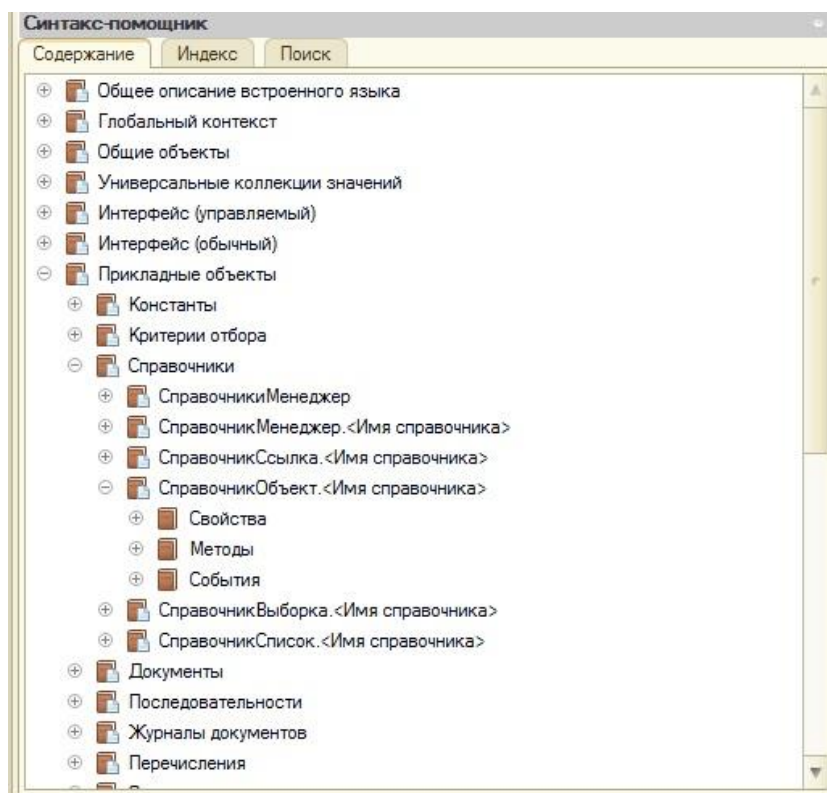


Например, у справочника, созданного выше, одни из свойств это Имя, Синоним и т.д., а методы, например, Записать, Заполнить и т.д.

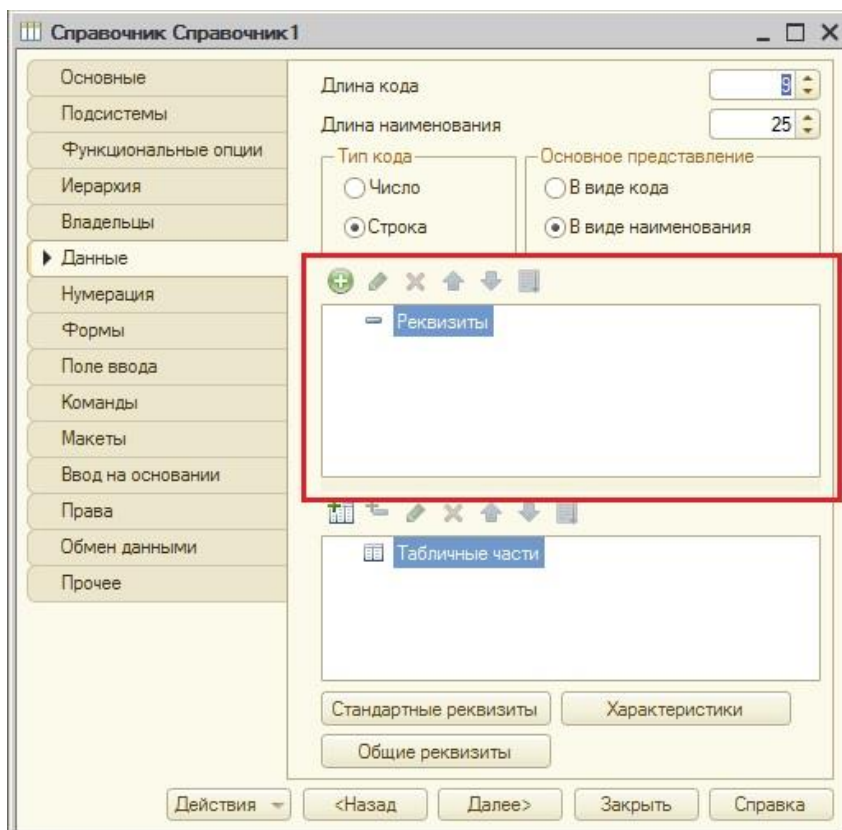
Также, у объекта могут быть события: это возможность перехватить какие-то действия с объектом, например, можно перехватить запись элемента справочника.

Набор методов, свойств и событий объекта определяется его прототипом. Например, у справочников может быть один набор свойств, а у документов совершенно другой.

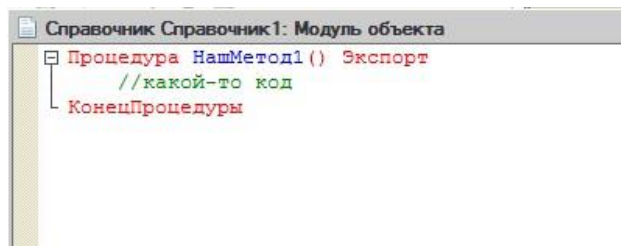
Посмотреть все методы, свойства и события прототипа того или иного объекта можно в синтакс-помощнике. В ветке для соответствующего прототипа.



Для объектов метаданных разработчик может самостоятельно создать те или иные свойства – реквизиты объектов.



А также, разработчик может создать методы – описав их в модуле объекта.

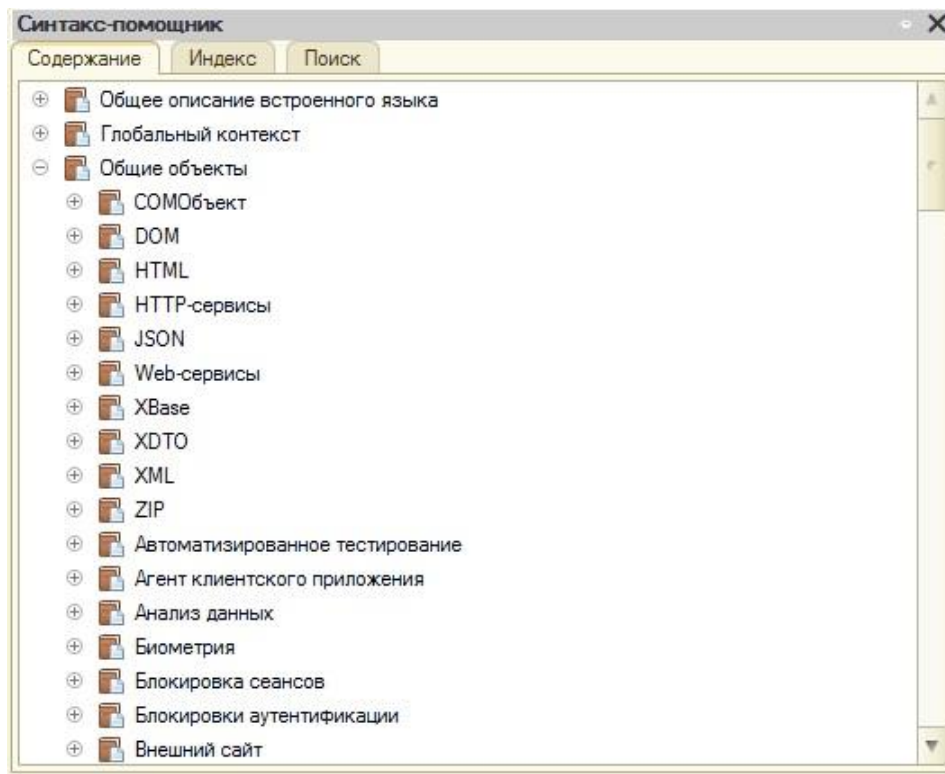


У некоторых объектов пользователь может создавать экземпляры этих объектов: например, можно создать экземпляр справочника или документа, и записать этот экземпляр в базу данных. А некоторые объекты метаданных пользователь может использовать для каких-то целей. Например, отчет или обработку.

ОБЩИЕ ОБЪЕКТЫ 1С

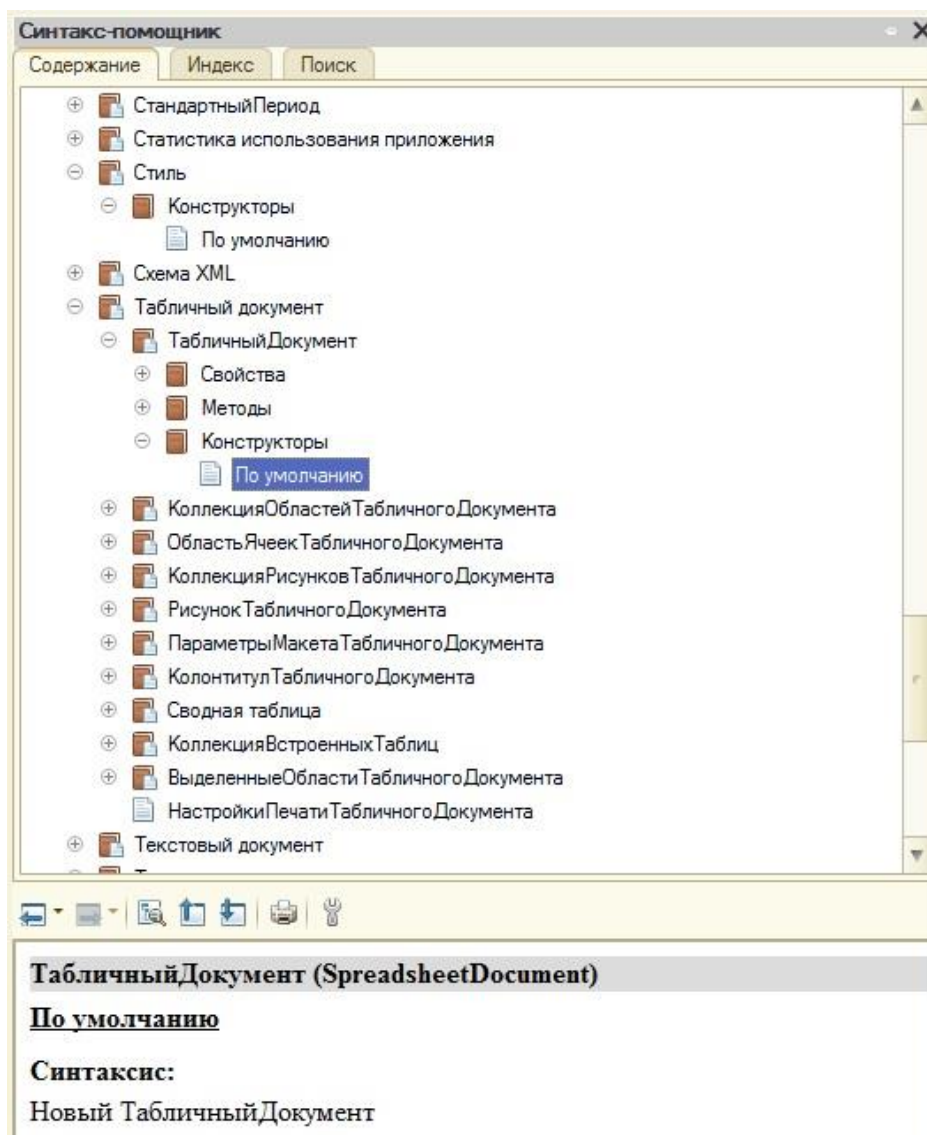
Общие объекты – это объекты встроенного языка программирования 1С, которыми может оперировать разработчик, необходимые для различных технических и вспомогательных нужд (например, такой общий объект, как ДиалогВыбораФайла предназначен для работы с диалогом открытия файла, сохранения файла и т.д.).

В синтакс-помощнике информация обо всех общих объектах расположена в одноименном каталоге.



Если у объектов метаданных были прототипы, и разработчик мог создать тот или иной объект какого-либо прототипа, то у общих объектов ни каких прототипов нет, и разработчик может довольствоваться только теми общими объектами, которые ему предоставляет платформа 1С.

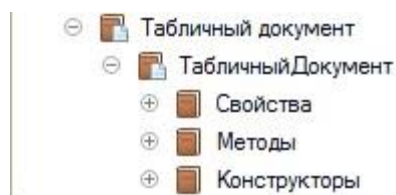
Разработчик имеет возможность самостоятельно создать только экземпляр какого-то объекта, для своих утилитарных целей. Делается это при помощи конструктора. Какой у того или иного объекта конструктор, можно посмотреть в синтакс-помощнике в одноименном каталоге.



Как правило, большинство общих объектов создается при помощи конструктора *Новый*. Например, так создаётся табличный документ.

НашТабличныйДокумент = Новый ТабличныйДокумент;

У общих объектов, также как и у объектов метаданных, могут быть свойства, методы и события. Но, в отличие от объектов метаданных, мы не можем, как разработчики, самостоятельно конструировать те или иные свойства и методы, а можем оперировать только тем, что предоставила платформа 1С. О том, какие свойства, методы и события имеются у того или иного общего объекта можно посмотреть во всё том же синтаксис-помощнике.



Не у всех общих объектов есть и методы, и свойства, и события. У некоторых объектов может не быть событий, а у некоторых свойств.

Обратиться к свойствам и методам можно через оператор «.»:

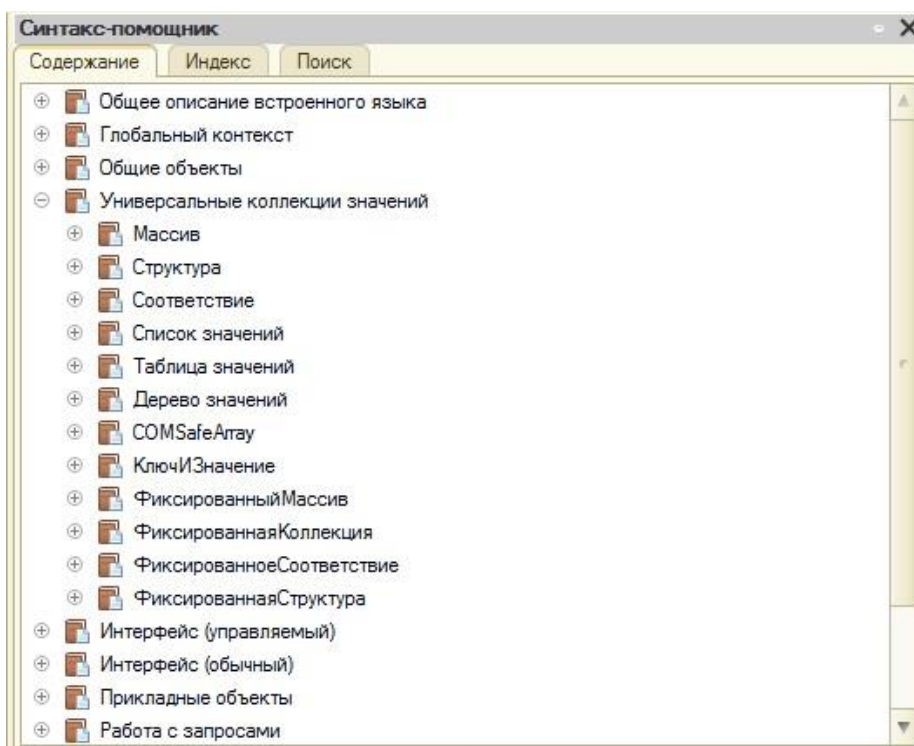
НашТабличныйДокумент = Новый ТабличныйДокумент;

НашТабличныйДокумент.Защита = Истина;

НашТабличныйДокумент.Показать();

УНИВЕРСАЛЬНЫЕ КОЛЛЕКЦИИ ЗНАЧЕНИЙ

Универсальные коллекции значений – это, также как и общие объекты, объекты встроенного языка программирования, которыми может оперировать разработчик. Но, в отличие от общих объектов, они предназначены для хранения наборов данных, причем эти наборы данных хранятся в оперативной памяти компьютера, во время запущенного сеанса. Таким образом, когда пользователь прекратит работу с сеансом 1С, то и вся информация этих наборов данных будет уничтожена.

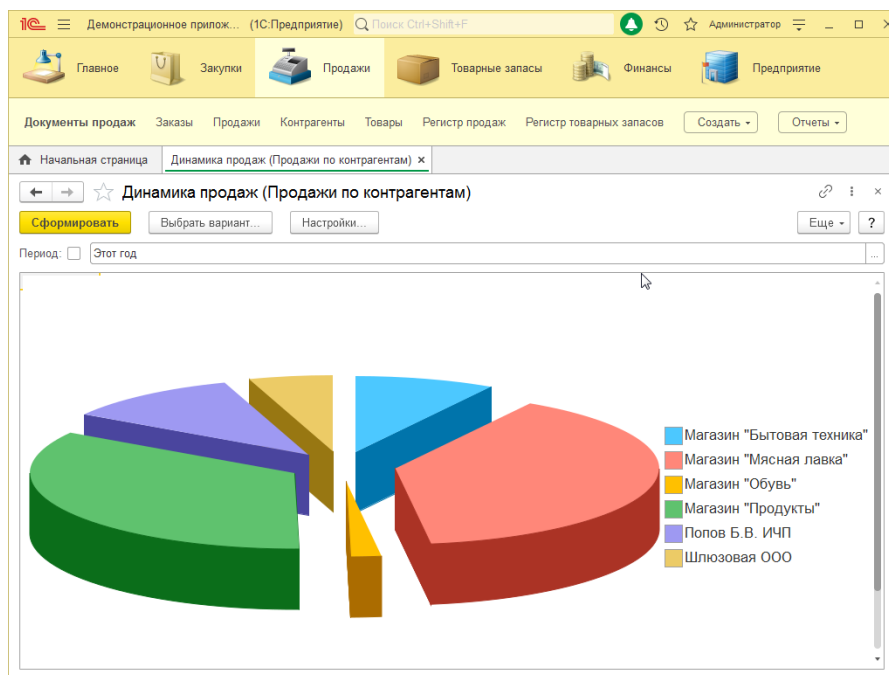


Все универсальные коллекции описаны в синтакс-помощнике в разделе «Универсальные коллекции значений». Это массив, структура, соответствие, список значений, таблица значений и т.д.

Как правило, все универсальные коллекции значений создаются при помощи конструктора Новый.

Лекция 29: Объект Диаграмма

Диаграмма — это объект встроенного языка, позволяющий отображать данные в графическом виде, удобном для анализа.

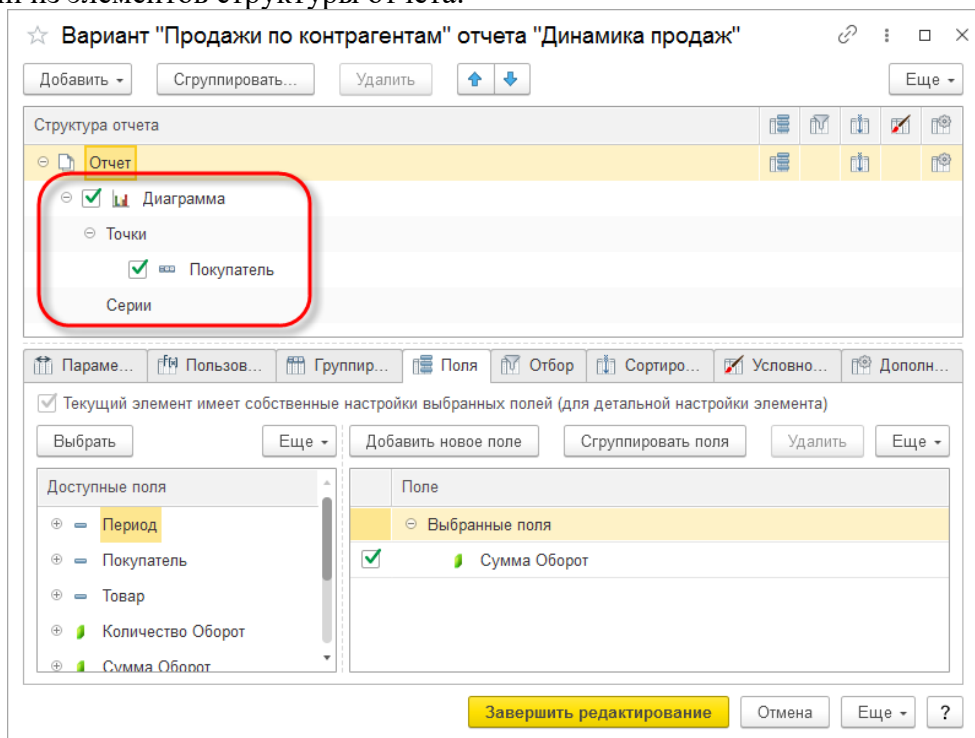


Платформа позволяет создавать диаграммы разных типов. Каждый из них удобен для представления тех или иных данных:

- График: обычный, с накоплением, по шагам, с областями, с областями нормированный, с областями и накоплением;
- Гистограмма вертикальная или горизонтальная: обычная, с накоплением, нормированная, объемная, с накоплением объемная, нормированная объемная;
- Круговая обычная и объемная;
- Кольцевая обычная и объемная;
- Биржевая обычная и «свеча»;
- Изометрическая: обычная, непрерывная, лента, пирамида;
- Водопад;
- Поверхность: обычная, каркасная, выпуклая, вогнутая, затенённая;
- Сотовая;
- Радарный график: обычный, с областями, с накоплением, с областями и накоплением, с областями нормированный;
- Измерительная;
- Воронка: обычная, объемная, нормированная, нормированная объемная;
- Точечная;

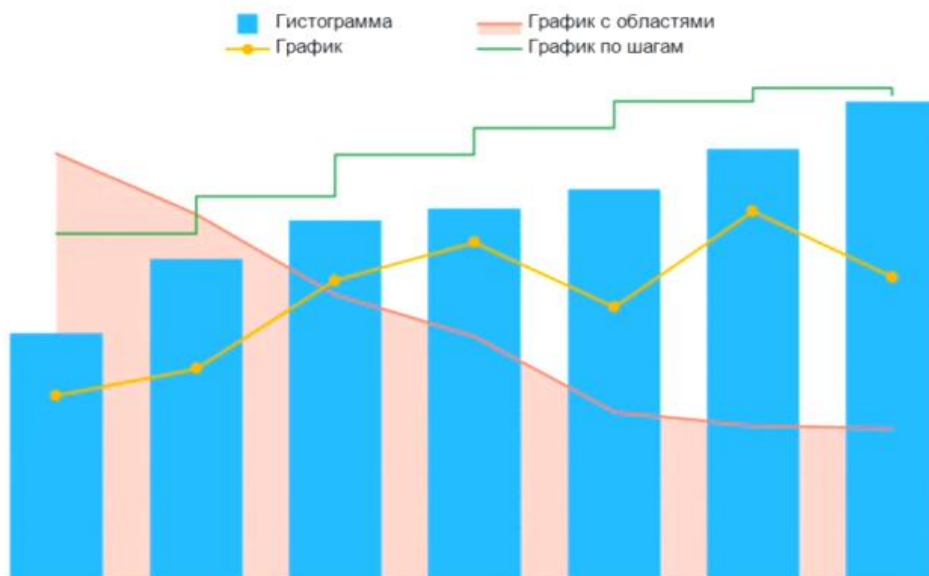
- Пузырьковая.

В отчётах, использующих систему компоновки данных, диаграмма описывается визуальными средствами как один из элементов структуры отчёта:



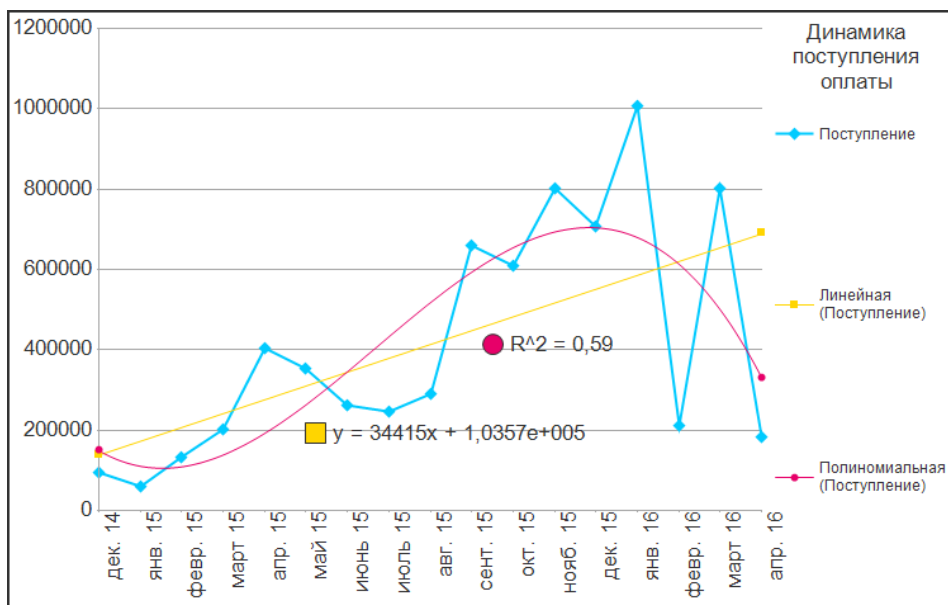
Комбинированные диаграммы

Можно совмещать разные виды диаграмм. Такие диаграммы называются комбинированными, они могут содержать одновременно график и гистограмму. Существует возможность вывода на них дополнительной шкалы значений.



Линии тренда

В диаграммах поддерживаются линии тренда. У одной серии их может быть несколько. Вместе с линией тренда можно отображать её уравнение и коэффициент детерминации.

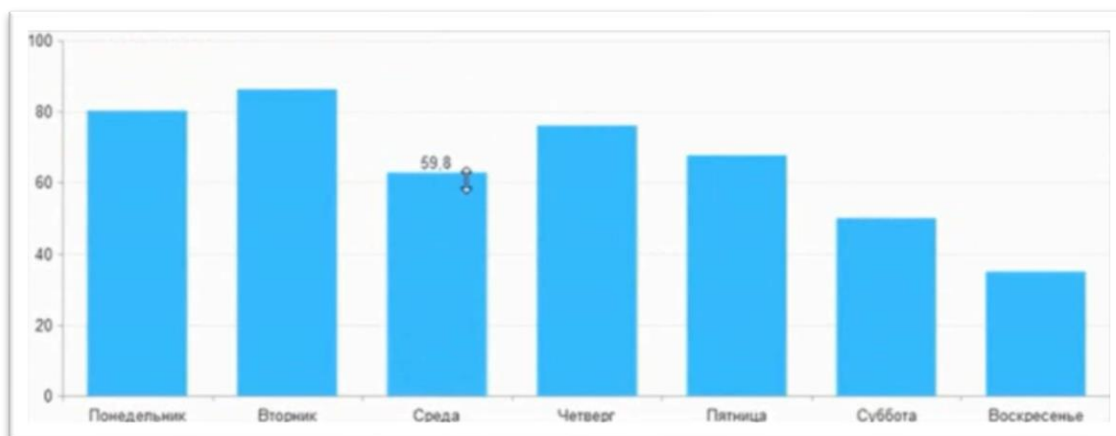


Анимация

Реализована автоматическая анимация при выводе диаграмм. При первоначальном отображении для каждого типа диаграммы платформа выбирает случайный вариант анимации из нескольких возможных. При необходимости анимация может быть отключена.

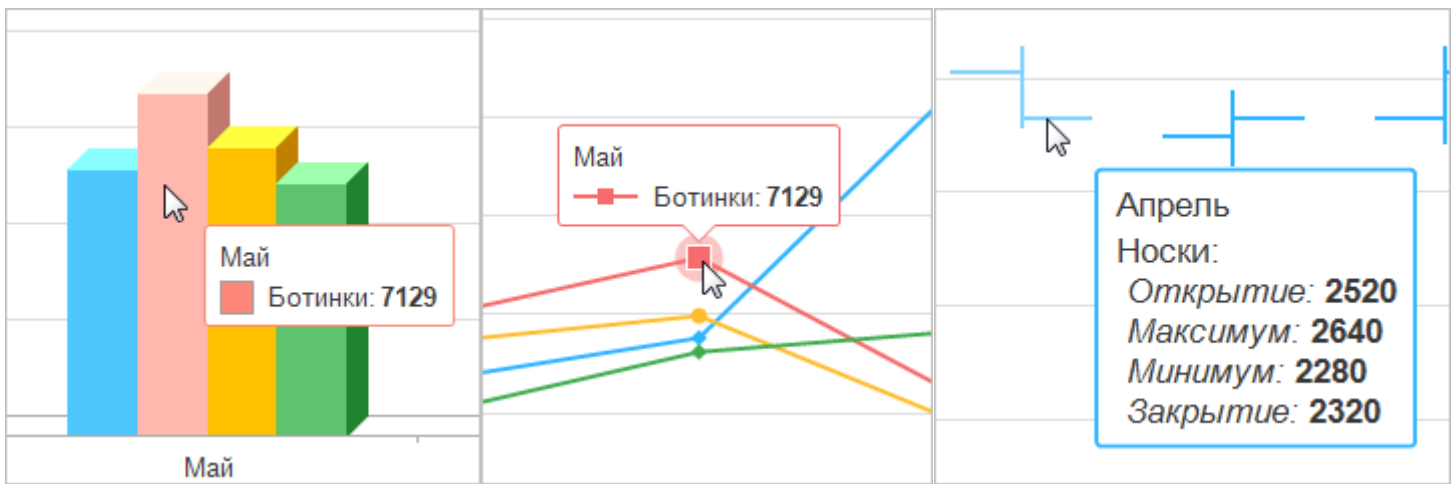
Интерактивное редактирование значений

Значения диаграмм можно редактировать интерактивно, просто перетаскивая их мышью. Во встроенном языке существует возможность обрабатывать такие изменения, ограничивать диапазон изменений или запрещать интерактивное редактирование совсем.

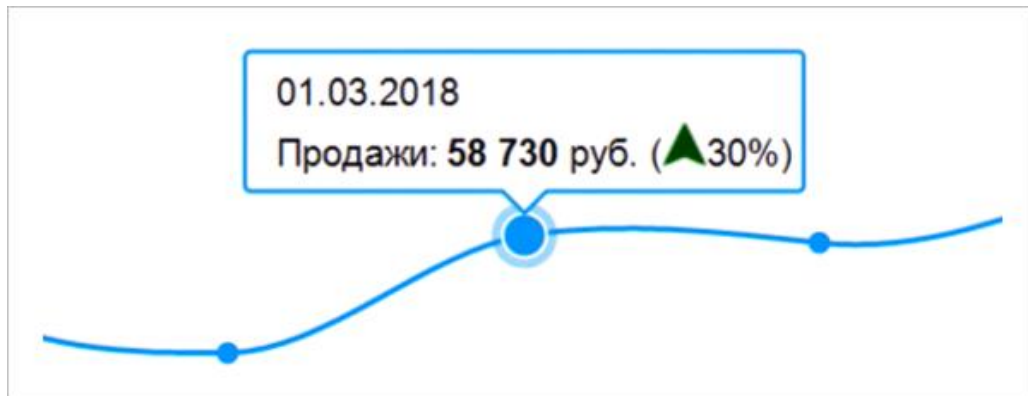


Всплывающие подсказки

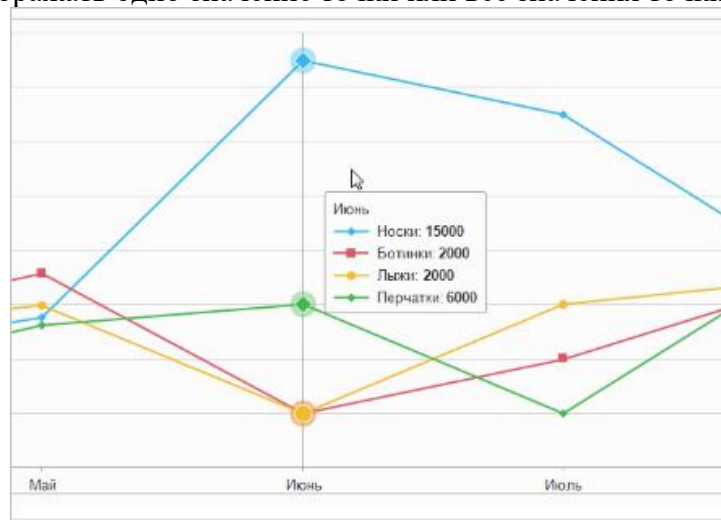
Всплывающие подсказки отображаются почти мгновенно, а их внешний вид немного меняется в зависимости от типа диаграммы.



В подсказку можно установить форматированную строку и с её помощью вывести текст и добавить картинки.



Подсказка может отображать одно значение точки или все значения точки во всех сериях.



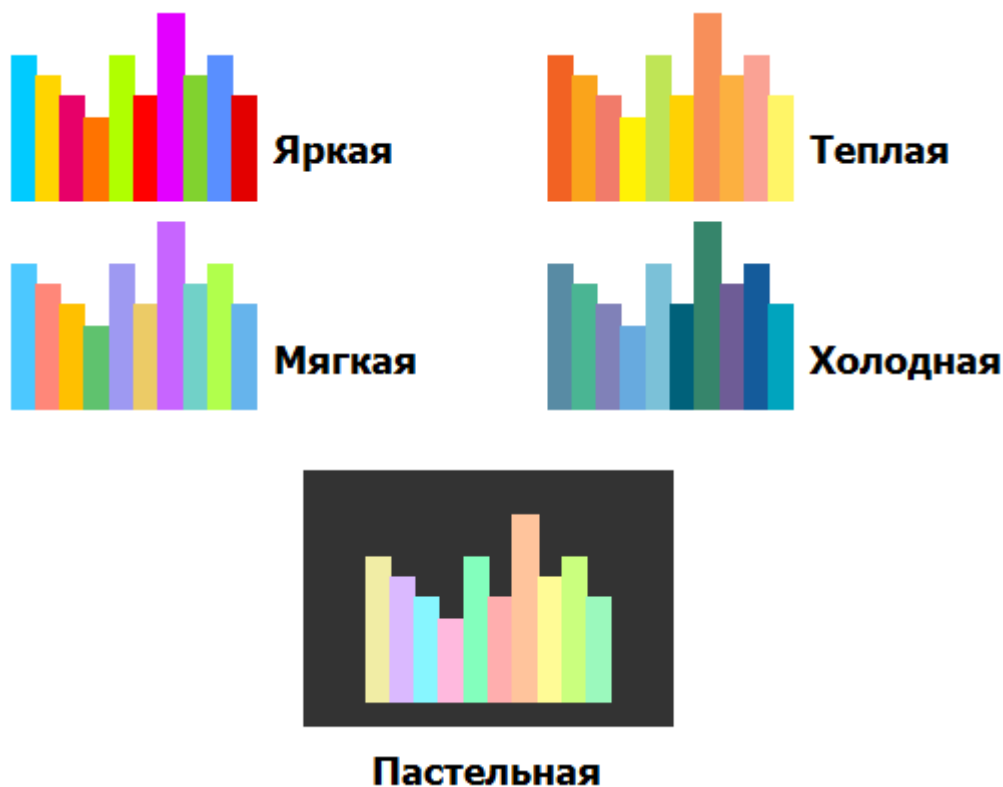
Существуют два режима отображения подсказок:

- **Чувствительный**, когда подсказка отображается постоянно и для того значения, которое находится ближе всего к указателю.
- **При наведении**, когда подсказка появляется только тогда, когда указатель мыши устанавливается над каким-нибудь значением.

Палитры цветов

Поддерживается использование большого количества predetermined палитр для оформления диаграмм: **Яркая, Мягкая, Пастельная, Теплая и Холодная** и другие. Также поддерживается набор

градиентных палитр и существует возможность автоматического создания собственной градиентной палитры на основе начального и конечного цветов.

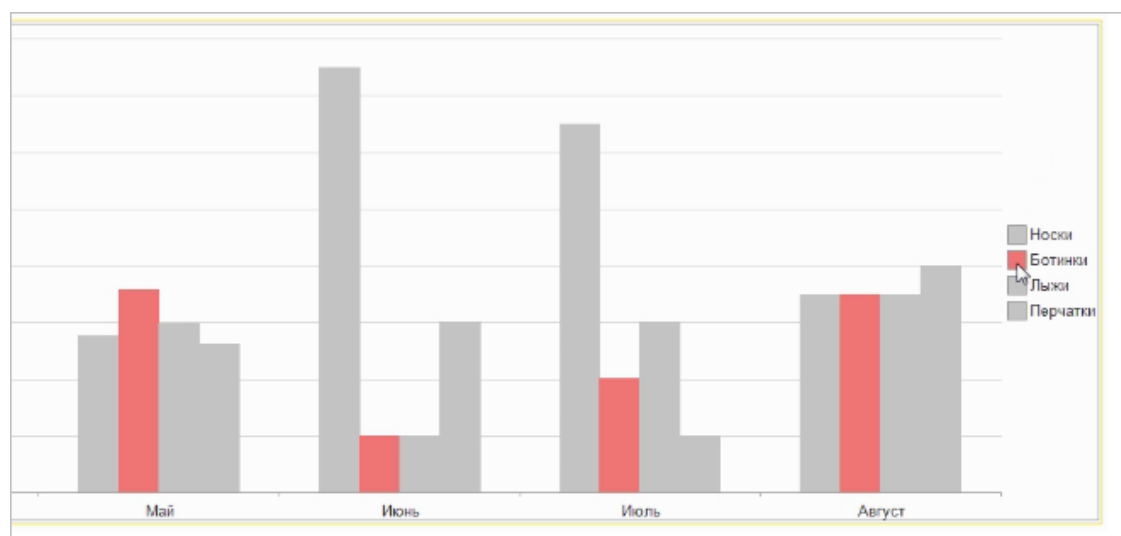


Информационные интервалы и информационные линии

Поддерживаются информационные интервалы и информационные линии. Например, с помощью информационных интервалов, выводимых по значениям, можно выделить в диаграмме область с низкими продажами, область со средними продажами и область с высокими продажами. А с помощью информационной линии, выводимой по точкам, можно обозначить дату начала рекламной компании. Можно задавать подписи информационных линий и информационных интервалов, настраивать их цвет, цвет фона, шрифт и так далее. Информационные интервалы можно выводить не на всю область построения диаграммы, а только на её часть. Это позволяет создавать «вложенные» интервалы, уточняющие более крупные.

Подсветка серий

Серии подсвечиваются, когда курсор мыши проходит над ними в легенде. Эта возможность упрощает анализ больших диаграмм.



Лекция 30: Работа с сериями диаграммы

Диаграмма предоставляет удобное графическое представление некоторого набора данных, чаще всего результатов каких-либо изменений. 1С предоставляет собственный (надо сказать, приятный) инструмент такого графического представления - объект "Диаграмма". Область диаграммы состоит из нескольких частей: область построения (там создается рисунок), заголовка (надписи над рисунком), легенды (расшифровки меток серий). В диаграмме имеются серии (т.е. наборы значений) и точки, из которых состоят серии (см. рис.1).



Рис. 1 Диаграмма в 1С

Создание диаграммы

Диаграмма вставляется в таблицу печатной формы с помощью пункта Вставить рисунок меню Таблица. Далее необходимо сделать следующие действия: определить тип диаграммы (гистограмма, график, круговая и т.п.), настроить внешний вид составных частей диаграммы, а затем описать процедуру "наполнения" данными при выводе печатной формы. Пройдемся поподробнее по каждому из этих действий.

Задание типа диаграммы

Для отображения разных наборов данных имеются более или менее удачные типы диаграмм. Задача программиста при конфигурировании определить тип будущей диаграммы, и задать его при форматировании области построения (контекстное меню области - закладка Тип диаграммы). Задав тип и закрыв окно форматирования, можно увидеть пример построенной диаграммы.

Настройка внешнего вида диаграммы

Для того чтобы включить режим форматирования областей диаграммы, необходимо выбрать пункт Изменить подменю Объект: 1С. V7. Диаграмма меню Действия (то же пункт есть в контекстном меню диаграммы). Каждая из составных частей диаграммы в режиме редактирования имеет пункт контекстного меню Формат области ... Настройка внешнего вида состоит из выбора заливки, рамки, цвета и шрифта текста и т.п. Там же можно определить, например, сетку для области построения. После завершения форматирования достаточно щелкнуть по любой ячейке таблицы, чтобы выключить режим форматирования.

Наполнение данными диаграммы

Диаграмма как объект системы 1С принадлежит таблице, в которую она вставлена. Обращение к объекту происходит в момент вывода на печать таблицы (метод Вывести()) или секции таблицы (метод ВывестиСекцию()), в которую вставлена диаграмма. Для того, чтобы при обращении были выполнены некоторые действия по подготовке данных для диаграммы, в свойствах объекта на закладке "Текст" необходимо указать строку вызова соответствующей процедуры, передав ей в качестве параметра контекст диаграммы как текущего объекта таблицы. Строка может выглядеть, например, так:

ОбработкаДиаграммы(Таб.ТекущийОбъект)

где: ОбработкаДиаграммы - имя процедуры, Таб - имя выходной таблицы (должна быть предварительно создана методом СоздатьОбъект()).

В модуле описывается процедура с идентификатором, через который будет производиться к контексту диаграммы:

Процедура СформироватьДиаграмму(Диаграмма1)
где Диаграмма1 - идентификатор, в который передается контекст объекта Диаграмма. Все методы объекта записываются через точку после идентификатора.

Основы программирования

Изменение заголовка таблицы

Диаграмма1.Заголовок = "Поступления от покупателей";

Работа с сериями диаграммы

//установим количество серий. Каждая при этом получит свой номер.

Диаграмма1.КоличествоСерий(3); // так у нас будет три серии

//однако если в другом методе будет передана серия с большим номером,

//количество серий будет автоматически увеличено.

КолСерий = Диаграмма1.КоличествоСерий();

// а так переменной КолСерий будет передано текущее количество серий

// так мы зададим имя "Рубли" серии № 1

Диаграмма1.УстановитьИмяСерии(1, "Рубли");

// установим цвет серии № 2

// этот метод не обязательный, как впрочем и некоторые другие

Диаграмма1.ЦветСерии(2, 35, 67, 90);

// установили: 35 единиц красной компоненты, 67 - зеленой, 90 - синей

//можно установить имена серий автоматически: Серия1, Серия2, и т.д.

Диаграмма1.АвтоУстановкаИменСерий(1);

//если поставить вместо флага 1 флаг 0, автоустановка имен будет запрещена.

Работа с точками диаграммы

//установим количество точек, скажем, 8 штук

Диаграмма1.КоличествоТочек(8);

//однако если в другом методе будет передана точка с большим номером,

//количество точек будет автоматически увеличено.

// метод может вернуть текущее количество точек

КолТочек = Диаграмма1.КоличествоТочек();

// можно установить имя любой точки

Диаграмма1.УстановитьИмяТочки(7, "7 точка");

//можно установить имена точек автоматически: 1, 2, 3, и т.д.

Диаграмма1. АвтоУстановкаИменТочек(1);

//если поставить вместо флага 1 флаг 0, автоустановка имен будет запрещена.

Передача значений точкам серий

Это самый важный метод диаграммы. Он устанавливает значение серии <НомерСерии> в точке <НомерТочки> равным <Значение> с возможностью последующей расшифровки. Все параметры, кроме параметра <Расшифровка> должны быть числовыми выражениями.
Синтаксис:

УстановитьЗначение(<НомерТочки>,<НомерСерии>,<Значение>,<Расшифровка>)

<Расшифровка> - параметр не обязательный, может иметь любой тип. Это значение, которое будет передано предопределенной процедуре ОбработкаЯчейкиТаблицы при двойном щелчке мышью на точке диаграммы. Так можно получать расшифровки точек диаграммы. Удобная штука!

Два вспомогательных метода диаграммы

Запрет перерисовки может ускорить работу системы во время наполнения таблицы данными.

Диаграмма1.Обновление(0);

//то же выражение, но с флагом 1 разрешит перерисовку

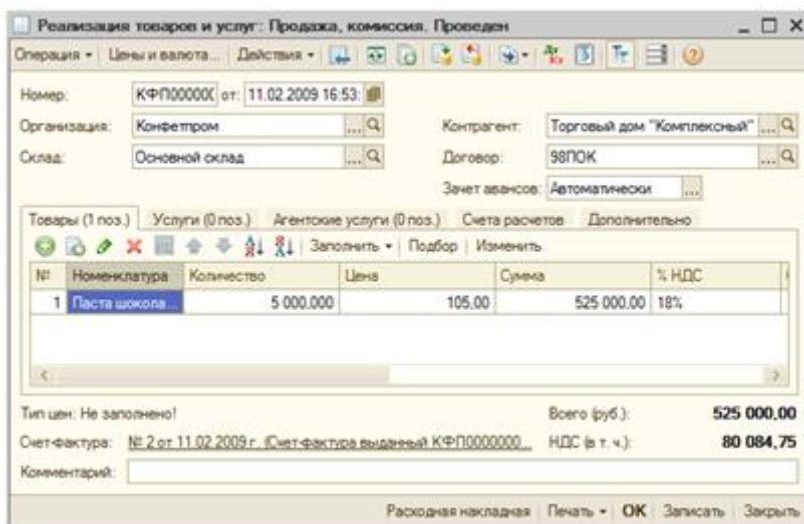
Очистка всех установленных значений имен и количеств точек, серий, диаграммы делается так:

Диаграмма1.Очистить();

Лекция 31: Форма

Разработка интерфейса в 1С состоит из двух частей – разработка меню или рабочего стола и разработка экранных форм 1С. Окна, с которыми пользователь работает в 1С, называют экранные формы 1С или просто формы 1С.

Пользователи в программе работают с формами 1С. Мало того — пользователи видят только формы 1С. Поэтому с точки зрения удобства работы в программе это один из важных элементов. При этом убить времени на разработку формы 1С можно больше, чем на программирование самого сложного алгоритма.



Частой ошибкой программистов является попытка нарисовать все по своему вкусу. Сделать фон синим, а надписи зеленым. Или желтый на черном. Или как он ему нравится в какой-то другой его любимой программе.

К сожалению, этот подход ошибочен, так как пользователи привыкли работать с типовыми формами 1С, которых в конфигурации большинство. Рисовать собственный велосипед, помечать его надписями Courier вида «Copyright Vasya Pupkin» — это явно дурной тон.

Что такое Формы 1С

Форма 1С – это метод представления объекта 1С пользователю. Обычно форма представляет собой ряд полей, которые требуется заполнить и набор кнопок (команд меню) для управления. Форма 1С есть у большинства объектов 1С.

В толстом клиенте 1С используются «обычные» формы 1С. Это значит, что программист просто рисует мышкой форму 1С также как это делается в Visual Studio и других фреймворках.

В тонком клиенте 1С и веб клиенте 1С используются управляемые формы 1С. Это значит, что их размер, форму 1С и расположение на них полей нельзя изменить мышкой. Они генерируются автоматически на основании настроек. Про управляемые формы 1С мы поговорим в следующих уроках.

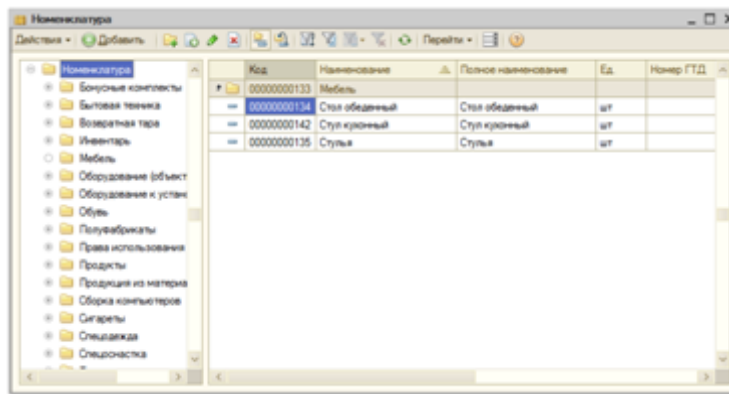
Большинство форм 1С типовых конфигураций в 1С имеют свое, типовое представление, привычное пользователю.

- **Форма списка 1С**

В 1С есть множество списков. Когда мы говорим «справочник номенклатуры», то подразумеваем «список элементов номенклатуры (т.е. товаров)». Когда мы говорим документы банка, то имеем ввиду «список документов ПКО и РКО».

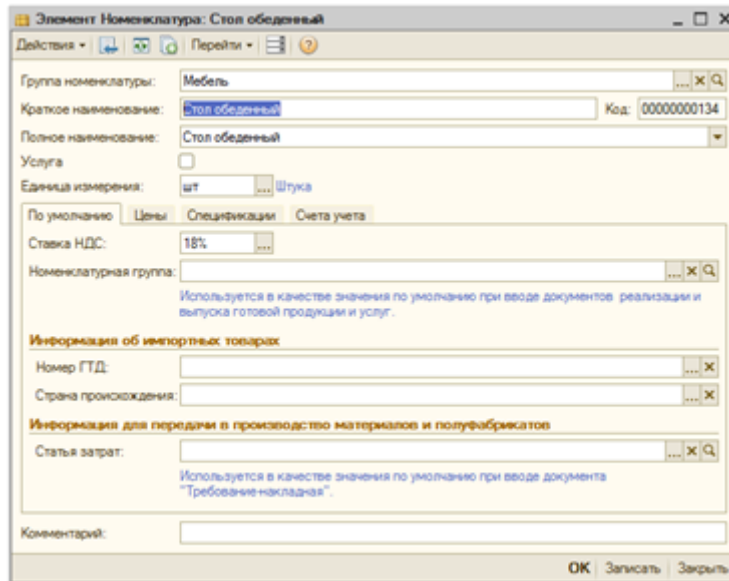
Для отображения списков есть формы списков 1С. Это всегда табличная часть на всю форму 1С и над ней командная панель с кнопками Добавить, Скопировать и т.п., отображенными в виде картинок.

Колонки и их порядок можно посмотреть в любом типовом справочнике и документе.



- **Форма элемента/документа 1С**

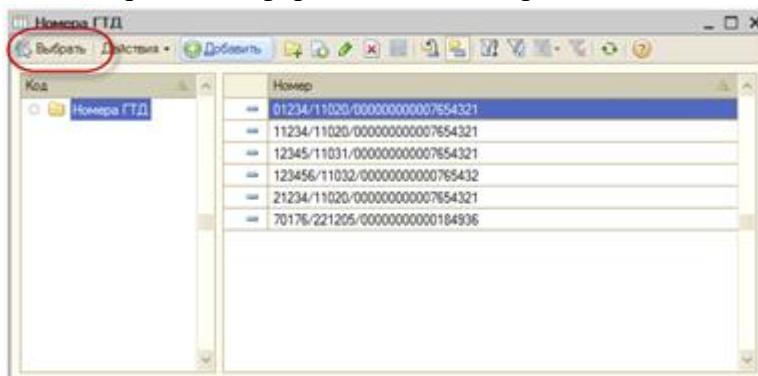
Когда с помощью списка пользователь определился, с каким конкретным документом он собирается работать – открывается форма документа/элемента справочника 1С.



Она обычно состоит из шапки (списка обязательных полей) и закладок с таблицами. Вверху находится также панель с кнопками, но на ней часто бывают текстовые меню Действия, Цены и валюта, Операция.

- **Форма выбора 1С**

Похожа на форму списка 1С. Зачастую форма списка 1С и используется в качестве формы выбора 1С. Она используется в том случае, когда при заполнении документа (или элемента справочника) пользователь нажимает кнопку выбора «...» и открывается форма 1С для выбора значения.



- **Форма отчета 1С**

- о Верхняя командная панель, обычно с кнопкой Настройка, по нажатию на которую открывается вторая форма отчета 1С – форма Настроек
- о Выбор периода «С .. ПО ..», кнопка изменения периода»
- о Одно или несколько полей быстрых отборов (например организация)

- о Большое табличное поле, куда будет выведен отчет
- о Кнопка формирования обычно вверху, крайняя справа «Сформировать».

Номенклатура, базовая единица измерения	Сумма продаж в USD	Сумма продаж без скидок в USD	Количество (в базовой единице)	Сумма
БОБОН морожильная камера, шт	385 557,76	385 557,76	3 519,000	6,000
Ассорти (конфеты), упак	337,15	337,15	80,000	81,000
Барбарис (конфеты), кг	773,34	773,34	26,000	26,000
Белочка (конфеты), кг	390,04	390,04	20,000	20,000
Ботаник женские демисезонные, пара	757,77	757,77	24,000	24,000
Ботаник женские натуральная кока, пара	1 516,12	1 516,12	16,000	16,000
Ботаник мужские, пара	606,22	606,22	24,000	24,000
Вентилятор настольный, шт	410,89	410,89	12,000	12,000
Вентилятор сконный, шт	136,82	136,82	18,000	18,000
Вентилятор ОРБИТА, STERLING, шт	246,85	246,85	94,000	94,000
Грильяж (конфеты), кг	1 915,43	1 915,43	22,000	22,000
Диван для отдыха, шт	13 907,43	13 907,43	26,000	26,000
Женские босоножки, пара	820,77	820,77	18,000	18,000
Женские ботфорты коричневый, пара	5 981,50	5 981,50	54,000	54,000
Женские модельные туфли, пара	6 622,51	6 622,51		

- **Форма обработки 1С**

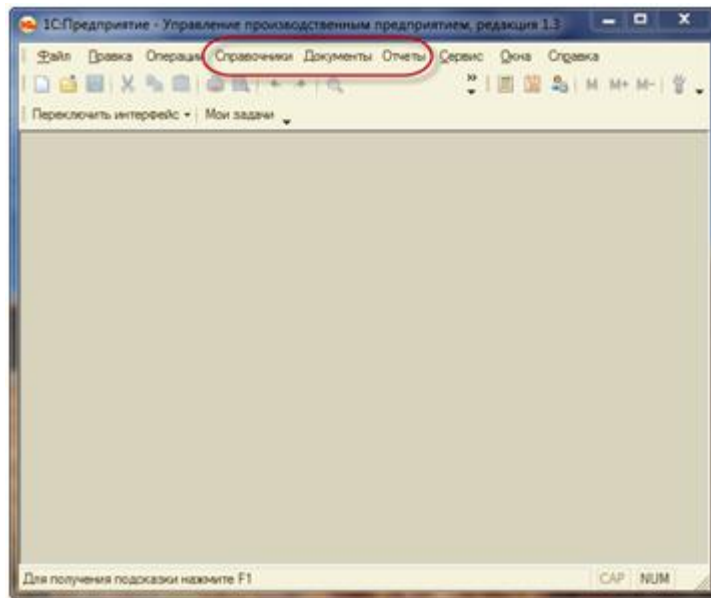
- о Имеет часто две командных панели – вверху и внизу, кнопка запуска и закрытия обычно внизу справа.
- о Может не иметь формы вообще
- о У отчетов и обработок формы списка нет.

Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	В
31.10.2011	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27

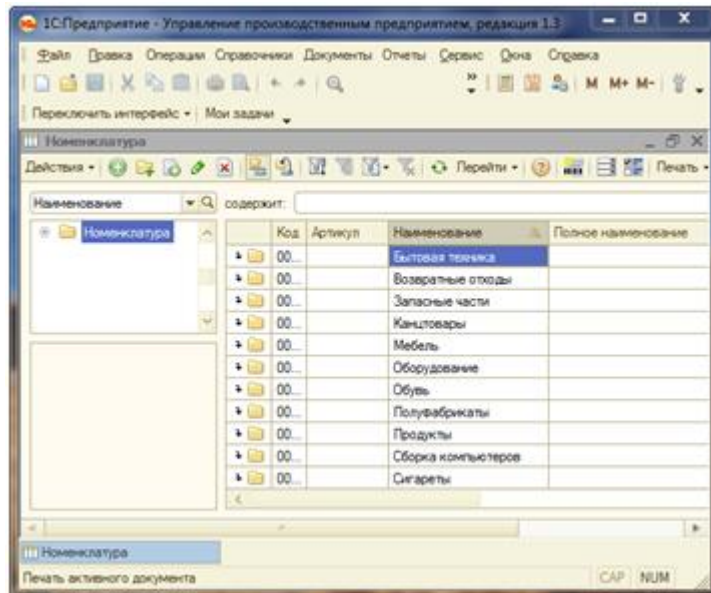
Как работают формы 1С

Логика (порядок) работы пользователя в 1С следующая:

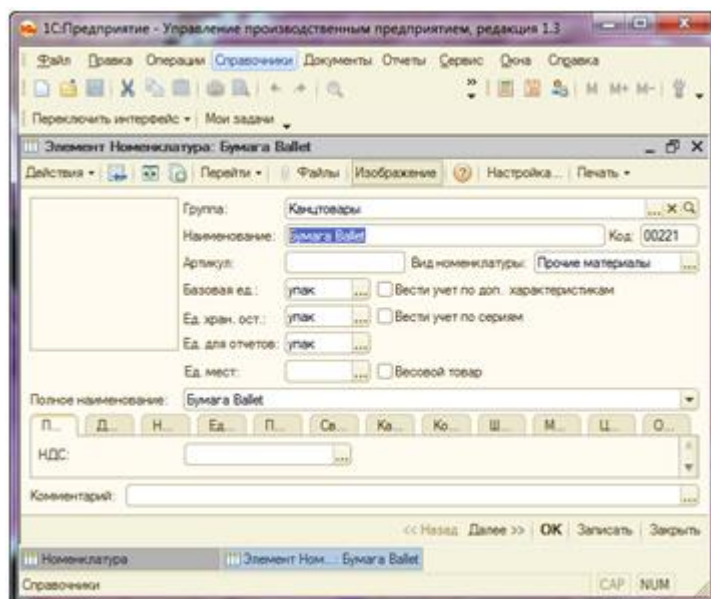
- Пользователь заходит в программу, видит интерфейс (в толстом клиенте обычно это меню, в тонком клиенте рабочий стол)



- Пользователь выбирает пункт меню или гиперссылку на рабочем столе – открывается форма списка 1С



- В форме списка 1С пользователь выбирает команду (например добавить) или существующий элемент списка (элемент справочника или документ) и начинает работать с формой 1С.



- Закончив работать с документом, пользователь закрывает форму документа 1С и возвращается в форму списка 1С. Закончив с ней – возвращается к меню.

Таким образом пользователь работает всегда с формами 1С, начиная с форма списка 1С и переходя к форме элемента 1С. Если программист не нарисовал формы, то 1С генерирует формы по умолчанию. Они естественно лишены идеальной красоты и совершенства, но работать позволяют.

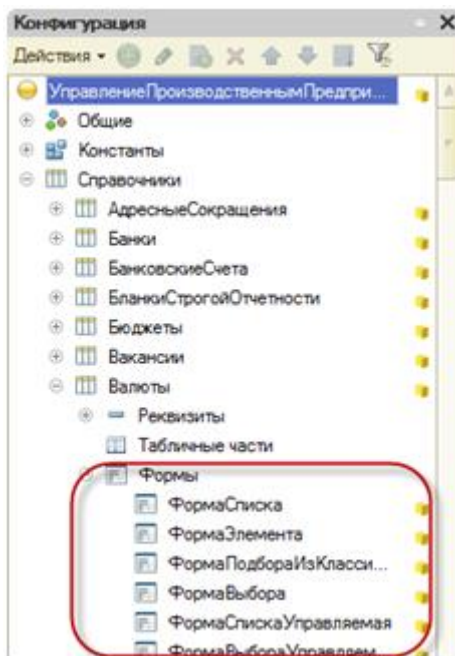
Автоматически сгенерированная форма списка 1С обычно содержит минимум полей (код/наименование и дата/номер соответственно). Автоматически сгенерированная форма элемента обычно содержит все поля (реквизиты), перечисленные сверху вниз.

Задача формы 1С – открыться и ждать действий пользователя. При действии – отреагировать. Таким образом основу модуля формы 1С составляют обработчики событий. Это такие функции, которые вызываются тогда, когда пользователь совершает какое то действие на форме 1С.

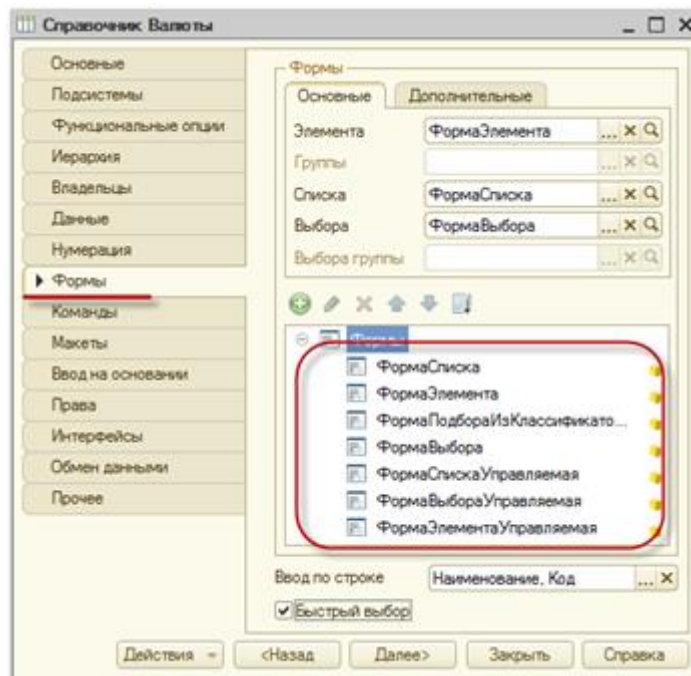
Где находятся формы 1С

В режиме 1С Предприятие при выборе практически любого объекта 1С (справочника, документа, отчета, обработки и т.п.) Вам откроется форма этого объекта.

В конфигураторе, в окне конфигурации, выберите нужный Вам объект, раскройте его ветку, чтобы увидеть вложенную ветку Формы 1С.



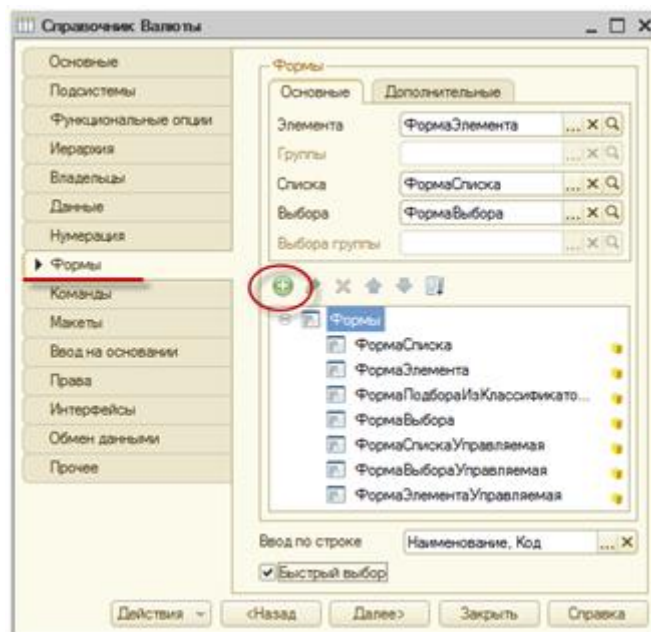
Другой вариант – открыть редактор объекта (два раза мышкой или поставить курсор и Enter) и перейти на закладку Формы 1С.



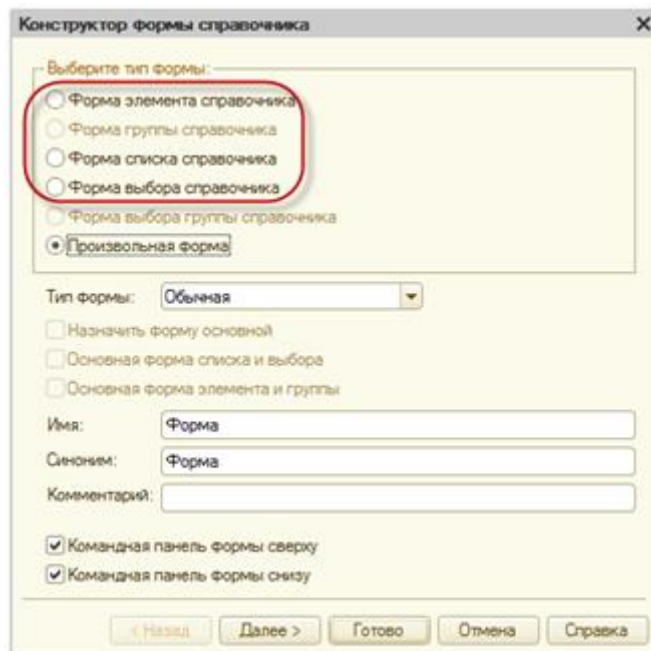
Здесь расположен список форм 1С. Одну из добавленных форм 1С можно добавить как форма по умолчанию (форма списка 1С, форма элемента 1С и так далее).

Создание форм 1С

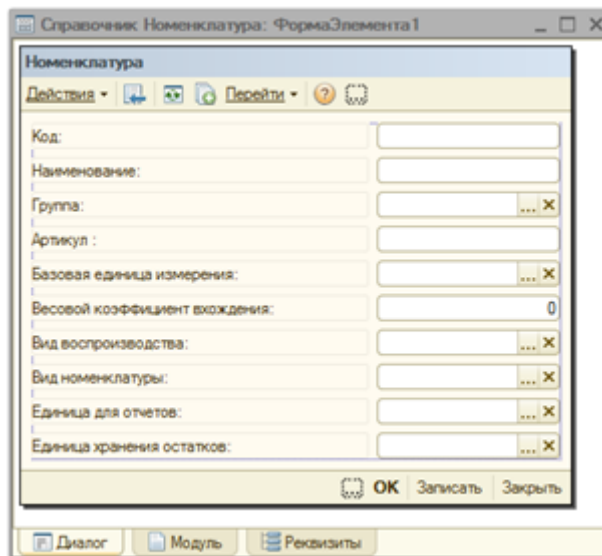
Чтобы добавить новую форму 1С, нужно нажать кнопку Добавить (Ins на клавиатуре). Чтобы войти в существующую – нажмите на нее два раза мышкой.



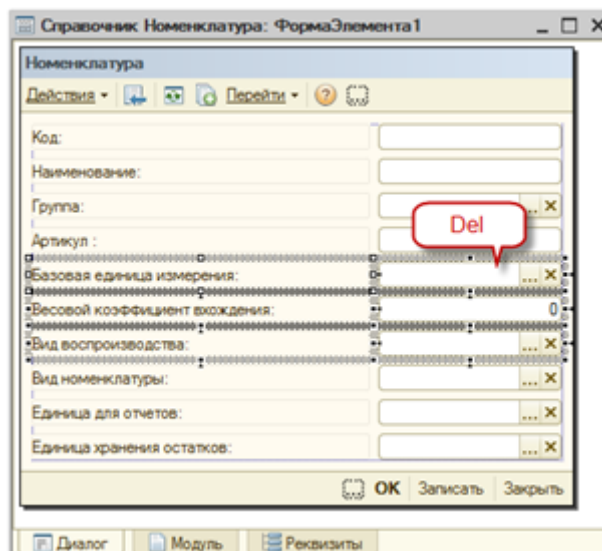
Конструктор предложит Вам выбрать вид формы 1С – форма элемента 1С, списка. Здесь же можно добавить или убрать командные панели на форме 1С. Чаще всего эти настройки оставляются как есть, по умолчанию.



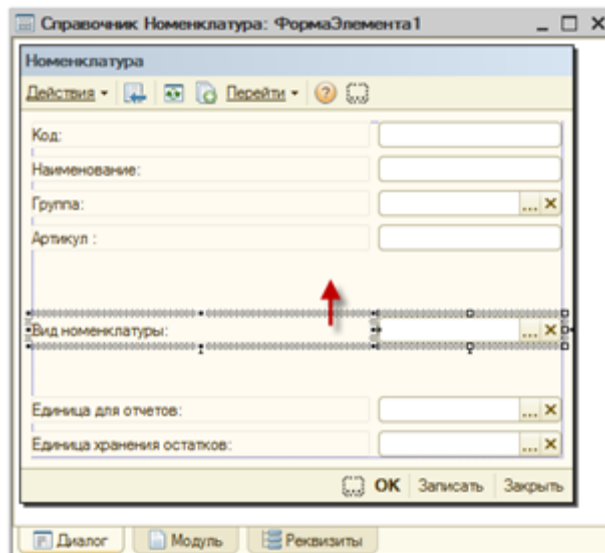
Открывается форма 1С, заполненная по умолчанию — на нее добавились все реквизиты объекта 1С, которые есть. Отметить галочками конкретный список требуемых полей можно на второй закладке конструктора.



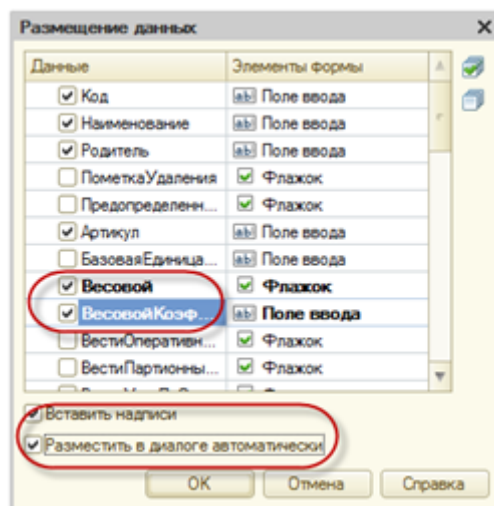
Ненужные реквизиты можно удалить. Для этого выделите одно или несколько полей и нажмите Del.



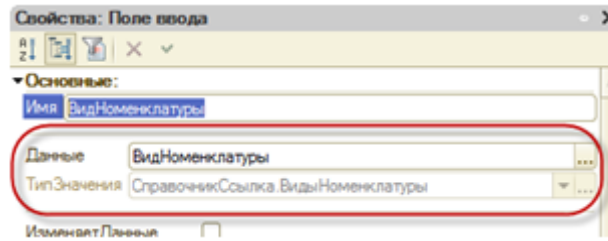
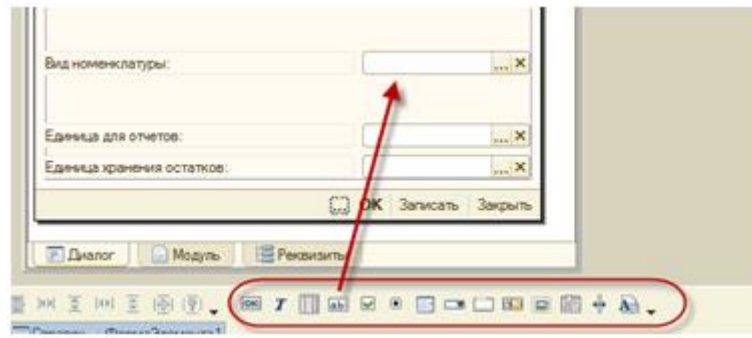
Чтобы в освободившееся место сдвинуть другие реквизиты –выделите их так же и перетащите мышкой.



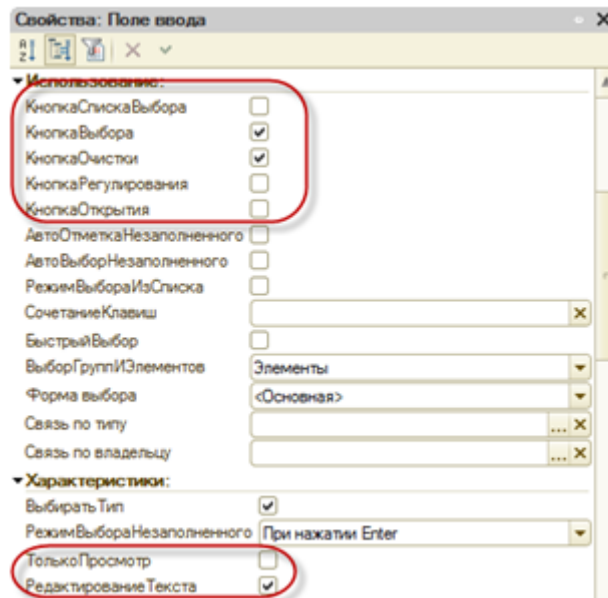
Чтобы добавить на форму 1С новые реквизиты, нажмите кнопку на панели Размещение данных (меню Форма/Размещение данных), установите галочки на тех элементах, которые нужно добавить, а также галочки «Вставить надписи» и «Разместить автоматически».



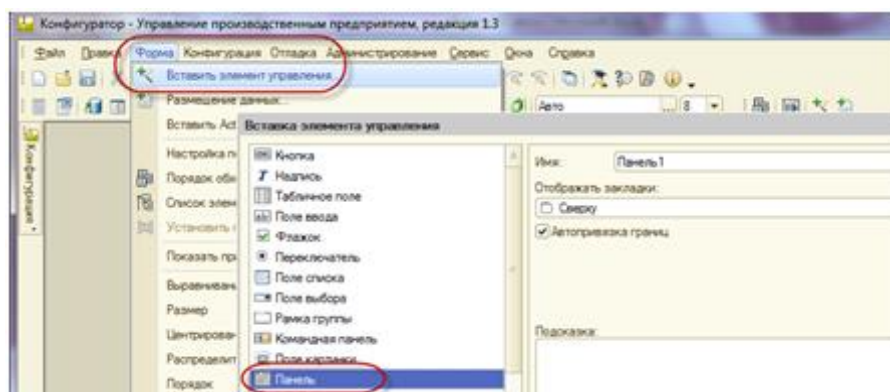
По другому можно просто добавить элемент управления, нажав соответствующую кнопку на панели внизу или выбрав в меню Форма/Вставить элемент управления. Нажмите на элементе управления (поле) два раза левой кнопкой мыши и откроются его свойства. Свойство «Данные» содержит название реквизита объекта 1С. Здесь его можно изменить или назначить.



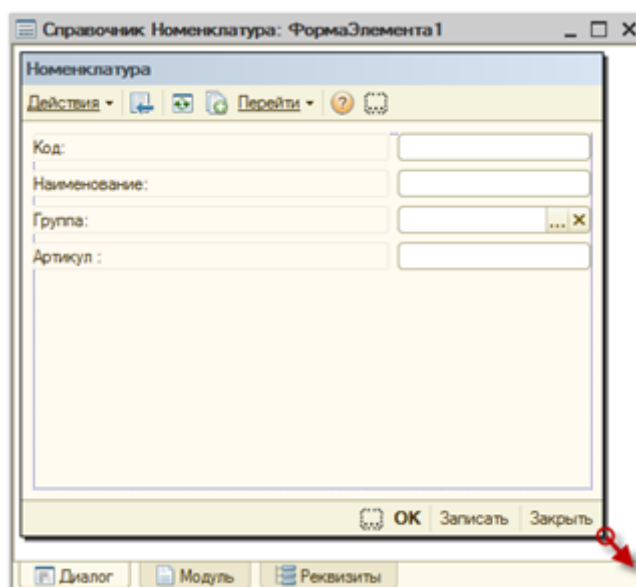
Свойства элемента управления также содержат галочки управления внешним видом элемента управления. С помощью галочек можно включать и отключать кнопки выбора, выпадающий список, очистки, видимости и доступности.



Почти на всех формах документа используются закладки. Добавляется закладка так же, как и другой элемент управления (см. выше), только нужно выбрать элемент управления «Панель». Чтобы добавить страницу панели, нажмите на нее правой кнопкой мыши и выберите «Добавить страницу». Другие элементы управления (поля) на странице панели просто перетаскиваются.



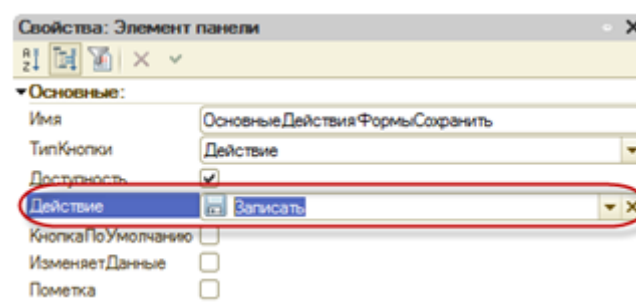
Чтобы изменить размер формы 1С, просто подведите курсор к краю формы 1С, нажмите левую кнопку мышки и просто перетащите край формы 1С.



Чтобы форма 1С заработала – т.е. делала что-то в ответ на действия пользователя – нужно добавить функции обработчики. Войдите в свойства любого элемента (щелкнув на нем два раза левой кнопкой мыши) или самой формы 1С (аналогично на заголовке формы). В самом низу окна свойств есть раздел «События». Выберите удобное событие (для всех полей это обычно «ПриИзменении», для формы «ПриОткрытии») и нажмите кнопку с лупой. Откроется ее обработчик события.



Для кнопок добавление происходит так же. Однако кроме произвольных обработчиков можно выбрать один из стандартных для этой формы (для формы справочника это одни стандартные действия, для формы документа – другие). Просто выберите одно из стандартных действий в свойстве «Действие» или нажмите крестик, если хочется сделать свой обработчик.



Задание 1. Просмотрите видео ролик на тему «Форма» <https://www.youtube.com/watch?v=yCx39EUkzOM>

Задание 2. Написать конспект

Лекция 32: Слои и закладки на форме

Сегодня поговорим о закладках в форме. Что такое закладки - не буду объяснять, надеюсь, все знакомы. Если не знакомы, то после первого же примера - познакомитесь. По этому сразу перейдем к вопросу реализации закладок.

Для работы с закладками используется 2 объекта

1. Объект **Форма**. Этот объект не надо создавать с помощью команды **СоздатьОбъект**. По умолчанию объект **Форма** уже присутствует в форме. Переменную **Форма** описывать тоже не надо. К объекту **Форма** применяется метод: **ИспользоватьЗакладки(Флаг)**, где флаг=1, чтобы разрешить работу с закладками, флаг=0, чтобы запретить использование закладок.
2. Объект **Форма.Закладки**. Этот объект по сути является **Списком Значений**. И к этому объекту применимы *ВСЕ* методы объекта **СписокЗначений**. Добавляя в этот список значений любое значение, Вы тем самым добавляете новую закладку. Представление добавляемого значения используется в качестве имени закладки. Для закладок можно организовать сортировку, динамическую вставку или удаление (например, после установки флажка).

Как правило закладки в форме организуются в предопределенной процедуре **ПриОткрытии()**, но можно и в любой другой процедуре, вызываемой, например, при щелчке по кнопке. Рассмотрим небольшой пример.

Процедура ПриОткрытии()

```
Форма.ИспользоватьЗакладки(1); //разрешили использовать закладки
форма.Закладки.ДобавитьЗначение("А", "Первая"); //добавляем закладки
форма.Закладки.ДобавитьЗначение("Б", "Вторая");
форма.Закладки.ДобавитьЗначение("В", "Третья");
```

КонецПроцедуры

В результате выполнения процедуры будет добавлено 3 закладки. Называться они будут "Первая", "Вторая", "Третья". Добавленные значения А, Б, В - пользователю будут недоступны и по этому эти значения Вы можете использовать по своему усмотрению (или не использовать вообще). После выполнения процедуры закладки в форме появятся сразу. Их можно будет переключать - система обеспечивает это автоматически. Но содержимое Вашей формы не будет меняться - это Вы должны обеспечить сами. В модуле формы, где расположены закладки, Вы должны использовать предопределенную процедуру **ПриВыбореЗакладки(пар1, пар2)**. Эта процедура вызывается системой автоматически при смене пользователем закладки. При этом пар1 и пар2 принимают определенные значения. Пар1 - номер закладки, выбранной пользователем; Пар2 - значение закладки - обратите внимание ЗНАЧЕНИЕ, а не представление. В процедуре Вы должны описать как должна измениться форма при выборе закладки. Приведем простейший пример процедуры

Процедура ПриВыбореЗакладки(номер, зн)

```
Если ЗН="А" тогда
    сообщить ("Вы выбрали закладку А");
    ////
    //... действия при выборе закладки А
    ////
иначеЕсли ЗН="Б" тогда
    сообщить ("Вы выбрали закладку Б");
    ////
    //... действия при выборе закладки Б
    ////
```

иначеЕсли ЗН="В" тогда

```
сообщить ("Вы выбрали закладку В");
```

```
////
```

```
//... действия при выборе закладки В
```

```
////
```

```
конецЕсли;
```

```
конецПроцедуры
```

Очевидно, что действия при выборе закладки, чаще всего приводят к тому, что часть реквизитов прячется, а часть показывается. Можно, конечно, воспользоваться конструкцией **форма.Реквизит.Видимость(1)** или **форма.Реквизит.Видимость(0)**, но это долго и не продуктивно. Более удобным, является использование слоев в форме. Когда Вы находитесь в форме - выберите "Диалог" - "Слой". Откроется небольшое окно, где по умолчанию присутствует один слой - Основной. Но Вы можете добавить столько слоев сколько Вам надо. Подробно на этом останавливаться не буду. Подскажу лишь только, что в открывшемся окне - рисунок "глаз" означает, что слой сейчас видно в форме, рисунок "карандаш" означает, что все реквизиты, которые Вы сейчас будете добавлять - будут находиться в этом слое. "Карандаши" и "глаза" можно переставлять по разным слоям. Одновременно может быть видно несколько слоев (глаз напротив нескольких слоев). А "карандаш" всегда будет находиться только у одного слоя.

Рекомендации по работе со слоями и закладками: как правило на каждую закладку создается по одному слою + один делается 1 слой, который будет содержать общие реквизиты, которые видны на любой закладке (например, кнопки "Ок", "Закрыть").

Управление слоями происходит с помощью метода **ИспользоватьСлой**, применяемого к объекту **Форма**. В общем виде метод выглядит так: **Форма.ИспользоватьСлой(имяСлоя, параметр)**; **ИмяСлоя** - строка, название слоя, как оно задано в конфигураторе, можно работать одновременно с несколькими слоями - в этом случае они перечисляются через запятую. **Параметр** - числовое значение 0 - скрыть слой <ИмяСлоя> в форме; 1 - показать слой <ИмяСлоя> в форме; 2 - показать слой <ИмяСлоя> и скрыть все остальные. Вернемся с нашему примеру с обработкой реакции системы на переключение закладок. Пусть у нас три закладки и 4 слоя "Первый" (для первой закладки), "Второй", "Третий" и "Основной" (слой с кнопками - общий для всех закладок). Как изменится процедура обработки выбора закладки:

```
Процедура ПриВыбореЗакладки(номер, зн)
```

```
Если ЗН="А" тогда
```

```
сообщить ("Вы выбрали закладку А");
```

```
форма.ИспользоватьСлой("Основной,Первый",2); // показываем нужные слои, а остальные прячем
```

```
иначеЕсли ЗН="Б" тогда
```

```
сообщить ("Вы выбрали закладку Б");
```

```
форма.ИспользоватьСлой("Основной,Второй",2); // показываем нужные слои, а остальные прячем
```

```
иначеЕсли ЗН="В" тогда
```

```
сообщить ("Вы выбрали закладку В");
```

```
форма.ИспользоватьСлой("Основной,Третий",2); // показываем нужные слои, а остальные прячем
```

```
конецЕсли;
```

```
конецПроцедуры
```

Получается не очень красиво и длинно. Как сделать короче? Когда создаем закладки - в качестве значения надо указать строку с названием слоев, которые должны быть показаны в момент выбора закладки. Покажем на том же примере:

```
Процедура ПриОткрытии()
```

```
Форма.ИспользоватьЗакладки(1); //разрешили использовать закладки
```

```
форма.Закладки.ДобавитьЗначение("Основной,Первый","Первая"); //добавляем закладки и сразу
```

```
указываем слои в значении
```

```
форма.Закладки.ДобавитьЗначение("Основной,Второй","Вторая");
```

форма.Закладки.ДобавитьЗначение("Основной,Третий","Третья");
КонецПроцедуры

Процедура ПриВыбореЗакладки(номер, зн)

форма.ИспользоватьСлой(зн,2); //используем значение в качестве показываемых слоев.

КонецПроцедуры

Важно: при работе со слоями в процедуре **ПриОткрытии()** надо не забыть указать слои которые будут видны при открытии формы - иначе отрываемая форма покажет сразу все слои.

Задачи для самостоятельной работы:

Задача 1. В форме создайте 2 кнопки "Добавить" и "Удалить". Одна кнопка добавляет закладки по одной справа, другая, соответственно удаляет по одной справа. При добавлении закладки называются названиями месяцев, начиная с января. Как только "год" будет заполнен - закладки перестают добавляться.

Задача 2. Тот пример, который приводился на уроке доработайте так, чтобы при открытии сразу активной была "Вторая" закладка.

Задача 3. В форме добавьте 2 кнопки "Лево" и "Право". При щелчке по кнопке текущая закладка (которая выбрана в форме) должна передвинуться на одну позицию влево или вправо.

Задача 4. В форме добавьте 7 флажков, названных днями недели. Форма должна работать просто: если флажок установлен, появляется соответствующая закладка, если флажок снят - закладка пропадает. Условие: закладки называются как дни недели. При этом они должны быть по порядку: "вторник" не может быть слева от "понедельника" - каждая закладка добавляется на свое место, а не справа.

Задача 5. Возьмите любой документ, содержащий табличную часть. Доработайте его таким образом, чтобы табличная часть была на отдельной закладке (Табличная Часть), все остальные реквизиты - на закладке "Шапка". Кнопки "Закреть", "Сохранить", "Печать" - должны присутствовать на обеих закладках.

Только для подписчиков рассылки 1С:Школа мы предлагаем:

- Для тех, кто совсем слаб в конфигурировании и у кого много вопросов по выходящей рассылке, мы предлагаем сервисную услугу. За небольшую абонентскую плату (5\$ в месяц) Вы получите возможность консультаций с автором рассылки по всем вопросам, которые у Вас возникают. Вопросы могут быть разные: по теории, по решению задач, по применению изученного материала. Условие только одно: задаваемые вопросы должны относиться к выпуску из того периода за который внесена абонентская плата. Начать консультации можно по Вашему выбору с любого выпуска, после того как поступила оплата. Длительность консультаций - ровно месяц: начали 23 марта - закончили 22 апреля.
- Мы предоставляем платные консультации по любым вопросам, касающимся конфигурирования компоненты бухгалтерский учет. Вопросы могут быть самые разнообразные постановка задачи, подсказки, советы, поиск оптимальной схемы решения поставленной задачи, поиск "неразрешимых" ошибок в Ваших модулях (тех, что не видно самому автору), написание "каркаса" модулей с объяснением их функционирования. Обращаем Ваше внимание на то, что мы не занимаемся консультациями по постановке учета бухгалтерского учета, по ведению бухгалтерского и налогового учета. Не делаем это в силу того, что по электронной почте тяжело решать такие вопросы. Консультации проходят в рамках месячного абонентского обслуживания по рабочим дням. Время

ответа на Ваши запросы - 1 сутки. Консультации проводятся только по рабочим дням. Очевидно, что вопрос - вопросу рознь, именно по этому мы рассчитываем на то, что средний размер ответа будет составлять не более 1 страницы текста (или модуля). Абонентская плата составляет 25\$ в месяц. Положитесь на наш опыт и мы поможем Вам осилить автоматизацию Вашей бухгалтерии.

- **Конфигурирование.** Если у вас есть документ, отчет с которым Вы не можете справиться самостоятельно, то мы можем Вам в этом помочь. От Вас требуется детальное описание того, что Вы хотите получить и конфигурация на которой это будет выполняться. После согласования сроков, оплаты и других условий мы выполним эту работу быстро и качественно.

Задание 1. Просмотрите видео ролик на тему «Создание формы в 1С и работа с обработчиками» <https://www.youtube.com/watch?v=yCx39EUkzOM>

Задание 2. Написать конспект

Задание 3. Выполнить практическую работу стр.58-63 (яндекс диск)

Лекция 33: Системные процедуры и функции. Предопределенные процедуры.

Программный модуль — это своеобразный «контейнер» для размещения текстов процедур и функций на встроенном языке 1С. Эти процедуры и функции вызываются системой в определенные моменты работы системы. Программные модули не имеют формальных границ своего описания типа: "Начало модуля" — "Конец модуля".

Программные модули размещаются в тех местах конфигурации, в которых может требоваться описание специфических алгоритмов функционирования. Эти алгоритмы оформляются в виде процедур или функций, которые будут вызваны самой системой в заранее предусмотренных ситуациях (например, при нажатии кнопки в диалоговом окне и т.д.). Каждый отдельный программный модуль воспринимается системой как единое целое.

Программные модули:

- **модуль управляемого приложения.** В данном модуле описываются процедуры (обработчики событий), которые инициализируются при старте и окончании работы системы. Например, при начале работы приложения можно обновить какие-либо данные конфигурации, вывести сообщения пользователю, а при завершении работы - поинтересоваться, стоит ли выходить из программы или, например, записать в базу данных время завершения работы пользователя.
- **модуль внешнего соединения.** Модуль внешнего соединения срабатывает, когда запуск приложения происходит в режиме com-соединения. В этом режиме происходит программная работа с информационной базой без запуска окна приложения.
- **модуль сеанса.** Это узкоспециализированный модуль, предназначенный исключительно для инициализации параметров сеанса.
- **общие модули.** Общие модули предназначены для описания некоторых общих алгоритмов (процедур и функций), которые будут вызываться из других модулей конфигурации.
- **модуль объекта конфигурации.** Данный модуль имеется у большинства объектов конфигурации и предназначен для обработки событий, непосредственно связанных с объектом. Например, события записи или удаления объектов, проверка заполнения реквизитов объекта, проведение документа и т.д.
- **модуль формы.** Модуль формы предназначен для обработки действий пользователя с данной формой (обработка события нажатия кнопки, изменения реквизита формы и т.д.).

Системные предопределенные процедуры предназначены для того, чтобы дать возможность программно обрабатывать системные события, возникающие при интерактивных действиях пользователя,

таких как ввод нового, изменение, удаление. Исключение составляют только predefined процедуры Модуля документа: `ОбработкаПроведения`, `ОбработкаУдаленияПроведения`, `АрхивироватьДокумент`, которые вызываются как при интерактивном, так и при программном возникновении события.

ВНИМАНИЕ! Системные predefined процедуры не являются встроенными процедурами языка. Для них зарезервированы только название и синтаксис. Тело процедур должно быть написано самим разработчиком конфигурации в соответствующих программных модулях.

Вызов системных predefined процедур (если они использованы в конфигурации) производится неявно самой системой 1С:Предприятие перед обработкой какого-либо интерактивного действия пользователя, например, при вводе нового документа. В этот момент система передает в predefined процедуру фактические значения параметров. В теле процедур переданные параметры могут использоваться для обработки события, например, выдачи предупреждающих сообщений, установки статуса возврата и т. п.

С помощью системной функции `СтатусВозврата` в теле predefined процедуры можно устанавливать значение статуса возврата процедуры. Статус возврата используется системой, чтобы определить — выполнить или нет действие того системного события, которое вызвало данную predefined процедуру.

Предопределенные процедуры Глобального модуля

Описанные в данном разделе системные predefined процедуры должны располагаться только в глобальном программном модуле. Данные процедуры, например, могут использоваться для расширения возможности программного управления правами доступа пользователя к системе.

ПриНачалеРаботыСистемы

Предопределенная процедура при начале работы задачи.

Синтаксис:

`ПриНачалеРаботыСистемы()`

Англоязычный синоним:

`OnStartSystem`

Описание:

Вызов predefined процедуры `ПриНачалеРаботыСистемы` производится системой 1С:Предприятие неявно при начале работы с программой. Если в данной predefined процедуре установить статус возврата — 0 (например, если время доступа данного пользователя к системе истекло), то пользователь не сможет войти в программу.

Данная predefined процедура может располагаться только в глобальном программном модуле.

Пример:

```
Процедура ПриНачалеРаботыСистемы()
Курс = Константа.ОсновнаяВалюта.Текущ_Курс.Получить(ТекущаяДата());
Предупреждение("Добро пожаловать! Сегодня " + " " + ТекущаяДата() +
РазделительСтрок + "Сейчас установлен курс доллара=" + Курс), 7);
КонецПроцедуры
```

ПриЗавершенииРаботыСистемы

Предопределенная процедура при завершении работы задачи.

Синтаксис:

`ПриЗавершенииРаботыСистемы()`

Англоязычный синоним:

`OnFinishSystem`

Описание:

Вызов predefined процедуры `ПриЗавершенииРаботыСистемы` производится системой 1С:Предприятие неявно при завершении работы пользователя с системой. Если в данной predefined процедуре установить статус возврата — 0 (например, если пользователь не выполнил какие-либо штатные операции), то работа системы не будет завершена, т. е. пользователь не сможет выйти из программы.

Данная predefined процедура может располагаться только в глобальном программном модуле.

Пример:

```
Процедура ПриЗавершенииРаботыСистемы()  
Предупреждение("До свидания!", 2);  
КонецПроцедуры
```

ПриУдаленииДокумента

Предопределенная процедура при удалении документа и при отмене пометки на удаление.

Синтаксис:

```
ПриУдаленииДокумента(<УдалДокум>, <Режим>)
```

Англоязычный синоним:

OnDeleteDoc

Описание:

Вызов предопределенной процедуры ПриУдаленииДокумента производится системой 1С:Предприятие неявно при интерактивном удалении документа и в случае отмены пометки на удаление. В этом случае, условия вызова можно узнать методом ПометкаУдаления документа. Если в данной предопределенной процедуре установить статус возврата — 0 (например, если не истек срок хранения документа), то удаление документа не будет выполнено.

Данная предопределенная процедура может располагаться только в глобальном программном модуле.

Пример:

```
Процедура ПриУдаленииДокумента(Док, Реж)  
Если НазваниеНабораПрав() = "Продавец" Тогда  
Если (Док.Вид() = "РасходнаяНакл") И (Реж = 1) Тогда  
Предупреждение("У вас нет права удалять накладную!", 2);  
СтатусВозврата(0);  
КонецЕсли;  
КонецЕсли;  
КонецПроцедуры
```

ПриУдаленииЭлемента

Предопределенная процедура при удалении элемента справочника и при отмене пометки на удаление.

Синтаксис:

```
ПриУдаленииЭлемента(<УдалЭлем>, <Режим>)
```

Англоязычный синоним:

OnDeleteItem

Описание:

Вызов предопределенной процедуры ПриУдаленииЭлемента производится системой 1С:Предприятие неявно при интерактивном удалении элемента справочника и в случае отмены пометки на удаление. В этом случае, условия вызова можно узнать методом ПометкаУдаления элемента справочника. Если в данной предопределенной процедуре установить статус возврата — 0 (например, если нарушается ссылочная целостность системы), то удаление элемента справочника не будет выполнено.

Данная предопределенная процедура может располагаться только в глобальном программном модуле.

Пример:

Процедура ПриУдаленииЭлемента(Элем, Реж)
Если НазваниеНабораПрав() = "Продавец" Тогда
Если (Элем.Вид() = "Товар") И (Реж = 1) Тогда
Предупреждение("У вас нет права удалять Товар!", 2);
СтатусВозврата(0);
КонецЕсли;
КонецЕсли;
КонецПроцедуры

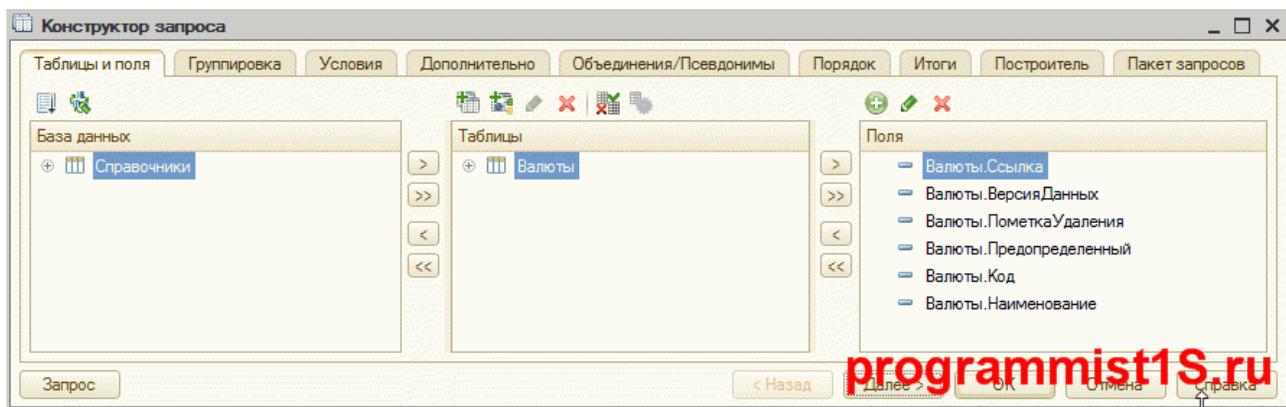
Задание 1. Просмотр видео «Глобальные переменные и модуль формы в 1С 8.3»
<https://www.youtube.com/watch?v=KFR5UriKR38>

Задание 2. Написать конспект

Задание 3. Выполнить практическую работу стр.63-68 (яндекс диск)

Лекция 34: Конструкторы

Конструктор запросов — инструмент, встроенный в систему 1С предприятие 8.3 и 8.2, существенно облегчающий разработку запросов 1С к базе данных.



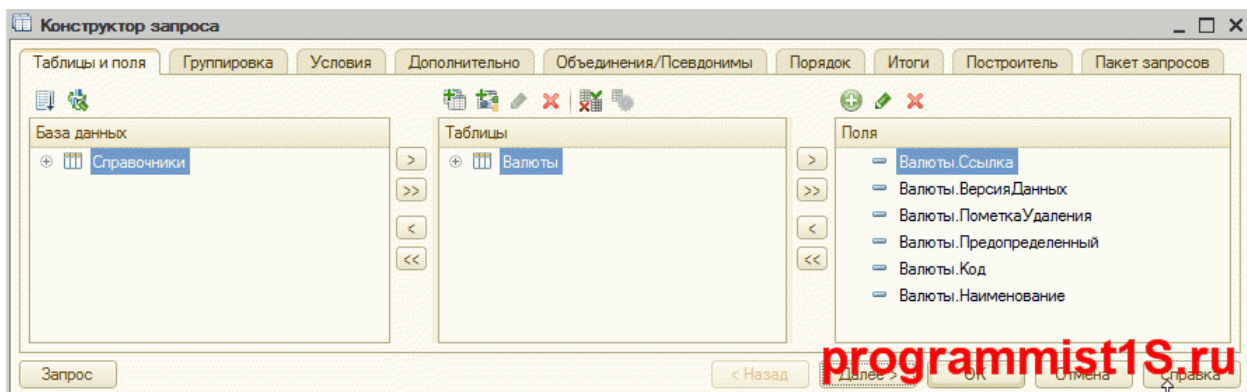
Конструктор запроса имеет достаточно простой, интуитивно понятный интерфейс. Тем не менее рассмотрим применение конструктора запроса подробнее.

Запуск конструктора текста запроса производится контекстным меню (правой кнопкой мыши) в нужном месте программного кода.

Описание конструктора запроса 1С

Рассмотрим каждую вкладку конструктора подробнее. Исключение — вкладка Построитель, это тема для отдельного разговора.

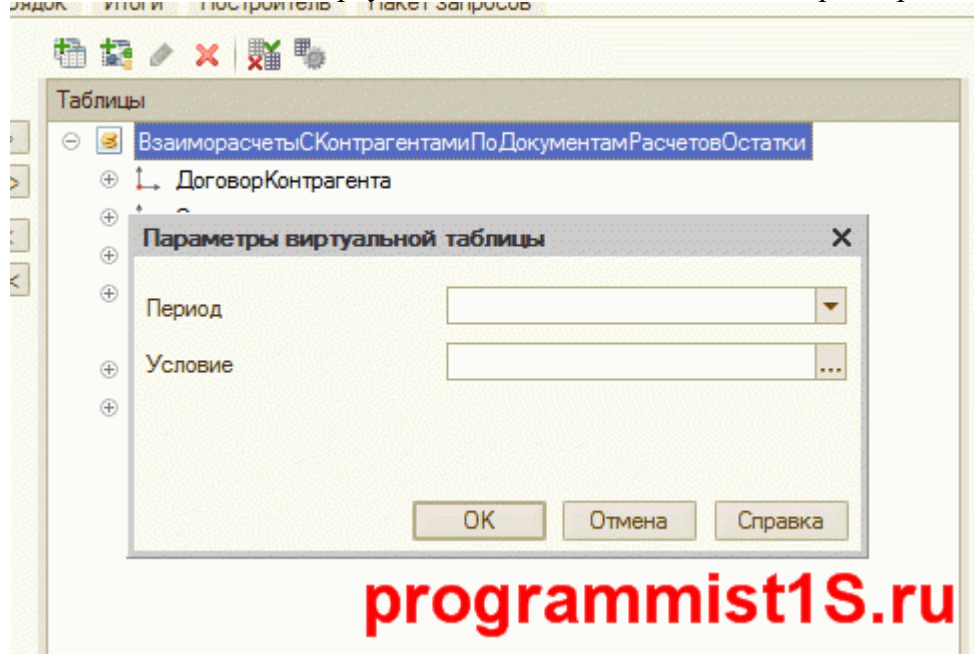
Вкладка Таблицы и поля



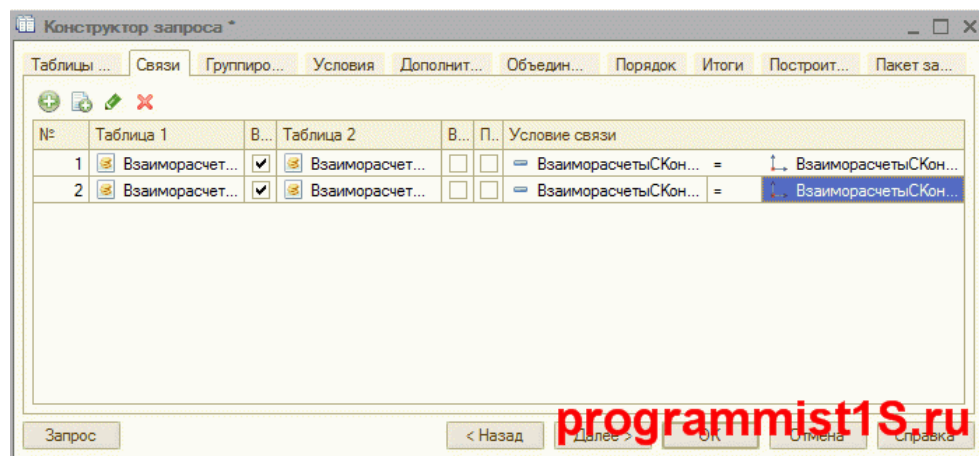
На этой вкладке указывается источник данных и поля, которые необходимо выводить в отчет. По сути, тут описываются конструкции ВЫБРАТЬ ... ИЗ.

В качестве источника может служить физическая таблица базы данных, виртуальная таблица регистров, временные таблицы, вложенные запросы и т.д.

В контекстном меню виртуальных таблиц можно задать параметры этих таблиц:

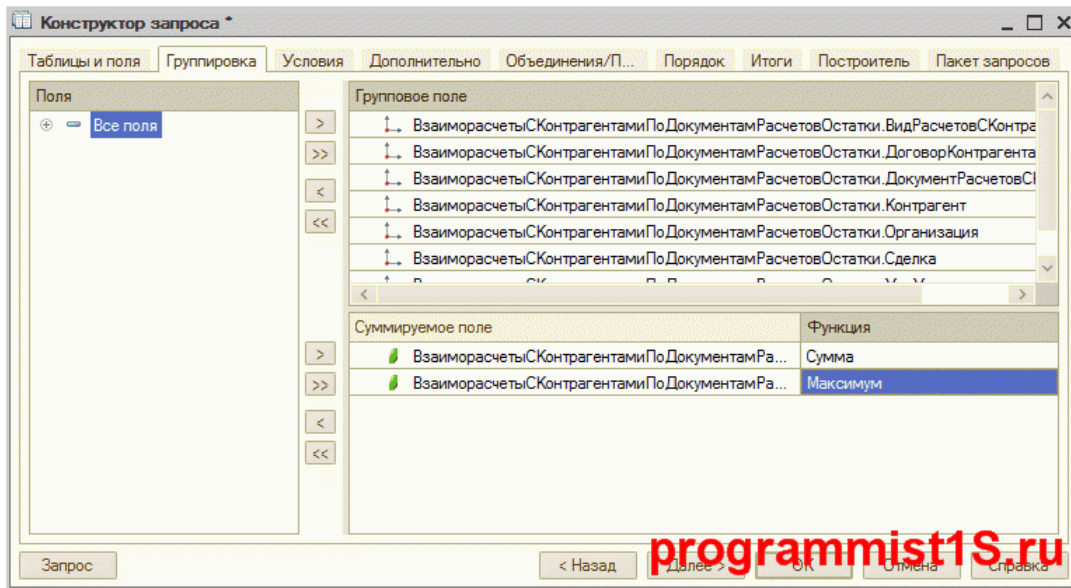


Вкладка Связи



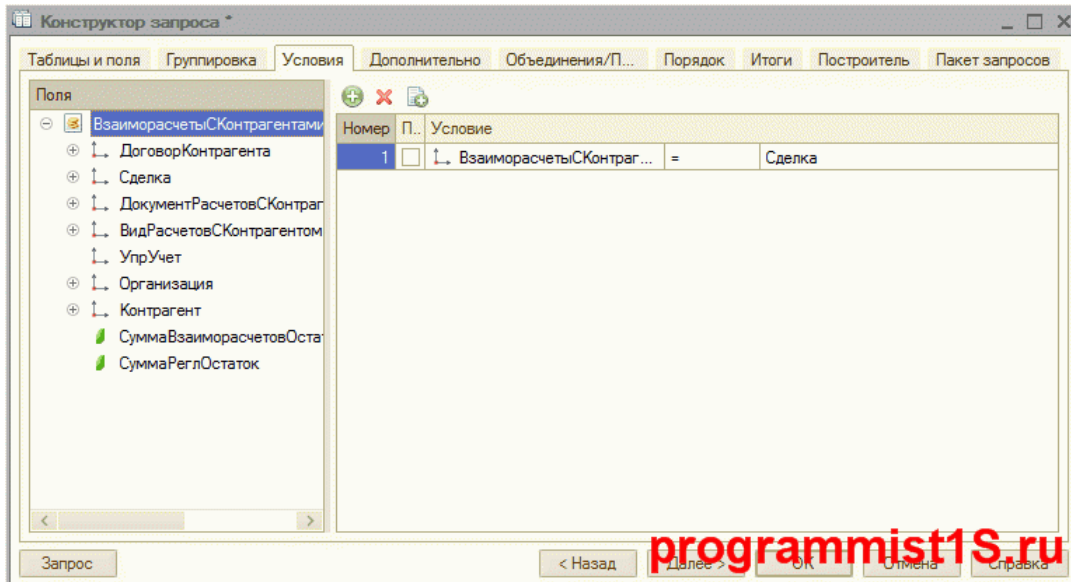
Вкладка служит для описания соединений нескольких таблиц, создает конструкции со словом СОЕДИНЕНИЕ.

Вкладка Группировка



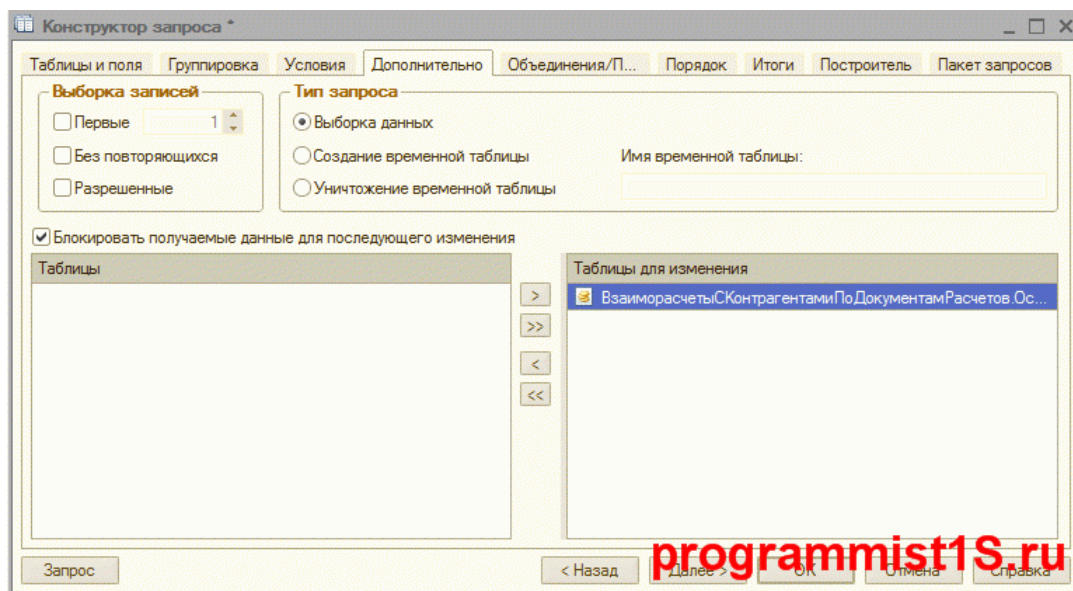
На данной вкладке система позволяет группировать и суммировать нужные поля результата таблицы. Описывается использование конструкций СГРУППИРОВАТЬ ПО, СУММА, МИНИМУМ, СРЕДНЕЕ, МАКСИМУМ, КОЛИЧЕСТВО, КОЛИЧЕСТВО РАЗЛИЧНЫХ.

Вкладка Условия



Отвечает за всё, что идёт в тексте запроса после конструкции ГДЕ, т.е. за все условия, накладываемые на получаемые данные.

Вкладка Дополнительно



Вкладка **Дополнительно** изобилует всевозможными параметрами, которые являются очень важными. Рассмотрим каждое из свойств.

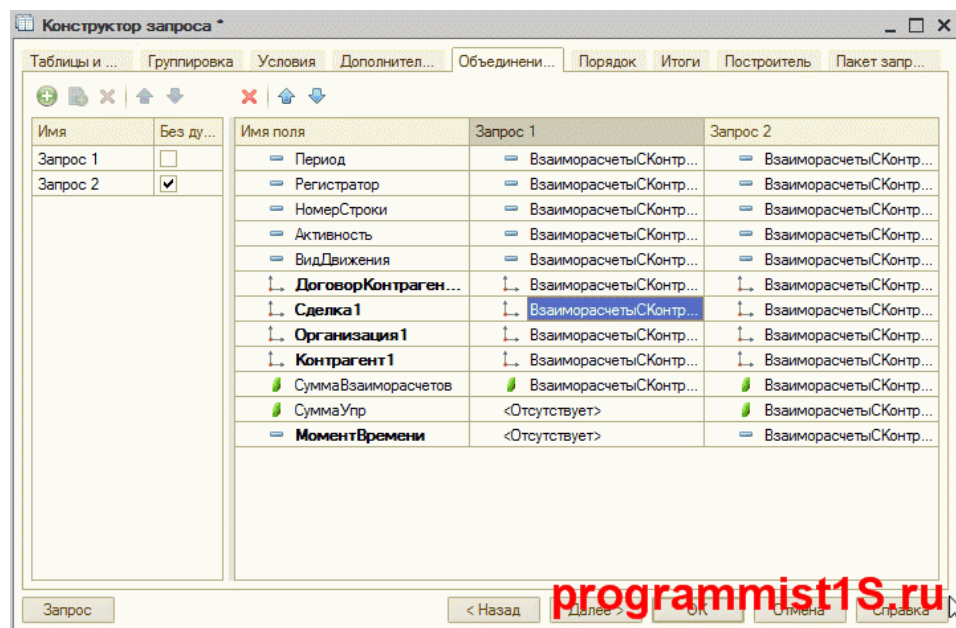
Группировка **Выборка записей**:

- Первые N — параметр, возвращающий в запрос только N записей (оператор ПЕРВЫЕ)
- Без повторяющихся — обеспечивает уникальность полученных записей (оператор РАЗЛИЧНЫЕ)
- Разрешенные — позволяет выбирать только те записи, которые позволяет выбрать система с учетом настройки RLS (конструкция РАЗРЕШЕННЫЕ)

Группировка **Тип запроса** определяет, каким будет тип запроса: выборка данных, создание временных таблицы или же уничтожение временной таблицы.

Ниже существует флаг **Блокировать полученные данные для последующего изменения**. Он позволяет включить возможность установки блокировки данных, которая обеспечивает сохранность данных от момента их чтения до изменения (актуально только для Автоматического режима блокировок, конструкция ДЛЯ ИЗМЕНЕНИЯ).

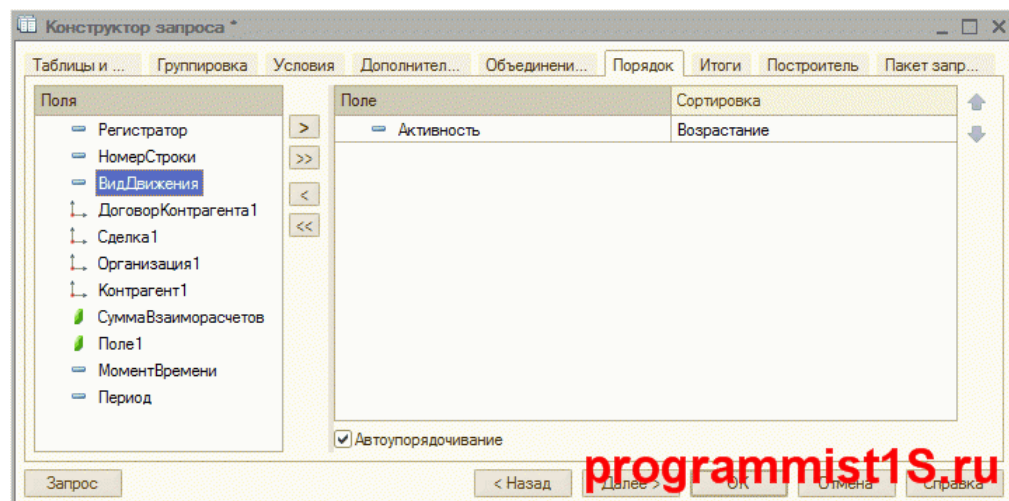
Вкладка Объединения/Псевдонимы



На этой вкладке конструктора запросов устанавливается возможность объединения разных таблиц и псевдонимы (конструкция КАК). В левой части указываются таблицы, если установить флаги напротив таблицы, будут использоваться конструкции ОБЪЕДИНИТЬ, иначе — ОБЪЕДИНИТЬ ВСЕ (отличия двух

способов [тут](#)). В правой части указываются соответствия полей в разных таблицах, если соответствие не указано, запрос будет возвращать NULL.

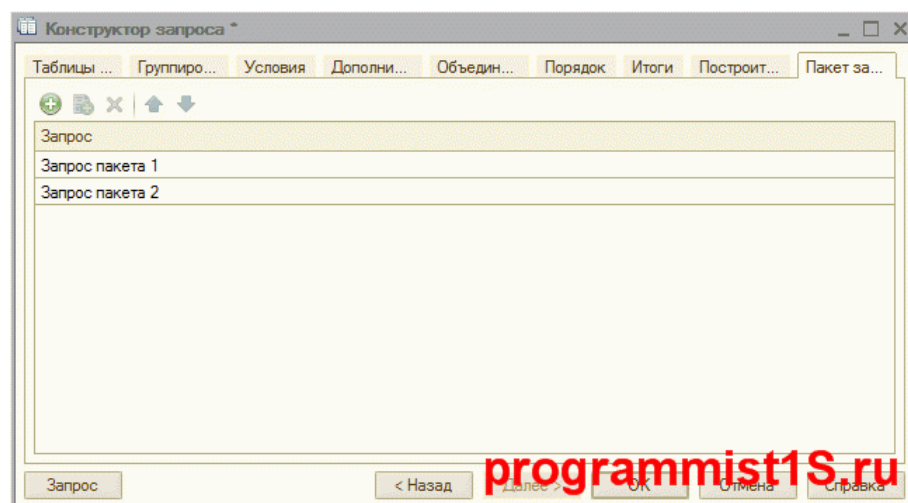
Вкладка Порядок



Здесь указывается порядок сортировки значений (УПОРЯДОЧИТЬ ПО) — по убыванию (УБЫВ) или возрастанию (ВОЗР).

Также есть интересный флаг — **Автоупорядочивание** (в запросе — АВТОУПОРЯДОЧИВАНИЕ). По умолчанию, система 1С 8.3 выводит данные в «хаотичном» порядке. Если установить этот флаг, система будет сортировать данные по внутренним данным.

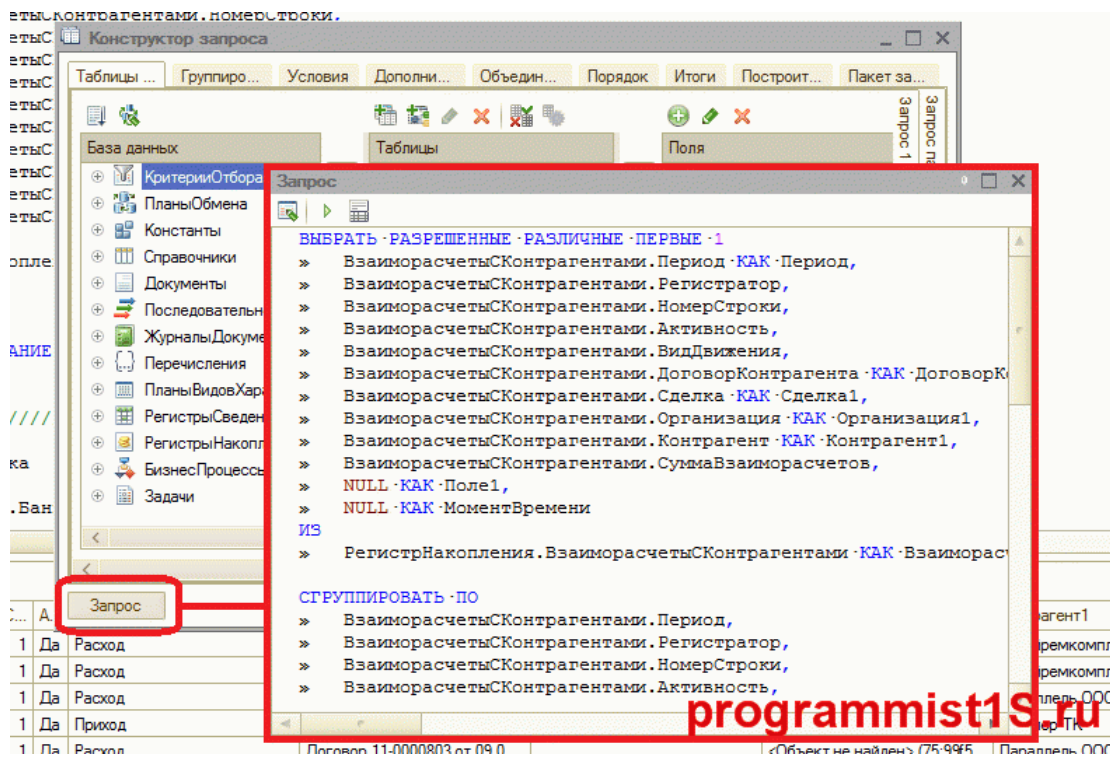
Вкладка Пакет запросов



На вкладке конструктора запросов можно создавать новые пакеты запросов, а также использовать её в роли навигации. В тексте запроса пакеты разделяются символом «;»(точка с запятой).

Кнопка «Запрос» в конструкторе запросов

В левом нижнем углу конструктора запроса есть кнопка **Запрос**, с помощью которого можно в любой момент просмотреть текст запроса:



Задание 1. Просмотр видео «Конструктор запросов»

https://www.youtube.com/watch?list=PL6Nx1KDcurkBdxssD1k56SDnWuTuX2bBr&time_continue=3&v=Awwv-cJ7sdg&feature=emb_logo

Задание 2. Написать конспект

Задание 3. Выполнить практическую работу стр.68-72 (яндекс диск)

Лекция 35: Компонента Расчет Зарплаты.

В конфигурации «Зарплата и управление персоналом 8 для Казахстана», ред. 3.0 автоматизирована возможность расчета зарплаты работникам, согласно назначенному окладу, тарифной ставке, штатному расписанию и т.д. Начисление заработной платы, а также налогов, взносов и отчислений выполняется документом Начисление зарплаты и взносов, который доступен в разделе Зарплата.

Месяц начисления	Тип	Дата	Номер	Организация	Начислено	Удержано	Ответственный	Комментарий
Январь 2016	Начисление зарплаты и взносов	31.01.2016	0000-000001	Крон-Ц	1 975 239,45	14 687,59	Орлова Е.Н. (Нач. стд. расчетов по о...)	
Февраль 2016	Начисление зарплаты и взносов	29.02.2016	0000-000004	Крон-Ц	6 352 875,75	87 055,33	Орлова Е.Н. (Нач. стд. расчетов по о...)	
Март 2016	Начисление зарплаты и взносов	31.03.2016	0000-000007	Крон-Ц	6 501 673,01	87 827,19	Орлова Е.Н. (Нач. стд. расчетов по о...)	
Апрель 2016	Начисление зарплаты и взносов	30.04.2016	0000-000010	Крон-Ц	6 643 287,75	87 301,23	Орлова Е.Н. (Нач. стд. расчетов по о...)	
Май 2016	Начисление зарплаты и взносов	31.05.2016	0000-000013	Крон-Ц	6 575 890,76	87 403,69	Орлова Е.Н. (Нач. стд. расчетов по о...)	
Июнь 2017	Начисление зарплаты и взносов	31.07.2017	0000-000001	Крон-Ц	6 625 177,54	531 082,37	Саввинская Э.Ю. (Системный програм...)	
Август 2017	Начисление зарплаты и взносов	31.08.2017	0000-000002	Крон-Ц	6 652 797,96	84 214,86	Саввинская Э.Ю. (Системный програм...)	
Октябрь 2017	Начисление зарплаты и взносов	31.10.2017	0000-000003	Крон-Ц	6 564 895,95	114 991,43	Саввинская Э.Ю. (Системный програм...)	

При создании документа указывается организация, дата документа и месяц начисления заработной платы. Документ может быть заполнен «вручную» путем подбора из справочника Сотрудники, а также автозаполнением списком сотрудников организации.

Начисление зарплаты и взносов (создание) *

Провести и закрыть | Записать | Провести

Еще ▾

Месяц: Октябрь 2017 | Дата: 23.10.2017 | Номер:

Организация: Крон-Ц

Заполнить ▾ | Подбор | Очистить

Начислено: 6 589 896,95 | Доначислено: 0,00 | Удержано: 114 991,43

Начисления | Доначисления, перерасчеты | Договоры | Удержания | Займы | Налоги, взносы, отчисления

Добавить | Найти... | Отменить поиск | Отмена исправлений | Расчетный листок | Еще ▾

N	Сотрудник	Подразделение	Начисление	Результат	Период	Основание
1	Булатов Игорь ...	Руководство	Оплата по окладу	350 000,00	01.10.2... 31.10.2017	
2	Минчев Аристо...	Руководство	Оплата по окладу	300 000,00	01.10.2... 31.10.2017	
3	Соподовникова ...	Управление бухга...	Оплата по окладу	265 000,00	01.10.2... 31.10.2017	
4	Мейерсон Соф...	Сметно-штатный ...	Оплата по окладу	255 000,00	01.10.2... 31.10.2017	
5	Орлова Елена ...	Отдел расчетов п...	Оплата по окладу	255 000,00	01.10.2... 31.10.2017	
6	Милова Светла...	Отдел расчетов п...	Оплата по окладу	245 000,00	01.10.2... 31.10.2017	
7	Яцкив Андрей ...	Юридический отд...	Оплата по окладу	265 000,00	01.10.2... 31.10.2017	
8	Громова Надеж...	Отдел по работе ...	Оплата по окладу	265 000,00	01.10.2... 31.10.2017	
9	Работкина Жан...	Отдел по работе ...	Оплата по окладу	230 000,00	01.10.2... 31.10.2017	
10	Габровская Св...	Сектор развития ...	Оплата по окладу	181 818,18	01.10.2... 31.10.2017	
11	Бальцер Герма...	Управление марк...	Оплата по окладу	265 000,00	01.10.2... 31.10.2017	

Ручное заполнение удобнее использовать при расчете зарплаты конкретному сотруднику (одному, двум). Если же необходимо начислить зарплату какому-либо подразделению или всей организации, то удобнее использовать автоматическое заполнение. При выборе команды Заполнить, программой заполняются сведения о должности, подразделении, виде и размере начисления сотрудников, согласно кадровых назначений.

Начисление зарплаты и взносов 0000-000003 от 31.10.2017

Провести и закрыть | Записать | Провести

Еще ▾

Месяц: Октябрь 2017 | Дата: 31.10.2017 | Номер: 0000-000003

Организация: Крон-Ц

Заполнить ▾ | Подбор | Очистить

Начислено: 0,00 | Удержано: 114 991,43

Начисления | Договоры | Удержания | Займы | Налоги, взносы, отчисления

Добавить | Найти... | Отменить поиск | Отмена исправлений | Расчетный листок

N	Сотрудник	Подразделение	Начисление	Результат	Период	Основание
1	Булатов Игорь Виленович	Руководство	Оплата по окладу	350 000,00	01.10.2017 31.10.2017	
2	Минчев Аристокель Борисович	Руководство	Оплата по окладу	300 000,00	01.10.2017 31.10.2017	
3	Соподовникова Мария Пахомовна	Управление бухгалтерского учета и отчетности	Оплата по окладу	265 000,00	01.10.2017 31.10.2017	
4	Мейерсон Софья Карловна	Сметно-штатный отдел	Оплата по окладу	255 000,00	01.10.2017 31.10.2017	
5	Орлова Елена Николаевна	Отдел расчетов по оплате труда	Оплата по окладу	255 000,00	01.10.2017 31.10.2017	
6	Милова Светлана Афанасьевна	Отдел расчетов по оплате труда	Оплата по окладу	245 000,00	01.10.2017 31.10.2017	
7	Яцкив Андрей Эдуардович	Юридический отдел	Оплата по окладу	265 000,00	01.10.2017 31.10.2017	
8	Громова Надежда Петровна	Отдел по работе с персоналом	Оплата по окладу	265 000,00	01.10.2017 31.10.2017	
9	Работкина Жанна Дмитриевна	Отдел по работе с персоналом	Оплата по окладу	230 000,00	01.10.2017 31.10.2017	
10	Габровская Светлана Марковна	Сектор развития персонала	Оплата по окладу	181 818,18	01.10.2017 31.10.2017	
11	Бальцер Герман Эдуардович	Управление маркетинга и обслуживания клиентов	Оплата по окладу	265 000,00	01.10.2017 31.10.2017	
12	Волков Марат Савельевич	Управление установки и эксплуатации оборудования	Оплата по окладу	265 000,00	01.10.2017 31.10.2017	
13	Нижамов Егор Кириллович	Отдел установки и эксплуатации оборудования	Оплата по окладу	250 000,00	01.10.2017 31.10.2017	
14	Мартынок Олег Егорович	Отдел установки и эксплуатации оборудования	Оплата по окладу	240 000,00	01.10.2017 31.10.2017	
15	Сорока Петр Максимович	Отдел установки и эксплуатации оборудования	Оплата по окладу	235 000,00	01.10.2017 31.10.2017	
16	Сорока Петр Максимович	Отдел установки и эксплуатации оборудования	За наставничество	23 500,00	01.10.2017 31.10.2017	

Для того чтобы сведения о плановом и фактически отработанном времени были заполнены корректно, необходимо предварительно проверить заполнение производственного календаря с учетом всех переносов праздничных дней, графиков работы и табеля учета рабочего времени.

← → ☆ **Начисление зарплаты и взносов 0000-000003 от 31.10.2017**

Пронести и закрыть Записать Провести

Месц: Октябрь 2017 Дата: 31.10.2017 Номер: 0000-000003

Организация: Крон-Ц

Заполнить Подбор Очистить

Начислено: 6 564 896,95 ? Доначислено: 0,00 Удержано: 114 991,43 ?

Начисления Доначисления, перерасчеты Договоры Удержания Займы Налоги, взносы, отчисления

Добавить Найти Отменить поиск Отмена исправлений **Расчетный листок** Еще

N	Сотру...	Подраз...	Начис...	Результат	Отработано (оплачено)	Норма времени	Показатели	Период
8	Громова Наде...	Отдел по работе...	Оплата по...	265 000,00	22,00 дн.	22,00 дн. 176,00 ч.	Оклад 265 000	01.10.2017 - 31.10.2017
9	Работкина Жанна...	Отдел по работе...	Оплата по...	230 000,00	22,00 дн.	22,00 дн. 176,00 ч.	Оклад 230 000	01.10.2017 - 31.10.2017
10	Габр. Светл.	Сектор развития...	Оплата по...	181 818,18	16,00 дн.	22,00 дн. 128,00 ч.	Оклад 250 000	01.10.2017 - 31.10.2017
11	Бальцер Герман...	Управл... маркетинг...	Оплата по...	265 000,00	22,00 дн.	22,00 дн. 176,00 ч.	Оклад 265 000	01.10.2017 - 31.10.2017

По кнопке **Расчетный листок**, пользователь может получить сводную информацию о видах начисления (оклад, отпускные, больничные и т.д.), а также информацию о начисленных налогах, взносах, отчислениях и удержаниях.

← → ☆ **Начисление зарплаты и взносов 0000-000003 от 31.10.2017**

Начислено: 6 564 896,95 ? Доначислено: 0,00 Удержано: 114 991,43 ?

Начисления Доначисления, перерасчеты Договоры Удержания Займы Налоги, взносы, отчисления

Добавить Найти Отменить поиск Отмена исправлений **Расчетный листок**

N	Сотрудник	Подразделение	Начисление	Результат	Период
7	Яцкив Андрей Эдуардович	Юридический отдел	Оплата по окладу	265 000,00	01.10.2... 31.10.2017
8	Громова Надежда Петровна	Отдел по работе с персоналом	Оплата по окладу	265 000,00	01.10.2... 31.10.2017
9	Работкина Жанна Дмитриевна	Отдел по работе с персоналом	Оплата по окладу	230 000,00	01.10.2... 31.10.2017
10	Габровская Светлана Марковна	Сектор развития персонала	Оплата по окладу	181 818,18	01.10.2... 31.10.2017
11	Бальцер Герман Эдуардович	Управление маркетинга и обслужи...	Оплата по окладу	265 000,00	01.10.2... 31.10.2017
12	Волков Марат Савельевич	Управление установок эксплуата...	Оплата по окладу	265 000,00	01.10.2... 31.10.2017
13	Никаноров Егор Кириллович	Отдел установки и эксплуатации в...	Оплата по окладу	250 000,00	01.10.2... 31.10.2017

Комментарий:

Печать документов

Печать Копий:

Организация: Крон-Ц

РАСЧЕТНЫЙ ЛИСТОК ЗА ОКТЯБРЬ 2017
Габровская Светлана Марковна (00009)
 Организация: Крон-Ц Должность: Начальник сектора
 Подразделение: Сектор развития персонала Оклад (тариф): 250 000

Вид	Период	Рабочие Дни	Часы	Оплачено	Сумма	Вид	Период	Сумма		
Начислено:					215 853,18	Удержано:				
Оплата по окладу	окт. 2017	16	128	16,00 дн.	181 818,18	ИПН	окт. 2017	16 900,89		
Оплата больничных листов за счет работодателя	04.10-11.10	6	48	6,00 дн.	34 035,00	ОПБ	окт. 2017	21 585,32		
Выплачено:										
Долг предприятия на начало					409 891,80	Долг предприятия на конец				
						587 176,77				

На закладке **Доначисления, перерасчеты** реализована возможность выполнения следующих действий:
Доначисление – позволяет выполнить расчет начисления за предыдущий месяц, если в предыдущем периоде оно не было выполнено; **Перерасчет** - возможность автоматического перерасчета сумм сотрудникам, связанных с изменениями данных по зарплате «задним числом». Например, после выполнения начисления по зарплате за прошлый месяц, возникла необходимость перерасчета фактически отработанного времени (сотрудник предоставил больничный лист за прошлый период). Для этого выполняется перерасчет начисления зарплаты со сторнированием предыдущего начисления.

Начисление зарплаты и взносов 0000-000003 от 31.10.2017 *

Провести и закрыть | Записать | Провести

Месяц: Октябрь 2017 | Дата: 31.10.2017 | Номер: 0000-000003

Организация: КронЦ

Начислено: 6 589 896,95 | Доначислено: 6 270,00 | Удержано: 114 991,43

Начисления: **Доначисления, перерасчеты** | Договоры | Удержания | Займы | Налоги, взносы, отчисления

N	Сотрудник	Подразделение	Начисление	Результат	Период	Основание
1	Вьялов Александр Матвеевич (инт. совм.)	Хозяйственный отдел	Доплата за работу в ночные часы	6 270,00	01.10.2017 - 31.10.2017	

На закладке Договоры отражаются сведения о начислениях по договорам гражданско-правового порядка (ГПХ).

Начисление зарплаты и взносов 0000-000003 от 31.10.2017 *

Провести и закрыть | Записать | Провести

Месяц: Октябрь 2017 | Дата: 31.10.2017 | Номер: 0000-000003

Организация: КронЦ

Начислено: 6 584 896,95 | Доначислено: 0,00 | Удержано: 114 991,43

Начисления: **Договоры** | Удержания | Займы | Налоги, взносы, отчисления

N	Сотрудник	Подразделение	Договор	Результат	Период	Дата выплаты	Начисление
1	Минчев Арсентьев Борисович		Договор (работы, услуги) 0000-000001 от 16.10.2017	25 000,00	01.10.2017 - 31.10.2017	05.11.2017	Оплата по договорам подряда

На закладке Удержания отражаются расчеты по дополнительным удержаниям, назначенным в программе соответствующими документами (исполнительный лист, компенсации за порчу имущества и т.д.). Например, удержания по исполнительному листу, согласно зарегистрированному в системе документу Исполнительный лист.

Начисление зарплаты и взносов 0000-000003 от 31.10.2017 *

Провести и закрыть | Записать | Провести

Месяц: Октябрь 2017 | Дата: 31.10.2017 | Номер: 0000-000003

Организация: КронЦ

Начислено: 6 589 896,95 | Доначислено: 6 270,00 | Удержано: 114 991,43

Начисления: **Удержания** | Займы | Налоги, взносы, отчисления

N	Сотрудник	Удержание	Результат	Период	Основание
1	Войткевич Игорь Борисович	Удержание по исполнительному документу	52 875,00	01.10.2017 - 31.10.2017	Исполнительный лист 0000-000001 от 01.02.2013

В случае наличия у сотрудников договоров займа с организацией, на закладке Займы отражается расчет удержания части долга по таким договорам за соответствующий период.

Начисление зарплаты и взносов 0000-000003 от 31.10.2017 *

Провести и закрыть | Записать | Провести

Месяц: Октябрь 2017 | Дата: 31.10.2017 | Номер: 0000-000003

Организация: КронЦ

Начислено: 6 589 896,95 | Доначислено: 6 270,00 | Удержано: 114 991,43

Начисления: **Займы** | Налоги, взносы, отчисления

N	Сотрудник	Погашено осн. долга	Начислено %	Погашено %	Договор займа	Подразделение
1	Бальцер Герман Одуардович	50 000,00	12 116,43	12 116,43	Договор займа сотруднику 0000-000001 от 01.02.2016	Управление маркетинга и обслуживания клиентов

Помимо расчета зарплаты, дополнительных начислений и удержаний в документе Начисление зарплаты и взносов выполняется расчет налогов, взносов, отчислений. Начисления по ним отражаются на закладке Налоги, взносы, отчисления. Начисления по регламентированным налогам, взносам и отчислениям выполняются при условии настройки в программе видов начислений, ставок налогов, отчислений, порядке расчета и т.д. Рассмотрим некоторые особенности настройки программы для расчета налогов, взносов и отчислений:

ИПН - начисление выполняется в соответствии с указанными для сотрудников видами вычетов, льгот по ИПН, гражданстве сотрудника и сведениями о ставке налога в регистре сведений Сведения о ставках налогов, сборов, отчислений;

ОПВ – на расчет ОПВ влияют сведения о том, является ли сотрудник плательщиком пенсионных взносов, является ли пенсионером, а также сведения о ставке взносов в регистре сведений Сведения о ставках налогов, сборов, отчислений;

ОППВ – начисление обязательных пенсионных профессиональных взносов выполняется в случае установки признака Является агентом по уплате ОППВ в настройках сведений об организации. Для штатной единицы также необходимо указать правила вхождения данной должности в перечень должностей с вредными условиями труда;

СН, СО – начисление социального налога и социальных отчислений выполняется согласно сведениям о ставках в регистре сведений Сведения о ставках налогов, сборов, отчислений, а также о применяемом организацией режиме налогообложения;

Вносы и отчисления на ОСМС – начисления выполняются согласно данным о ставках взносов и отчислений, указанным в регистре сведений Сведения о ставках обязательного социального медицинского страхования. Помимо этого для сотрудников необходимо проверить установку признака Подлежит социальному медицинскому страхованию в справочнике Сотрудники.

После выполнения расчета зарплаты, взносов, удержаний в программе необходимо отразить все начисления в регламентированном учете, для этого создается документ Отражение зарплаты в бухучете (раздел Зарплата).

После того, как начисления зарплаты и налогов, взносов и отчислений отражены в бухучете, пользователь может сформировать документы на перечисление.

Таким образом, конфигурация «Зарплата и управление персоналом 8 для Казахстана» позволяет выполнять расчет заработной платы с учетом зарегистрированных отсутствий сотрудника, назначенных начислений и удержаний.

Лекция 36: Журналы расчетов. Правила перерасчета

Компонента Расчет в системе 1С:Предприятие предназначена для реализации сложных периодических расчетов. Обычно она используется для расчета зарплаты.

Журналы расчетов

Журнал расчетов предназначен для хранения данных сложных периодических расчетов, поддерживаемых компонентой «Расчет», и их предыстории. Основным свойством журнала расчетов является его принадлежность тому или иному справочнику системы. Такой справочник называется справочником объектов расчета. Фактически, справочник определяет вид объектов, по которым ведется расчет. Например, это может быть: журнал расчетов зарплаты, причем списком объектов расчета будет выступать справочник сотрудников; журнал расчетов дивидендов, тогда списком объектов расчета будет выступать справочник акционеров; журнал расчетов амортизации — списком объектов расчета будет выступать справочник основных средств. В процессе конфигурирования настраивается неограниченное число журналов расчета, каждый из которых будет решать ту или иную задачу предметной области.

Непременными атрибутами каждой записи журнала расчетов являются: объект, для которого данный расчет проведен (обычно сотрудник); вид расчета, по которому данный расчет проведен; дата начала и дата окончания действия данного расчета и результат расчета.

Для одного справочника могут быть созданы несколько журналов расчетов, каждый из которых будет содержать данные определенной предметной области. Например, в том случае, если предприятие — акционерное общество закрытого типа, справочник сотрудников может выступать списком объектов расчета для журнала расчетов заработной платы и для журнала расчетов дивидендов акционеров.

Виды расчета

Для описания алгоритмов, по которым выполняются те или иные вычисления, служит понятие виды расчетов. На этапе конфигурирования можно описать неограниченное количество видов расчетов. В отличие

от справочников, журналов расчета и документов, за понятием «вид расчета» не лежит реальных данных — это не более чем алгоритм вычисления, оперирующий данными журналов расчета, документов и справочников.

Алгоритм вида расчета описывается с помощью встроенного языка. Типичными примерами видов расчетов являются «начисление по окладу», «подходный налог».

Приоритет.

Часто очень важным бывает очередность, в которой выполняются те или иные расчеты. Например, в случае расчета заработной платы, как минимум, важно сначала рассчитать начисления, а затем удержания.

Для организации последовательности выполнения расчетов служит понятие их приоритета. Этот параметр может принимать значение от 0 до 999 и действует следующим образом: чем меньше приоритет расчета, тем раньше он выполняется. Это отражается в последовательности расположения записей журнала расчетов — записи с видами расчетов меньшего приоритета располагаются перед записями с видами расчетов большего приоритета.

Вытесняющие расчеты.

В некоторых случаях требуется автоматизировать предметные области, в которых расчет каких-либо величин может вестись взаимно исключаящими способами. Возвращаясь вновь к задаче расчета заработной платы, рассмотрим пример расчета следующих начислений: оплата рабочих дней по окладу, оплата дней болезни по больничному листу и оплата дней отпуска.

Очевидно, что в этом случае выполняется условие взаимоисключения перечисленных видов расчета — нельзя, например, одновременно и болеть и работать, т. е. получать и оклад, и деньги по больничному листу. Такие расчеты взаимно исключают друг друга во времени и система должна гарантировать, что ввод одного из них приведет к исключению другого. (Как выяснилось из реальной жизни можно заболеть во время отпуска ;-)

Группы расчетов

Для того чтобы при тех или иных расчетах можно было оперировать не только результатами расчетов по конкретным видам, но и результатами по нескольким видам расчетов, объединенных по определенному принципу, служит понятие групп расчетов. Примерами групп расчета могут служить: «начисления, облагаемые подходным налогом», «начисления, входящие в расчет средней зарплаты», «все удержания» и другие.

Правила перерасчета

«Правила перерасчета» являются вспомогательными объектами метаданных, предназначенными для автоматического отслеживания актуальности результатов проведенных расчетов при вводе новых записей журнала расчетов, удалении существующих или ручном исправлении результата расчета.

При создании правила перерасчета в Конфигураторе определяются виды расчетов, при редактировании которых правило перерасчета «срабатывает» (*едущие виды расчета*), и виды расчетов, которые должны быть перерассчитаны при срабатывании данного правила (*зависимые виды расчета*).

Например, для организации правильного перерасчета доплат к основным начислениям следует в качестве ведущих видов расчета указать оклад, тариф, сдельная оплата), а в качестве зависимых видов расчета следует указать доплаты к окладу, доплаты к тарифу и т.д.

Контекст работы с правилами перерасчета

Средства языка предоставляют возможность непосредственного доступа к атрибутам и методам правил перерасчета, объявленных в конфигураторе, в любом программном модуле (все объявленные в конфигураторе правила перерасчета *принадлежат глобальному контексту* конфигурации). Во всех текстах программных модулей для доступа к атрибутам или вызова методов правил перерасчета можно записать соответствующий метод или атрибут через точку после полного имени правила перерасчета.

Полное имя правила перерасчета записывается следующим образом:

ПравилоПерерасчета.<XXXXXX>,

где <XXXXXX> — идентификатор Правила Перерасчета, объявленный в конфигураторе.

Англоязычный синоним ключевого слова ПравилоПерерасчета — RecalculationRule.

Пример:

КолПериодов = ПравилоПерерасчета.Главное.КоличествоПериодов;

ВР = ВидРасчета.ПоОкладу;

ЗависитОтОклада = ПравилоПерерасчета.ПересчДоплат.ИмеетВедущий(ВР);

Атрибуты правил перерасчета

Тип

Тип правила перерасчета.

Синтаксис:

Тип

Англоязычный синоним:

Type

Описание:

Атрибут Тип позволяет прочитать/установить значение типа правила перерасчета. Данный атрибут может принимать следующие значения:

- 0 — зависимые виды расчетов должны быть перерасчитаны в том же периоде, что и вводимая запись журнала расчетов;
- 1 — зависимые виды расчетов должны быть перерасчитаны в нескольких периодах, следующих за периодом действия вводимой записи журнала расчетов (количество периодов задано атрибутом КоличествоПериодов).

Пример:

Тип = ПравилоПерерасчета.Главное.Тип;

КоличествоПериодов

Количество перерасчитываемых периодов для правила перерасчета.

Синтаксис:

КоличествоПериодов

Англоязычный синоним:

NumberOfPeriods

Описание:

Атрибут КоличествоПериодов позволяет прочитать/установить количество периодов перерасчета.

Пример:

КолПер = ПравилоПерерасчета.Главное.КоличествоПериодов;

Методы правил перерасчета

КоличествоВедущих

Возвращает количество «ведущих» видов расчета.

Синтаксис:

КоличествоВедущих()

Англоязычный синоним:

NumberOfLeadings

Возвращаемое значение:

Количество «ведущих» видов расчета.

Описание:

Метод КоличествоВедущих возвращает количество «ведущих» видов расчета (при вводе которых в журнал расчетов необходимо произвести перерасчеты согласно данного правила).

Пример:

// выведем все виды расчета от которых зависят доплаты

Для Сч = 1 По ПравилоПерерасчета.Доплаты.КоличествоВедущих() Цикл

Сообщить(ПравилоПерерасчета.Доплаты.ПолучитьВедущий(Сч));

КонецЦикла;

ИмеетВедущий

Установить принадлежность вида расчета к ведущим для правила перерасчета.

Синтаксис:

ИмеетВедущий(<ВидРасчета>)

Англоязычный синоним:

HasLeading

Параметры:

<ВидРасчета> Вид расчета.

Возвращаемое значение:

Число: 1, если данное правило перерасчета имеет заданный вид расчета в качестве ведущего (т. е. при вводе этого вида расчета в журнал расчетов сработает данное правило перерасчета); 0, если данное правило перерасчета не имеет заданный вид расчета в качестве ведущего.

Описание:

Метод `ИмеетВедущий` позволяет установить принадлежность вида расчета к ведущим для правила перерасчета.

Пример:

// зависит ли доплата от премии?

Зависит = `ПравилоПерерасчета.Доплаты.ИмеетВедущий(ВидРасчета.Премия);`

ПолучитьВедущий

Возвращает вид расчета из списка «ведущих» расчетов.

Синтаксис:

`ПолучитьВедущий(<НомерВР>)`

Англоязычный синоним:

`GetLeading`

Параметры:

<НомерВР> Числовое выражение — номер вида расчета в списке «ведущих» расчетов.

Возвращаемое значение:

Вид расчета с номером <НомерВР> из списка «ведущих» расчетов.

Описание:

Метод `ПолучитьВедущий` возвращает вид расчета с номером <НомерВР> из списка «ведущих» расчетов.

Пример:

`ВидВедРасч = ПравилоПерерасчета.Доплаты.ПолучитьВедущий(Инд);`

ДобавитьКакВедущий

Добавляет вид расчета в список «ведущих».

Синтаксис:

`ДобавитьКакВедущий(<ВидРасчета>)`

Англоязычный синоним:

`AddAsLeading`

Параметры:

<ВидРасчета> Вид расчета.

Описание:

Метод `ДобавитьКакВедущий` добавляет <ВидРасчета> в список «ведущих».

Пример:

`ПравилоПерерасчета.Доплаты.ДобавитьКакВедущий(ВидРасчета.Премия);`

УдалитьВсеВедущие

Удаляет все виды расчета из списка «ведущих».

Синтаксис:

`УдалитьВсеВедущие()`

Англоязычный синоним:

`DeleteAllLeadings`

Описание:

Метод `УдалитьВсеВедущие` удаляет все виды расчета из списка «ведущих».

Пример:

`ПравилоПерерасчета.Доплаты.УдалитьВсеВедущие();`

КоличествоПодчиненных

Возвращает количество «подчиненных» видов расчета.

Синтаксис:

`КоличествоПодчиненных()`

Англоязычный синоним:

`NumberOfDependents`

Возвращаемое значение:

Количество «подчиненных» видов расчета.

Описание:

Метод КоличествоПодчиненных возвращает количество «подчиненных» видов расчета (для которых необходимо произвести перерасчеты согласно данного правила).

Пример:

Колич = ПравилоПерерасчета.Доплаты.КоличествоПодчиненных();

ИмеетПодчиненный

Установить принадлежность вида расчета к «подчиненным» для правила перерасчета.

Синтаксис:

ИмеетПодчиненный(<ВидРасчета>)

Англоязычный синоним:

HasDependent

Параметры:

<ВидРасчета> Вид расчета.

Возвращаемое значение:

Число: 1, если данное правило перерасчета имеет заданный вид расчета в качестве подчиненного (т. е. для этого вида расчета в журнале расчетов сработает данное правило перерасчета); 0, если данное правило перерасчета не имеет заданный вид расчета в качестве подчиненного.

Описание:

Метод ИмеетПодчиненный позволяет установить принадлежность вида расчета к подчиненным для правила перерасчета.

Пример:

Подч = ПравилоПерерасчета.Доплаты.ИмеетПодчиненный(ВидРасчета.Премия);

ПолучитьПодчиненный

Возвращает вид расчета из списка «подчиненных» расчетов.

Синтаксис:

ПолучитьПодчиненный(<НомерВР>)

Англоязычный синоним:

GetDependent

Параметры:

<НомерВР> Числовое выражение — номер вида расчета в списке «подчиненных» расчетов.

Возвращаемое значение:

Вид расчета с номером <НомерВР> из списка «подчиненных» расчетов.

Описание:

Метод ПолучитьПодчиненный возвращает вид расчета с номером <НомерВР> из списка «подчиненных» расчетов.

Пример:

ВидПодчРасч = ПравилоПерерасчета.Доплаты.ПолучитьПодчиненный(Инд);

ДобавитьКакПодчиненный

Добавляет вид расчета в список «подчиненных».

Синтаксис:

ДобавитьКакПодчиненный(<ВидРасчета>)

Англоязычный синоним:

AddAsDependent

Параметры:

<ВидРасчета> Вид расчета.

Описание:

Метод ДобавитьКакПодчиненный добавляет <ВидРасчета> в список «подчиненных».

Пример:

ПравилоПерерасчета.Доплаты.ДобавитьКакПодчиненный(ВидРасчета.Премия);

УдалитьВсеПодчиненные

Удаляет все виды расчета из списка «подчиненных».

Синтаксис:

УдалитьВсеПодчиненные()

Англоязычный синоним:

DeleteAllDependents

Описание:

Метод УдалитьВсеПодчиненные удаляет все виды расчета из списка «подчиненных».

Пример:

ПравилоПерерасчета.Доплаты.УдалитьВсеПодчиненные();

Применять

Устанавливает необходимость применять или не применять все правила перерасчетов при вводе новых записей в журнал расчетов.

Синтаксис:

Применять(<ФлагПрименения>)

Англоязычный синоним:

Use

Параметры:

<ФлагПрименения> Число: 1 — применять; 0 — не применять.

Возвращаемое значение:

Текущее значение флага применения (до исполнения метода).

Описание:

Метод Применять устанавливает необходимость применять или не применять все заданные в конфигурации правила перерасчетов при изменении состава или исправлении записей журнала расчетов.

Пример:

ПравилоПерерасчета.Применять(1);

Календари и Праздники

Для ведения рабочего графика предприятия или сторонних организаций, для учета рабочего времени сотрудников разных категорий в системе 1С:Предприятие служат календари. Примерами календарей могут служить: календарь рабочего графика предприятия (где отмечены выходные и рабочие дни); календарь работы банка, через который осуществляются взаиморасчеты; календарь работы служащих; календарь, в котором проставлена продолжительность рабочих дней для рабочих.

Календарь представляет собой объект, который устанавливает для каждой календарной даты некоторое числовое значение, которое может интерпретироваться при работе программы тем или иным образом. Например, в календаре, представляющем собой график работы организации или предприятия, каждому рабочему дню может соответствовать единица, а каждому выходному — ноль. В календаре учета рабочего времени сотрудников предприятия каждой дате календаря может соответствовать число — продолжительность рабочего дня в часах и т. д.

Праздники. Допустим, что у вас на предприятии несколько категорий работников, например 5 категорий, и для каждой категории заведен свой календарь. Представьте, что в стране объявлен всеобщий праздник, например, "День 1С". Используя данную функцию вы можете один раз завести этот праздник, и он автоматически отразится во всех календарях, у которых стоит флажок "Учитывать праздники".

Дидактический материал**Лабораторная работа № 1 Знакомство, создание информационной базы.**

1С:Предприятие является универсальной системой автоматизации экономической и организационной деятельности предприятия. Поскольку такая деятельность может быть довольно разнообразной, система 1С:Предприятие может «приспосабливаться» к особенностям конкретной области деятельности, в которой она применяется. Для обозначения такой способности используется термин конфигурируемость, то есть возможность настройки системы на особенности конкретного предприятия и класса решаемых задач.

Логически всю систему можно разделить на две большие части, которые тесно взаимодействуют друг с другом: конфигурацию и платформу, которая управляет работой конфигурации.

Существует одна платформа (1С:Предприятие 8) и множество конфигураций. Для функционирования какого-либо прикладного решения всегда необходима платформа и какая-либо конфигурация (рис.

1.1).

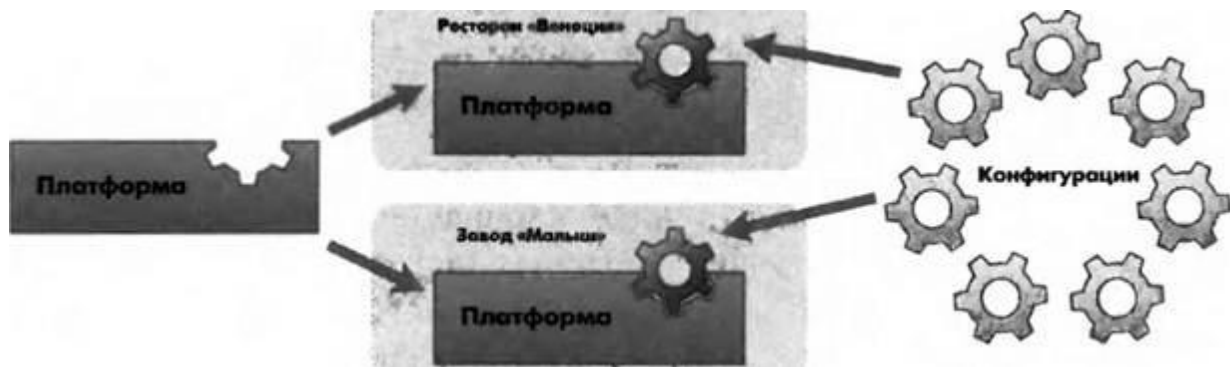


Рис. 1.1. Конфигураций много, а платформа одна

Сама по себе платформа не может выполнить никаких задач автоматизации, так как она создана для обеспечения работы какой-либо конфигурации. То же самое с конфигурацией: чтобы выполнить те задачи, для которых она создана, необходимо наличие платформы, управляющей ее работой.

Итак, поскольку задачи автоматизации могут быть самыми разными, фирма «1С» и ее партнеры выпускают прикладные решения, каждое из которых предназначено для автоматизации одной определенной области человеческой деятельности. В качестве примера существующих прикладных решений можно перечислить следующие типовые решения:

- 1С:Управление небольшой фирмой,
- 1С бухгалтерия,
- 1С:Предприятие. Управление торговлей,
- 1С: Зарплата и Управление Персоналом,
- 1С:Предприятие. Управление производственным предприятием,
- 1С:Консолидация.

Режимы работы системы:

1С: Предприятие имеет различные режимы работы: 1С: Предприятие и Конфигуратор.

Режим 1С Предприятие является основным и служит для работы пользователей системы. В этом режиме пользователи вносят данные, обрабатывают их и получают итоговые результаты.

Режим Конфигуратор используется разработчиками и администраторами информационных баз.

Именно этот режим и предоставляет инструменты, необходимые для модификации существующей или создания новой конфигурации.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Создание новой информационной базы

Запустите 1С Предприятие. В открывшемся диалоге вы увидите список информационных баз, с которыми вы работаете. Если этот список пуст, система сама предложит вам создать новую базу. Если

же в списке информационных баз содержится какая-либо база, например, у вас установлена демонстрационная конфигурация, то для создания новой базы нажмите кнопку **Добавить** (рис. 1.2).

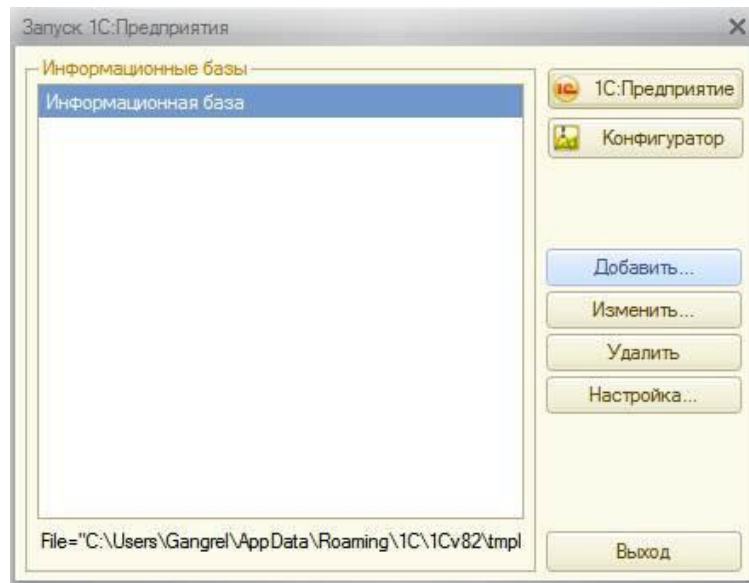


Рис. 1.2. Создание новой информационной базы

- В открывшемся диалоге выберите пункт **«Создание новой информационной базы»;**
- Нажмите кнопку **«Далее»** На следующем шаге выберите пункт **«Создание информационной базы без конфигурации»;**
- Нажмите кнопку **«Далее»**. На следующем шаге задайте наименование вашей информационной базы и выберите тип ее расположения **«На данном компьютере»;**
- Нажмите кнопку **«Далее»**. На следующем шаге укажите каталог для расположения вашей информационной базы. Язык по умолчанию установлен в значение **Русский;**
- Нажмите кнопку **«Далее»**. На следующем шаге нажмите кнопку **«Готово»;**
- В диалоге запуска 1С:Предприятия, в списке информационных баз, вы увидите созданную вами новую пустую базу.

Знакомство с конфигуратором:

Итак, запустим 1С:Предприятие в режиме Конфигуратор. Для этого нажмем кнопку **«Конфигуратор»**

в диалоге запуска системы.

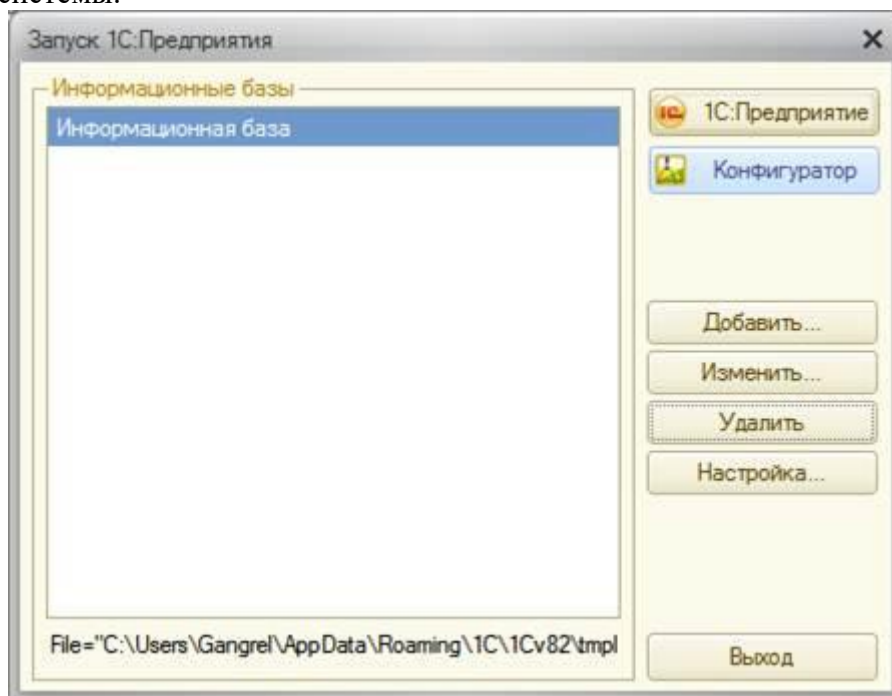


Рис. 1.3. Конфигуратор

Перед вами окно конфигуратора:

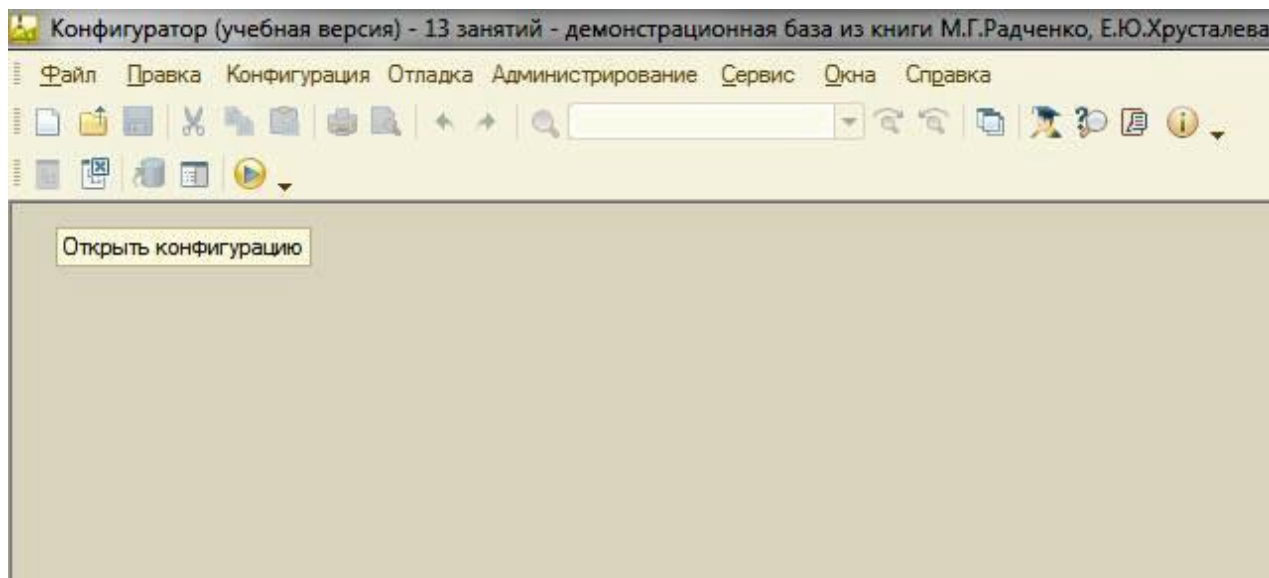


Рис. 1.4. Окно конфигуратора

Именно с помощью этого инструмента мы будем создавать нашу конфигурацию. Сразу под заголовком окна находится главное меню конфигуратора, содержащее пункты **Файл**, **Правка**, **Конфигурация**, **Администрирование** и т.п. В каждом из этих пунктов содержится много подпунктов, вызов которых обеспечивает выполнение различных действий конфигуратора.

Ниже находится панель инструментов конфигуратора, в которую в виде кнопок-пиктограмм помещены наиболее часто используемые действия, вызываемые из меню.

Таким образом, одни и те же действия можно выполнить двумя разными способами: вызвав определенный пункт меню или нажав соответствующую ему кнопку на панели инструментов. Выполним первую команду, с которой начинается работа с любой конфигурацией, - откроем конфигурацию с помощью пункта меню **Конфигурация > Открыть конфигурацию** или соответствующей кнопки на панели инструментов.

На экране *откроется дерево объектов конфигурации*:

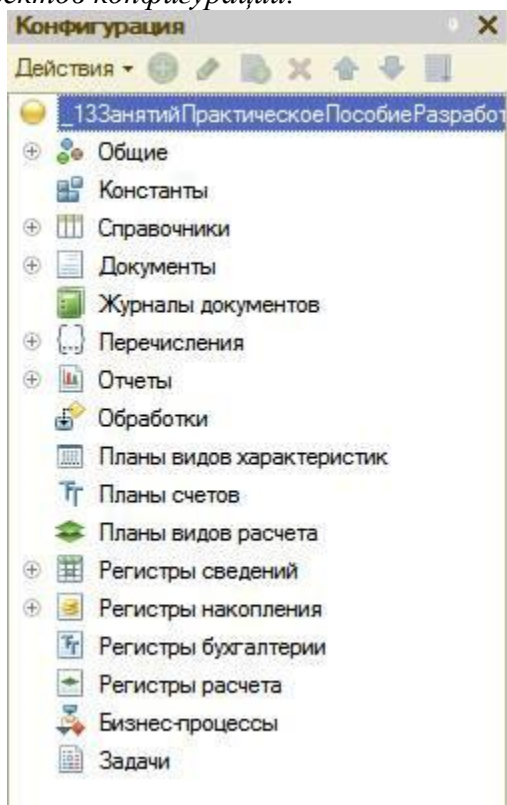


Рис. 1.5. Дерево объектов конфигурации

Можно сказать, что дерево объектов конфигурации - основной инструмент, с которым работает разработчик. Дерево объектов конфигурации содержит в себе практически всю информацию о том, из чего состоит конфигурация.

Что такое объекты конфигурации:

Конфигурация представляет собой описание. Она описывает структуру данных, которые пользователь будет использовать в режиме работы 1С:Предприятие.

Кроме этого, конфигурация описывает всевозможные алгоритмы обработки этих данных, содержит информацию о том, как эти данные должны будут выглядеть на экране и на принтере, и т.д.

С одной стороны, объекты конфигурации представляют собой детали «конструктора», из которого собирается конфигурация. Обычно в конструкторе существует некоторый набор деталей. Детали могут быть разного вида. Теперь представьте, что деталей каждого вида мы можем создавать столько, сколько нам нужно. Мы можем соединять детали между собой различными способами.

То же и с объектами конфигурации. Мы можем создавать только объекты определенных видов. Но каждого вида объектов мы можем создать столько, сколько нам нужно. Объекты одного вида отличаются от объектов другого вида тем, что имеют разные свойства. Объекты могут взаимодействовать друг с другом, и мы можем описать такое взаимодействие.

Объекты конфигурации также обладают различным поведением, и оно зависит от вида объекта. Одни объекты могут выполнять какие-то действия, другие этих действий выполнять не могут, зато у них есть свой собственный набор действий.

«Сложные» объекты конфигурации состоят из более «простых», и одни и те же «простые» объекты могут входить в состав сложных объектов. Такая структура позволяет упростить работу с объектами конфигурации, поскольку если мы знаем, как работать с каким-либо «простым» объектом, то в любом «сложном» объекте, в состав которого он входит, мы будем работать с ним все тем же образом.

Самое важное качество объектов конфигурации - это их прикладная направленность. Объекты конфигурации не просто некие абстрактные конструкции, при помощи которых разработчик пытается описать поставленную перед ним задачу. Они представляют собой аналоги реальных объектов, которыми оперирует предприятие в ходе своей работы.

Как добавить объект конфигурации:

Перед началом работы следует объяснить некоторые приемы работы с конфигуратором:

Для того чтобы открыть и закрыть конфигурацию, следует использовать пункты меню **Конфигурация > Открыть конфигурацию** и **Конфигурация > Закрыть конфигурацию** или соответствующие им кнопки на панели инструментов.

После того как конфигурация открыта, ее состав появляется в окне дерева конфигурации (см. рис. 1.5). Это окно вы можете закрыть, как любое другое окно Windows, при этом конфигурация останется открытой. Чтобы снова отобразить на экране окно дерева конфигурации, следует воспользоваться командой меню **Конфигурация > Окно конфигурации**.

Добавить новый объект конфигурации можно несколькими способами, и вы можете использовать наиболее понятный и удобный для вас.

Первый способ: Необходимо установить курсор на ту ветку объектов конфигурации, которая вас интересует, и в командной панели окна конфигурации нажать кнопку **Действия > Добавить**.

Второй способ. Вы можете воспользоваться контекстным меню, которое вызывается при нажатии на правую клавишу мыши. Установите курсор на интересующую вас ветку объектов конфигурации и нажмите правую клавишу мыши. В появившемся меню выберите пункт **Добавить**.

Третий способ. Установите курсор на интересующую вас ветку объектов конфигурации и в командной панели окна конфигурации нажмите кнопку **«Добавить»** (с пиктограммой «+»).

Палитра свойств:

Палитра свойств - это специальное служебное окно, которое позволяет редактировать все свойства объекта конфигурации и другую связанную с ним информацию. Поскольку разные объекты конфигурации имеют самые разные свойства, содержимое этого окна будет меняться в зависимости от того,

- какой объект является текущим (на каком объекте конфигурации установлен курсор).
- Выделим в дереве объектов конфигурации корневой элемент «**Конфигурация**» и двойным щелчком мыши откроем его палитру свойств.

- Зададим имя конфигурации «**Фамилия**». Соответствующий ему синоним устанавливается автоматически, но его можно изменить по своему усмотрению. В дальнейшем именно его мы будем видеть в рабочем окне 1С предприятия.

В этом случае, как и сейчас, палитра свойств откроется и будет закреплена на рабочей области конфигуратора. То есть при выделении какого-либо объекта конфигурации окно его свойств всегда будет открыто. Однако есть удобная возможность «открепить» палитру свойств, используя символ кнопки в заголовке окна палитры свойств.

Запуск отладки в режиме 1С:Предприятие:

Проверим первые изменения в режиме 1С:Предприятие.

Для этого выполним пункт меню **Отладка > Начать отладку** или нажмем соответствующую кнопку > на панели инструментов конфигуратора. Система сама анализирует наличие изменений в конфигурации и выдает соответствующий вопрос об обновлении конфигурации базы данных. На вопрос конфигуратора ответим «**ДА**», и на экране появится окно 1С:Предприятия.

В режиме 1С:Предприятие:

Внешний вид интерфейса прикладного решения

В заголовке окна видим название нашей конфигурации. Пустое пространство - это рабочая область приложения, которая пока ничем не заполнена.

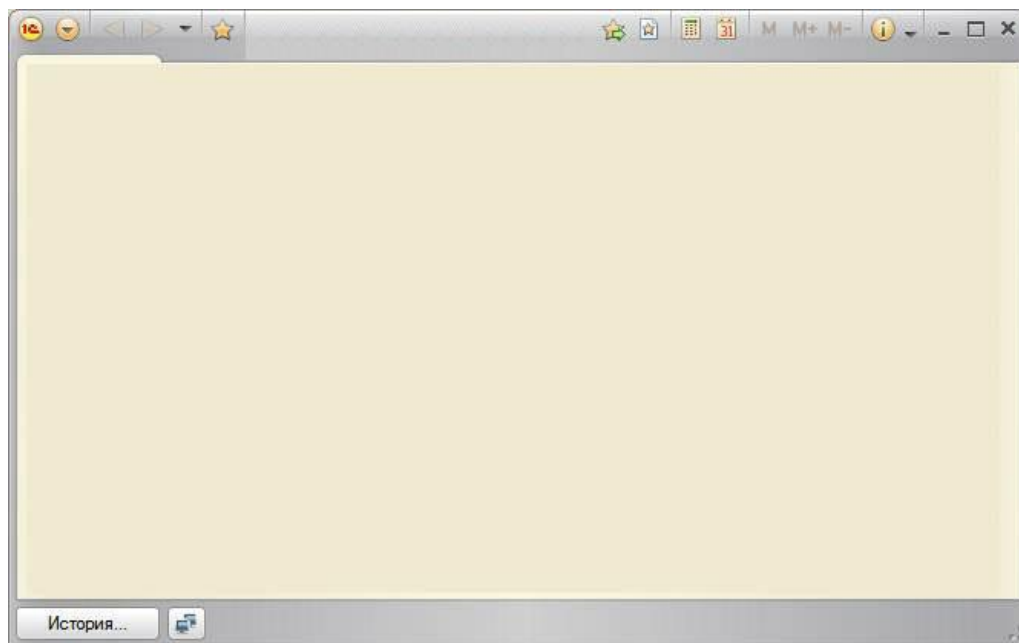


Рис. 1.6. Внешний вид интерфейса прикладного решения

Кроме заголовка конфигурации в окне 1С:Предприятия ничего не появилось. И этого следовало ожидать.

Контрольные вопросы:

Что такое конфигурируемость системы 1С:Предприятие.

Из каких основных частей состоит система.

Что такое платформа и что такое конфигурация.

Для чего используются разные режимы запуска системы 1С. Предприятие.

Что такое дерево объектов конфигурации.

Что такое объекты конфигурации.

Что создает система на основе объектов конфигурации.

Какими способами можно добавить новый объект конфигурации.

Зачем нужна палитра свойств.

Как запустить 1С:Предприятие в режиме отладки.

Лабораторная работа № 2 Подсистемы

Что такое подсистема:

Подсистемы - это основные элементы для построения интерфейса 1С:Предприятия. Поэтому первое, с чего следует начинать разработку конфигурации, - это проектирование состава подсистем. При этом перед разработчиком стоит важная и ответственная задача - тщательно продумать состав подсистем, и затем аккуратно и осмысленно привязать к подсистемам те объекты конфигурации, которые он будет создавать.

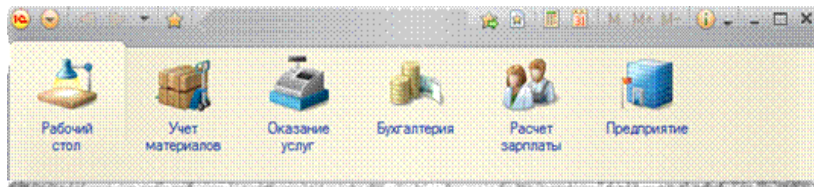


Рис. 2.1. Подсистемы

ПРАКТИЧЕСКАЯ ЧАСТЬ

Каждый объект конфигурации может быть включен в одну или сразу несколько подсистем, в составе которых он будет отображаться.

Сейчас мы создадим в нашей конфигурации пять новых объектов конфигурации Подсистема, которые будут иметь имена: «**Бухгалтерия**», «**РасчетЗарплаты**», «**УчетМатериалов**», «**ОказаниеУслуг**» и «**Предприятие**». Чтобы это сделать, выполним следующие действия.

Добавление подсистемы в режиме «Конфигуратор»:

Чтобы создать новые подсистемы, раскроем ветвь «**Общие**» в дереве объектов конфигурации, нажав на + слева от нее.

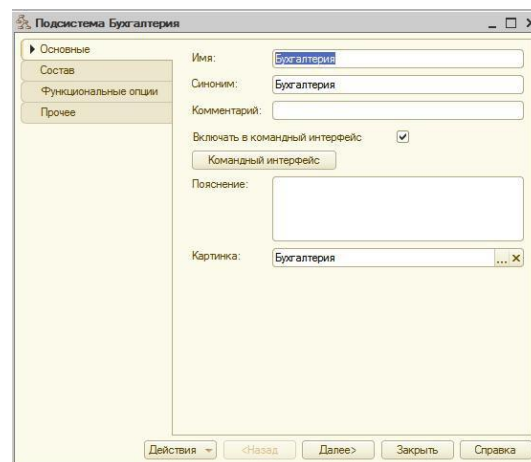
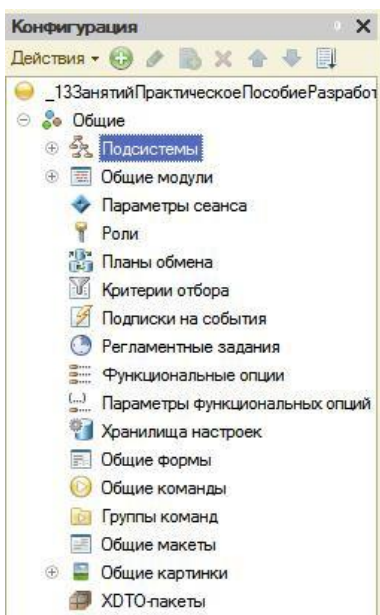
Затем выделим ветвь «**Подсистемы**», вызовем ее контекстное меню и выберем пункт «**Добавить**» или нажмем соответствующую кнопку в командной панели окна конфигурации.

Рис. 2.2. Дерево объектов конфигурации

Рис. 2.3. Подсистема Бухгалтерия

После этого система откроет окно редактирования объекта конфигурации.

Зададим имя подсистемы – «**Бухгалтерия**». На основании имени платформа автоматически создаст



синоним – «**Бухгалтерия**».

Имя и синоним объекта конфигурации:

Имя является основным свойством любого объекта конфигурации. При создании нового объекта система автоматически присваивает ему некоторое имя.

Свойство **Синоним** также есть у любого объекта конфигурации. Оно предназначено для хранения «альтернативного» наименования объекта конфигурации, которое будет использовано в элементах интерфейса нашей программы, то есть будет показано пользователю. Поэтому на синоним практически нет никаких ограничений.

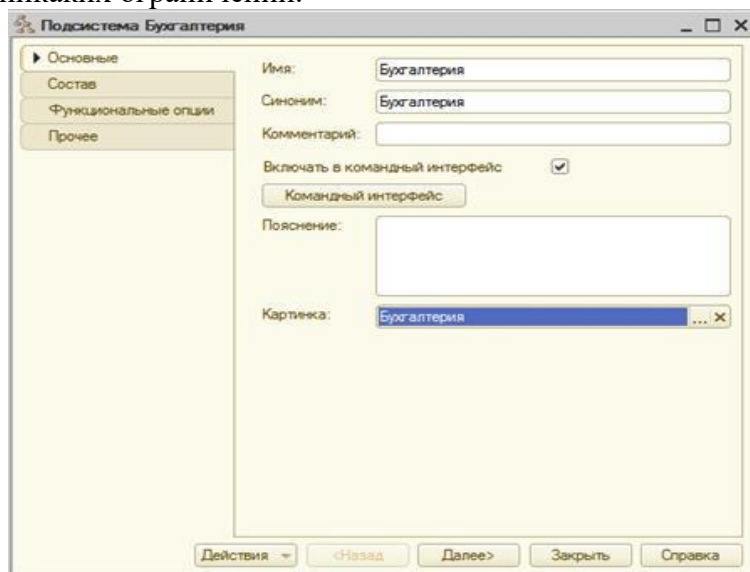


Рис. 2.4. Подсистема Бухгалтерия

В целях усовершенствования интерфейса приложения мы можем также задать картинку для отображения подсистемы.

Система создаст объект конфигурации Общая картинка и откроет окно редактирования его свойств. Картинке можно присвоить свое имя, а так же выбрать свое изображение с локального диска. Для просмотра изображений поставим флажок **«Просмотр»**. Выбранная нами картинка появится в окне редактирования общей картинки. Закроем окно редактирования объекта конфигурации. Итак, мы вернулись в окно редактирования объекта конфигурации Подсистема Бухгалтерия. Мы видим, что выбранная нами одноименная картинка установилась в качестве картинки для подсистемы.

Таким образом, в интерфейсе 1С Предприятия в качестве названия раздела будет показан синоним подсистемы, и над ним будет выводиться указанная картинка.

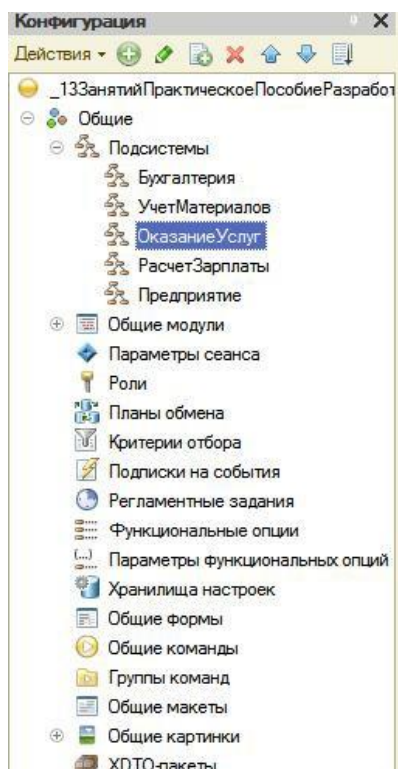


Рис. 2.5. Дерево объектов конфигурации

Снова выделим ветвь «Подсистемы», нажмем кнопку «Добавить» в дереве объектов конфигурации и создадим подсистемы с именами «УчетМатериалов» и «ОказаниеУслуг». Установим для них картинки.

Теперь воспользуемся другим способом для добавления подсистем. Вызовем контекстное меню одной из созданных подсистем. Выберем в нем пункт «Добавить». Он разбивается на два подпункта. Выбор подпункта «Подсистема» позволяет добавить подсистему того же уровня иерархии, что и выделенная. Выбор подпункта «Подчиненная Подсистема» позволяет добавить подсистему, подчиненную выделенной. Поскольку в нашей конфигурации не планируется сложной многоуровневой структуры, выберем первый вариант и добавим подсистему «РасчетЗарплаты». Установим для нее в качестве картинки общую картинку «Зарплата», добавив ее из файла

«Зарплата».

В заключение добавим подсистему Предприятие для доступа к административным и сервисным функциям.

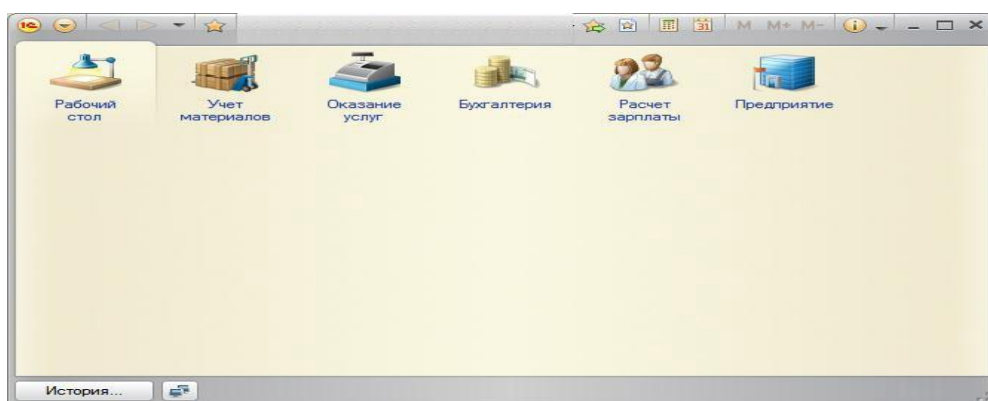


Рис. 2.6. Подсистемы

Панель разделов прикладного решения в режиме 1С:Предприятие:

Запустим 1С Предприятие в режиме отладки и увидим результат наших изменений. Вид разрабатываемого нами приложения изменился. Сразу под главным меню располагается панель разделов приложения, где и отражены созданные нами подсистемы. Причем все разделы выводятся с выбранными в их свойствах картинками. Разделы представлены в форме гиперссылок, нажав на которые пользователь может открыть связанные с ними документы, справочники, отчеты и т. п.

Обратите внимание, что раздел **Рабочий стол** формируется платформой по умолчанию. Он предназначен для размещения наиболее часто используемых пользователем документов, отчетов и т. п.

Порядок разделов в режиме Конфигуратор:

Однако порядок расположения подсистем нас не совсем устраивает. Изменим его. Закроем приложение и вернемся в конфигуратор. Выделим корень дерева объектов конфигурации «**Фамилия**», нажатием правой кнопки мыши вызовем контекстное меню и выберем пункт

«Открыть командный интерфейс конфигурации»

В открывшемся окне «**Командный интерфейс**» вы увидите список созданных вами подсистем (разделов приложения). Расположим сначала подсистемы, отражающие производственную деятельность нашей фирмы: «**Учет материалов**» и «**Оказание услуг**», затем бухгалтерскую деятельность и расчет зарплаты сотрудников: «**Бухгалтерия**» и «**Расчет зарплаты**», а затем подсистему «**Предприятие**».

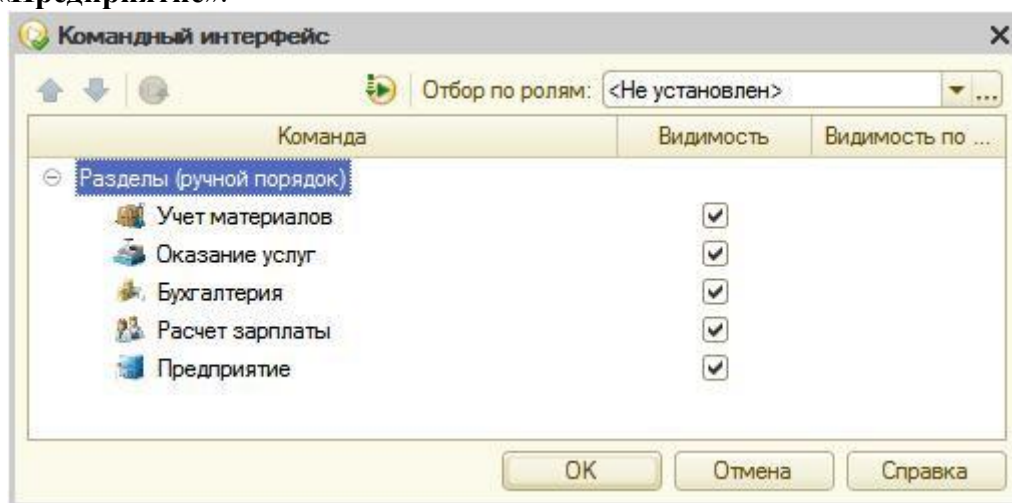


Рис. 2.7. Командный интерфейс

В режиме 1С:Предприятие:

Запустим 1С Предприятие в режиме отладки и увидим, что порядок расположения подсистем в панели разделов приложения изменился так, как мы его задали.

Примечание:

После успешного завершения каждого занятия рекомендуются сохранять конфигурацию, выполнив команду главного меню **Администрирование > Выгрузить информационную базу**. Для загрузки БД выполните: **Администрирование > Загрузить информационную базу**.

Окно редактирования объекта конфигурации и палитра свойств

На первый взгляд окно редактирования объекта и палитра свойств дублируют друг друга. *Окно редактирования объекта конфигурации предназначено в первую очередь для быстрого создания новых объектов.* Окно редактирования объекта помогает быстро создать незнакомый объект конфигурации и обеспечивает удобный доступ к нужным свойствам.

Что же касается палитры свойств, то она предоставляет одну абсолютно незаменимую возможность. Дело в том, что она не привязана по своей структуре к какому-то конкретному виду объектов конфигурации. Ее содержимое меняется в зависимости от того, какой объект является текущим. За счет этого она может «запоминать», какое свойство объекта в ней выбрано, и при переходе в дереве к другому объекту будет подсвечивать у себя все то же свойство, но уже другого объекта.

Контрольные вопросы:

Для чего используется объект конфигурации Подсистема.

Как описать логическую структуру конфигурации при помощи объектов Подсистема.

Как управлять порядком вывода и отображением подсистем в конфигурации.

Что такое окно редактирования объекта конфигурации и в чем его отличие от палитры свойств.

Лабораторная работа № 3 Справочники.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Что такое справочник

Объект конфигурации Справочник предназначен для работы со списками данных. Свойства и структура этих списков описываются в объектах конфигурации Справочник, на основе которых платформа создает в базе данных таблицы для хранения информации из этих справочников.

Справочник состоит из элементов. Пользователь в процессе работы может самостоятельно добавлять новые элементы в справочник.

В базе данных каждый элемент справочника представляет собой отдельную запись в основной таблице, хранящей информацию из этого справочника.

Каждый элемент справочника, как правило, содержит некоторую дополнительную информацию, которая подробнее описывает этот элемент. Поскольку эти объекты конфигурации логически связаны с объектом Справочник, они называются подчиненными этому объекту.

В БД справочник хранится в виде таблицы, в строках которой расположены элементы списка, а каждому реквизиту (стандартному или созданному разработчиком) в этой таблице соответствует отдельный столбец.

Система скрывает от разработчика всю «техническую» часть, связанную с хранением данных: в базе данных для справочника создаются несколько таблиц, эти таблицы связываются по уникальному полю (**Ссылка**), поля таблиц имеют определенные типы и т.д. Все это система делает сама. Нам лишь нужно добавить в объект конфигурации Справочник подчиненный ему объект **«Табличная часть»**.

Для удобства использования элементы справочника могут быть сгруппированы по какому-либо принципу. Группировка называется иерархией, выделяют два вида иерархии:

- иерархия групп и элементов (*в качестве родителя выступает группа элементов справочника*)
- иерархия элементов (родитель один из элементов справочника)

Элементы одного справочника могут быть подчинены элементам или группам другого справочника. В системе 1С: Предприятие это достигается путем указания списка владельцев справочника для каждого объекта конфигурации Справочник.

Порой возникают ситуации, когда необходимо, чтобы в справочнике некоторые элементы существовали всегда, независимо от действий пользователя. Объект конфигурации Справочник позволяет описать любое количество таких элементов справочника. Они называются *предопределенными элементами справочника*.

Предопределенные элементы отличаются от обычных тем, что они создаются в конфигураторе и что пользователь не может их удалить. Все остальные действия с ними он делать может, в том числе и переименовывать. В интерфейсе предопределенные элементы справочника помечены специальной пиктограммой.

Предопределенные элементы

Выделяют следующие особенности предопределенных элементов:

- на предопределенные элементы могут опираться алгоритмы работы конфигурации (т. е. возможно обращение к ним из встроенного языка по имени);
- предопределенные элементы являются объектами базы данных, которые нельзя удалить в режиме 1С Предприятия.

Обычные элементы «непостоянны» для конфигурации. В процессе работы пользователя они могут появиться, исчезнуть. Поэтому конфигурация хоть и может отличить их друг от друга, но рассчитывать на них в выполнении каких-либо алгоритмов она не может в силу их «непостоянства».

Предопределенные элементы, напротив, «постоянны». В процессе работы пользователя они находятся

всегда на своих местах и исчезнуть не могут.

Основная конфигурация и конфигурация базы данных

Конфигурация, предназначенная для разработчика, называется Основная конфигурация. Конфигурация, с которой работают пользователи, называется Конфигурация базы данных. Основную конфигурацию можно редактировать. Конфигурацию базы данных редактировать нельзя, можно только произвести обновление конфигурации базы данных на основе основной конфигурации. Хранилище конфигурации содержит конфигурацию, предназначенную для групповой разработки. Она хранится не в виде единой конфигурации, а в виде отдельных объектов в разрезе версий конфигурации. Таким образом, мы можем получить из хранилища конфигурацию любой версии - для этого она «собирается» из объектов нужной версии.

Теперь представьте, что между всеми этими видами конфигураций существует возможность сравнения и обновления. В этом случае очень легко запутаться, и название Основная конфигурация как нельзя лучше отражает конечную цель всех изменений.

Если эти изменения не затрагивают структуру базы данных (например, если не нужно изменять таблицы, если поменялся только программный код в каком-то модуле), то обновить конфигурацию базы данных можно не прерывая работы пользователей - динамическое обновление.

Но если изменения касаются структуры базы данных, например, добавился новый реквизит у справочника или изменился тип существующего реквизита, то тогда нужно завершить работу всех пользователей. Разработчик всегда может сравнить основную конфигурацию и конфигурацию базы данных, может вернуться к исходному состоянию основной конфигурации, используя конфигурацию базы данных.

Формы справочника:

В зависимости от того, какие действия мы хотим выполнять со справочником, нам требуется изображать справочник в «разном виде».

Система может самостоятельно сгенерировать все формы, которые нужны для представления данных, содержащихся в справочнике. Причем система «знает», какие именно формы нужно использовать в каких ситуациях.

Вообще говоря, для отображения справочника в различных ситуациях требуется максимум пять форм для справочника.

Таблица 3.1. Формы справочника

В контекстном меню и в палитре свойств (Рис.3.3)	В конструкторе форм (Рис. 3.2)	На закладке формы (Формы...) (Рис.3.1)
Форма объекта	Форма элемента справочника	Элемента
Форма группы	Форма группы справочника	Группы
Форма списка	Форма списка справочника	Списка
Форма для выбора	Форма выбора справочника	Выбора
Форма для выбора группы	Форма выбора группы справочника	Выбора группы

Обратите внимание, что в различных местах configurатора одни и те же формы называются немного по-разному. Дело в том, что в контекстном меню и палитре свойств отображаются свойства объектов конфигурации. Они одинаковые для всех объектов конфигурации. А в конструкторе форм и на закладке формы отображаются представления этих свойств в более понятном виде.

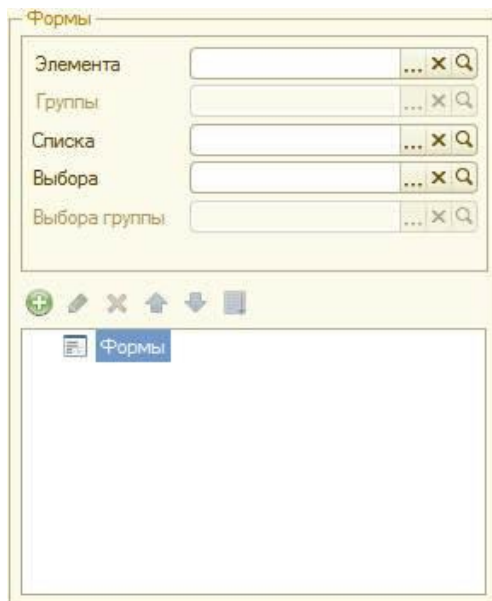


Рис.3.1 Формы

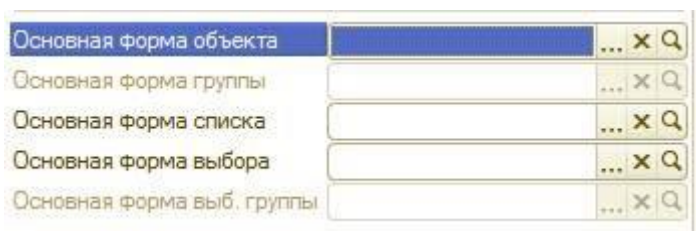
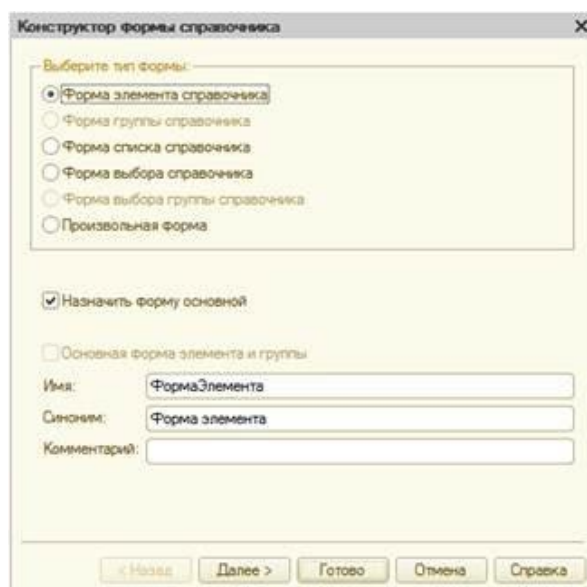


Рис.3.2 Форма элемента справочника

Рис.3.3 Основная форма объекта

«**Форма элемента**» используется для редактирования или создания элемента справочника.

«**Форма группы**» используется для редактирования или создания группы справочника. Группа, как правило, содержит гораздо меньше информации, чем сам элемент справочника. Поэтому для нее нужна отдельная форма, отличная от формы элемента.

«**Форма списка**» используется для отображения списка элементов справочника.

«**Форма выбора**» используется для того, чтобы в поле некоторой формы выбрать один из элементов справочника. При этом форма выбора проще, чем форма списка, так как в форме списка может

показываться много реквизитов. А при выборе элемента (в документе, например), нам нужно знать только наименование. Поэтому можно для выбора использовать отдельную, более простую форму. «**Форма выбора группы**» используется, когда в поле некоторой формы нужно выбрать не просто элемент справочника, а одну из его групп. При этом форма выбора группы проще, чем форма выбора элемента, так как группа, как правило, содержит гораздо меньше информации, чем сам элемент справочника.

Любая форма может быть описана в конфигураторе. Для создания такого описания существует подчиненный объект конфигурации Форма, она подчинена одному из прикладных объектов, но может существовать и самостоятельно.

Форма служит для «визуализации» данных, находящихся в базе данных. Она представляет эти данные в удобном для пользователя виде и позволяет описать алгоритмы, которые будут сопровождать работу пользователя с данными, показанными в форме.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Для начала нам понадобится список сотрудников предприятия, которые будут оказывать услуги. Затем нам будет нужен список клиентов, с которыми работает наше предприятие. После этого нам понадобится перечень услуг, которые может оказывать наше предприятие, и список материалов, которые могут быть израсходованы.

Кроме этого, нам потребуется список складов, на которых могут находиться материалы.

1. Создание простого справочника

Создадим справочник, в котором будут храниться наименования наших клиентов

- **В режиме Конфигуратор:**

Откроем в конфигураторе конфигурацию, выделим в дереве объектов конфигурации ветвь «Справочники» и нажмем кнопку «Добавить» в командной панели окна конфигурации.

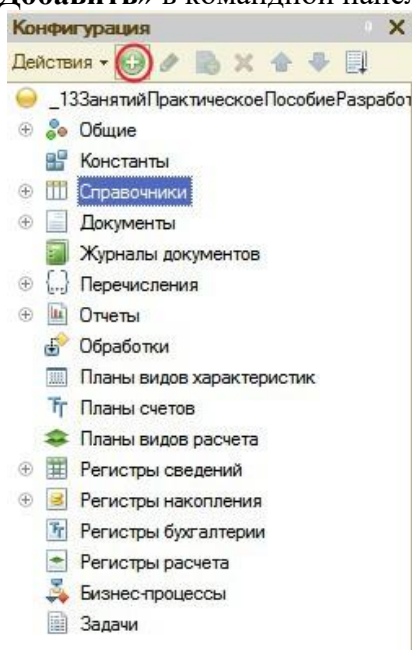


Рис. 3.4. Дерево объектов конфигурации

В открывшемся окне редактирования объекта конфигурации зададим имя справочника - **Клиенты**. На основании имени платформа автоматически создаст синоним - **Клиенты**. Напомним, что свойство **Синоним** служит для представления объекта в интерфейсе нашей программы.

Также у разработчика есть возможность установки дополнительных свойств, определяющих пользовательское представление объектов. Эти свойства задавать не обязательно.

- **Представления объекта в конфигурации:**

Представление объекта определяет название объекта в единственном числе и используется в названии стандартной команды, например, команды создания объекта - **Клиент: создать**. Представление объекта нужно задавать тогда, когда синоним объекта конфигурации задан во множественном числе, или когда он описывает множество объектов. Потому что в интерфейсе автоматически формируются команды

открытия списка справочника и команды создания нового элемента справочника.

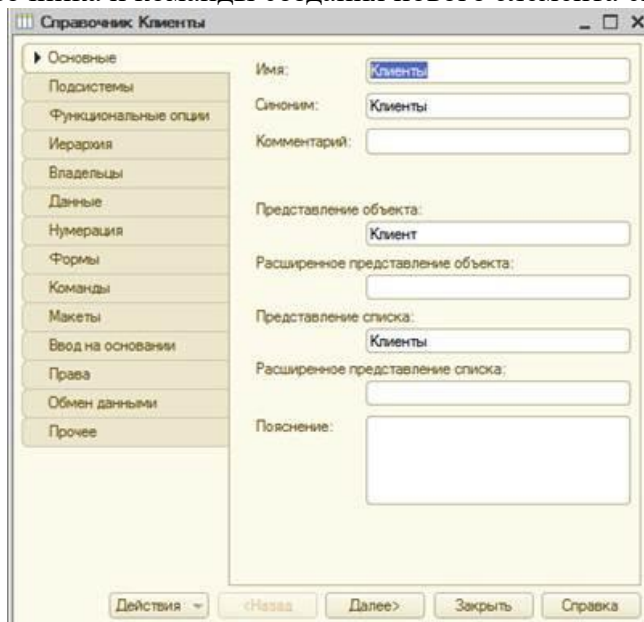


Рис. 3.5. Справочник Клиенты

Представление объекта используется для того, чтобы описать, как будет выглядеть в интерфейсе команда добавления нового клиента. Также оно будет использовано в заголовке формы и в представлении ссылки на клиента.

Расширенное представление объекта определяет заголовок формы объекта, например формы для создания нового элемента справочника. Если это свойство не задано, то вместо него используется свойство **Представление объекта**.

Представление списка определяет название списка объектов и используется в названии стандартной команды, например, команды открытия списка объектов - **Клиенты: открыть**. Представление списка нужно задавать тогда, когда синоним задан в единственном числе.

Расширенное представление списка определяет заголовок формы списка, например формы списка справочника. Если это свойство не задано, то вместо него используется свойство **Представление списка**.

В лабораторной работе зададим два свойства **Представление объекта - Клиент** и **Представление списка - Клиенты**. Последнее можно было и не задавать, так как синоним справочника совпадает со свойством **Представление списка**.

- **Принадлежность объекта к подсистемам:**

Нажмем кнопку **Далее** и перейдем на закладку **Подсистемы**. На этой закладке определяется, в каких подсистемах будет отображаться данный справочник.

В списке подсистем мы видим подсистемы, созданные ранее при определении структуры приложения.

Логично предположить, что список клиентов должен быть доступен в разделе **Оказание услуг**.

Бухгалтерская отчетность, формируемая в разделе **Бухгалтерия**, также может быть представлена в разрезе клиентов.

Поэтому отметим в списке подсистемы **Бухгалтерия** и **ОказаниеУслуг**.

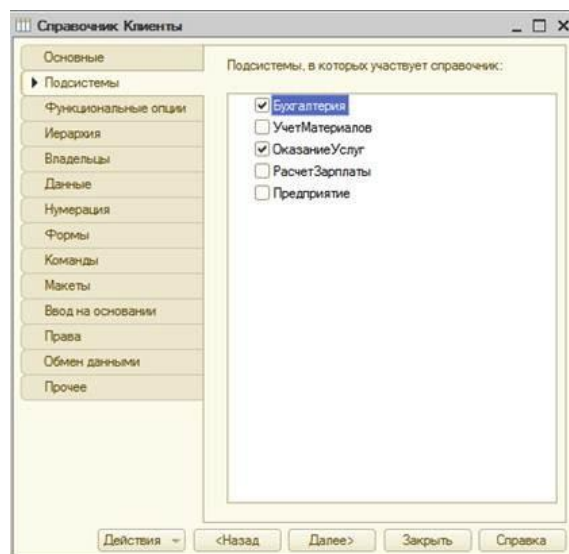


Рис. 3.6. Справочник Клиенты

- **Код и наименование справочника**

В окне редактирования объекта конфигурации Справочник нажмем на вкладку **Данные**. Здесь для нас представляют интерес длина кода и длина наименования.

Длина кода - количество элементов, содержащихся в справочнике.. Как правило, код справочника используется для идентификации элементов справочника и содержит уникальные для каждого элемента справочника значения. Платформа может сама контролировать уникальность кодов и поддерживать автоматическую нумерацию элементов справочника.

Зададим длину кода - 9 символов, а длине наименования увеличим длину до 50.

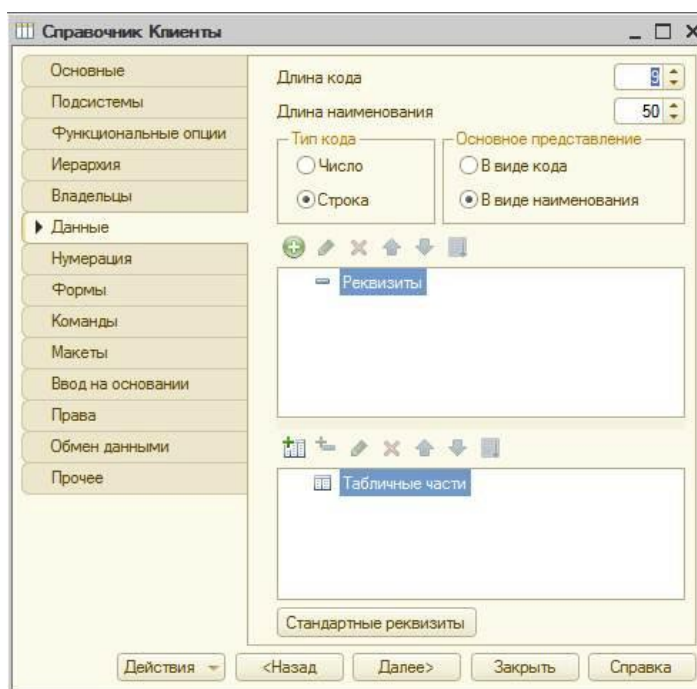


Рис. 3.7. Справочник Клиенты

- **Команда добавления нового элемента**

Прежде чем запускать 1С предприятие, настроим интерфейс приложения, чтобы было удобнее вводить новые элементы справочника.

Команда для открытия списка справочника, как и команда для создания его новых элементов, добавляется в интерфейс тех разделов (подсистем), в которых будет отображаться справочник. Но команда создания новых элементов по умолчанию невидима в интерфейсе приложения.

Сделаем доступной в панели действий раздела **ОказаниеУслуг** стандартную команду для создания новых клиентов.

Для этого в дереве объектов конфигурации выделим ветвь **Подсистемы**, вызовем ее контекстное меню и выберем пункт **Все подсистемы**, выделим подсистему **ОказаниеУслуг**. В списке **Командный**

интерфейс отразятся все команды выбранной подсистемы. В группу **Панель действий.Создать** добавилась команда **Клиент: создать** для создания нового элемента справочника, включим видимость у этой команды.

*Для подсистемы **Бухгалтерия** никаких команд добавлять в панель действий не будем, так как это определяется прикладной логикой работы.*

Закроем окно редактирования справочника **Клиенты** и запустим 1С предприятие в режиме отладки.

Ответим утвердительно на запрос конфигуратора об обновлении конфигурации и нажмем кнопку

Принять

- **В режиме 1С: Предприятие. Панель навигации и панель разделов**

Перед нами откроется окно системы в режиме 1С:Предприятие. Мы видим, что если перейти в раздел **Оказание услуг** или **Бухгалтерия**, то слева в вертикальной области окна появится панель навигации (см рис).

Панель навигации отображает структуру выбранного раздела, предназначена для быстрого перехода к различным спискам в пределах выбранного раздела программы.

Также в разделе **Оказание услуг** появилась панель действий. Панель действий содержит команды, которые соответствуют текущему разделу, выбранному в панели разделов.

Сейчас в панели действий раздела **Оказание услуг** в группе **Создать** доступна команда для создания элементов нашего первого справочника **Клиенты**. Этой командой мы и воспользуемся для создания новых элементов справочника, не открывая при этом списка клиентов.



Рис. 3.8. Панель навигации и панель разделов

- **Создание элементов справочника**

Пока наш справочник пуст, поэтому добавим в него несколько элементов. Для этого выполним команду **Клиент** в панели действий раздела **Оказание услуг**.

Перед нами откроется форма для создания элемента справочника.

Внесем последовательно в справочник 3-х новых клиентов. Код вносить не нужно, так как он генерируется автоматически. После последовательного добавления клиентов нажмем **Записать и закрыть**.

Чтобы открыть существующий элемент справочника для редактирования, нужно дважды щелкнуть на нем мышью. А кроме этого, нажав на ссылку в информационной панели, мы тоже можем открыть для редактирования один из последних измененных элементов справочника.

2. Создание справочника с табличной частью

Теперь мы можем перейти к созданию второго справочника, который будет использоваться в нашей конфигурации, - справочника **Сотрудники**.

Этот справочник будет устроен несколько сложнее, чем справочник **Клиенты**. Дело в том, что в нем мы будем хранить не только фамилию, имя и отчество сотрудника, но и информацию о его прошлой трудовой деятельности.

- **В режиме Конфигуратор**

Добавим новый объект конфигурации Справочник. Назовем его **Сотрудники**.

На основании имени платформа автоматически заполнит его синоним.

Зададим **Представление объекта** как **Сотрудник**. **Представление списка** устанавливать не будем, а **Расширенное представление списка** зададим как Список сотрудников.

Нажмем кнопку **Далее** и перейдем на закладку **Подсистемы**.

По логике нашей конфигурации список сотрудников должен быть доступен в разделах **Оказание услуги** и **Расчет зарплаты**.

Перейдем на закладку **Данные**. Оставим по умолчанию длину и тип кода, длину наименования справочника зададим равной 50 символам.

- **Табличная часть**

Наша задача - создать справочник, имеющий табличную часть. Поэтому добавим в справочник новую табличную часть с именем **ТрудоваяДеятельность**.

Для этого нажмем кнопку **Добавить табличную часть** над списком табличных частей справочника.

Зададим имя – **ТрудоваяДеятельность**.

После чего добавим реквизиты табличной части, для этого нажмем на кнопку **Добавить реквизит**.

Добавим следующие реквизиты:

- **Организация** - тип **Строка**, длина 100;
- **НачалоРаботы** - тип **Дата**, состав даты - **Дата**;
- **ОкончаниеРаботы** - тип **Дата**, состав даты - **Дата**;
- **Должность** - тип **Строка**, длина 100.

В заключение отредактируем командный интерфейс, чтобы нам было удобнее вводить новые элементы справочника. Сделаем видимой в панели действий подсистемы **РасчетЗарплаты** стандартную команду для создания новых сотрудников.

Сделаем это в дереве объектов конфигурации в ветви **Подсистемы\Все подсистемы**:

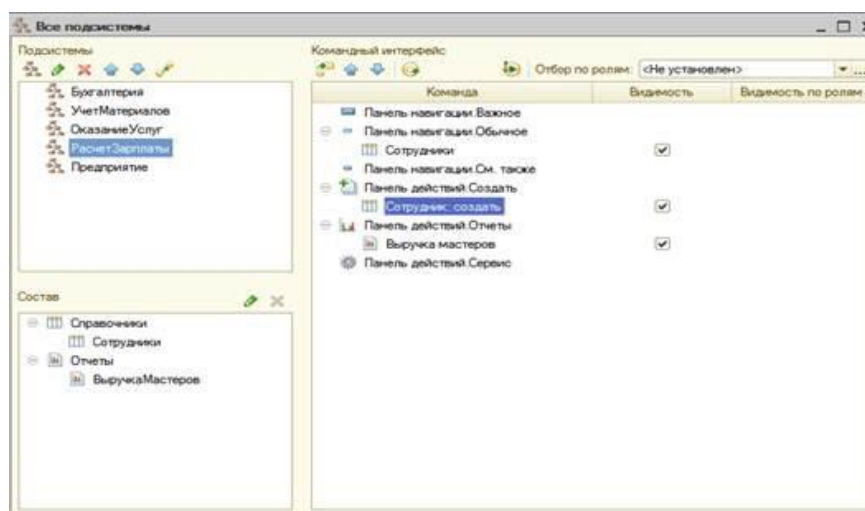


Рис. 3.9. Все подсистемы

Для подсистемы **ОказаниеУслуг** никаких команд добавлять в панель действий не будем, так как вряд ли понадобится пополнять список сотрудников в этом разделе. На этом создание справочника **Сотрудники** завершено.

Обновим конфигурацию, подтвердим обновление и запустим программу в режиме **Предприятие**.

- **Заполнение табличной части в режиме 1С: Предприятие**

В панели действий раздела **Расчет зарплаты** появилась команда **Сотрудник** для создания новых сотрудников. Этой командой мы и воспользуемся для создания новых элементов справочника, не открывая при этом списка сотрудников. Выполним команду **Сотрудник**. Перед нами откроется форма для создания элемента справочника - основная форма объекта. Заголовок этой формы определяется свойством **Представление объекта**.

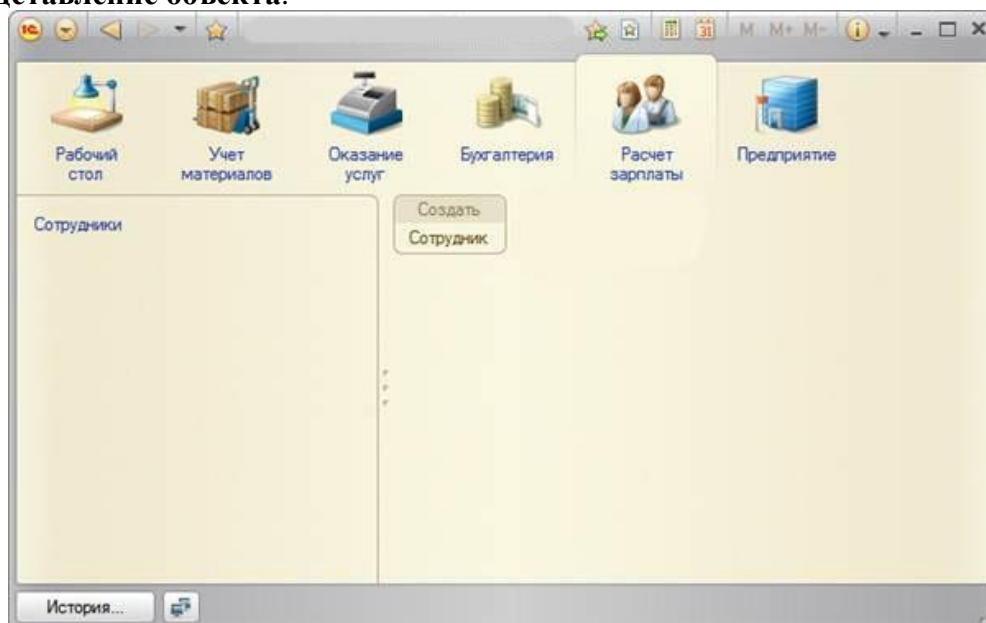


Рис. 3.10. Заполнение табличной части

Эта форма содержит табличную часть с реквизитами, которые мы описали в конфигураторе для этого справочника. Теперь приступим к созданию сотрудников, в этом вам поможет ваша фантазия (для корректной работы заполните все требующиеся поля таблицы)



Рис. 3.11. Заполнение элемента справочника "Сотрудники"

3. Создание иерархического справочника

Справочник **Номенклатура** будет содержать информацию об услугах, которые оказывает наше предприятие, и о тех материалах, которые при этом могут быть использованы.

Этот справочник не будет сложным. Единственная особенность, которой он будет обладать, - наличие иерархической структуры. Для того чтобы справочником было удобно пользоваться, мы сгруппируем услуги в одну группу, а материалы - в другую.

- **В режиме Конфигуратор**

Создадим новый объект конфигурации Справочник и назовем его **Номенклатура**. На основании имени платформа автоматически заполнит его синоним.

Поскольку понятие **Номенклатура** не имеет единственного числа, больше никаких свойств, определяющих представление объекта в интерфейсе приложения, задавать не будем. Вместо **Представления объекта** и **Представления списка** будет использоваться **Синоним** объекта - **Номенклатура**.

- На закладке **Подсистемы**: По логике нашей конфигурации список номенклатуры должен быть доступен в разделах **Учет материалов**, **Оказание услуг** и **Бухгалтерия**, поэтому отметим в списке подсистем эти подсистемы.

- На закладке **Иерархия**: установим флажок **Иерархический справочник**.

- На закладке **Данные**: оставим по умолчанию длину и тип кода, длину наименования справочника зададим равной 100 символам.

- В дереве объектов конфигурации (**Подсистемы\Все подсистемы**): в открывшемся окне слева в списке **Подсистемы** выделим подсистему **УчетМатериалов**. В группе **Панель действий.Создать** включим видимость у команды **Номенклатура: создать**.

Выделив в списке подсистем **ОказаниеУслуг**, сделаем те же действия.

А для подсистемы **Бухгалтерия** никаких команд добавлять в панель действий не будем, так как вряд ли понадобится пополнять список номенклатуры в этом разделе.

Теперь заполним справочник **Номенклатура**. В процессе заполнения вы научитесь создавать группы и переносить элементы из одной группы в другую.

- **В режиме 1С:Предприятие**

В открывшемся окне 1С:Предприятия мы видим, что в панели навигации разделов **Учет материалов**, **Оказание услуг** и **Бухгалтерия** появилась команда **Номенклатура**.

Название этой команды определяется синонимом объекта, так как других представлений мы для этого справочника не задавали.

Выполним команду **Номенклатура** в панели навигации раздела **Учет материалов**. Справа от панели навигации в рабочей области окна приложения откроется основная форма списка.

- Создадим две группы в корне справочника: **Материалы** и **Услуги**.

Для этого нажмем кнопку **Создать группу** в командной панели формы списка.

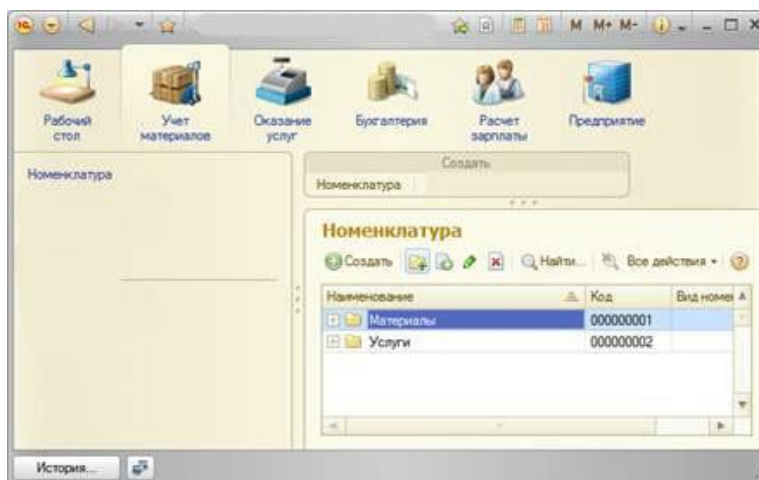


Рис. 3.12. Создание групп в корне справочника

А) Зададим наименование групп: **Материалы**, Услуги. Поля Родитель и Код заполнять не будем.

Б) Раскроем группу **Материалы** (нажатием на +) и создадим в ней пять элементов. После этого раскроем группу **Услуги** и тоже создадим в ней несколько элементов (не более 3).

- *Перенос элементов в другие группы*

Разнесите услуги по двум смысловым группам: Для этого в группе Услуги необходимо создать еще две подчиненных группы, которые по смыслу относятся к предоставляемым услугам.

Для того чтобы переместить услуги в соответствующие группы, в окне списка установим курсор на ту услугу, которую мы хотим переместить, и выполним команду **Все действия > Переместить в группу**. В открывшемся окне выберем новую группу.

4. Создание справочника с предопределенными элементами

В заключение мы создадим справочник **Склады**. Этот справочник будет включать в себя один предопределенный элемент - склад **Основной**, на который будут поступать все материалы. *Наша задача - создать справочник, содержащий предопределенные элементы.*

• В режиме Конфигуратор

Откроем конфигуратор и создадим новый объект конфигурации Справочник с именем **Склады**. На основании имени платформа автоматически заполнит его синоним.

- На вкладке **Основные**: Зададим **Представление объекта - Склад. Представления списка - Склады**.

- На вкладке **Подсистемы**: Отметим в списке подсистем подсистемы **Оказание услуг** и **Учет материалов**.

- На вкладке **Формы**: установим флажок в поле **Быстрый выбор**.

Свойство **Быстрый выбор** как раз позволяет выбирать элементы не из отдельной формы, а из небольшого выпадающего списка, заполненного элементами этого справочника. Этот вариант наиболее удобен для списка складов, так как их, вероятно, будет немного.

- На вкладке **Прочее**: нажмем кнопку **Предопределенные**.

Система откроет список предопределенных элементов справочника.

Сейчас он пуст, поэтому нажмем кнопку **Добавить** и создадим предопределенный элемент с именем

Основной.

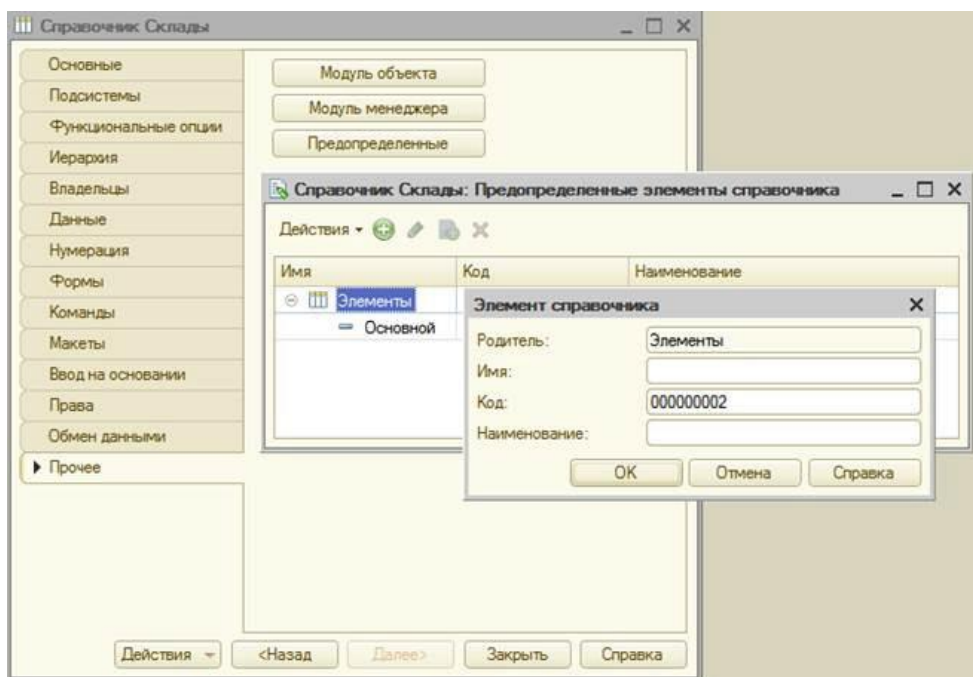


Рис. 3.13. Создание справочника с предопределенными элементами

- В дереве объектов конфигурации (**Подсистемы**\Все подсистемы): в открывшемся окне слева в списке **Подсистемы** выделим подсистему **Учет Материалов**. В группе **Панель действий.Создать** включим видимость у команды **Склад: создать**.

- **В режиме 1С: Предприятие**

В открывшемся окне 1С:Предприятия мы видим, что в панели действий раздела **Учет материалов** появилась команда **Склад** для создания новых складов. Также в панели навигации разделов **Оказание услуг** и **Учет материалов** появилась команда **Склады** для открытия списка складов.

- Выполним команду **Склады** в панели навигации раздела **Учет материалов**.

В списке складов уже есть один элемент с наименованием **Основной**. Это предопределенный элемент, который мы создали в конфигураторе.

Выполнив команду **Склад** в панели действий, добавим в справочник еще один склад, который назовем

Розничный.

Контрольные вопросы

- Для чего предназначен объект конфигурации *Справочник*.
- Каковы характерные особенности справочника.
- Для чего используются реквизиты и табличные части справочника.
- Зачем нужны иерархические справочники и что такое родитель.
- Зачем нужны подчиненные справочники и что такое владлец.
- Какие основные формы существуют у справочника.
- Что такое предопределенные элементы.
- Чем с точки зрения конфигурации отличаются обычные элементы справочника от предопределенных элементов.
- Как пользователь может отличить обычные элементы справочника от предопределенных элементов.
- Как создать объект конфигурации *Справочник* и описать его структуру.
- Как добавить новые элементы в справочник.
- Как создать группу справочника.
- Как переместить элементы из одной группы справочника в другую.
- Зачем нужна основная конфигурация и конфигурация базы данных.
- Как изменить конфигурацию базы данных.
- Как связаны объекты конфигурации и объекты базы данных.
- Что такое подчиненные объекты конфигурации.
- Зачем нужна проверка заполнения у реквизитов справочника.
- Что такое быстрый выбор и когда его использовать.

- Как отобразить справочник и определить его представление в различных разделах интерфейса приложения.
- Как отобразить команды создания нового элемента справочника в интерфейсе подсистем.
- Как редактировать командный интерфейс подсистем.

Лабораторная работа № 4. Документы.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Документ

Объект конфигурации Документ предназначен для описания информации о совершенных хозяйственных операциях или о событиях, произошедших в жизни организации вообще. Как правило, в работе любой фирмы используются такие документы, как приходные накладные, приказы о приеме на работу, платежные поручения, счета и т.д. Свойства и структура этих документов описываются в объектах конфигурации Документ, на основе которых платформа создает в базе данных таблицы для хранения информации из этих документов.

Логика работы документов отличается от логики работы других объектов конфигурации. Документ обладает способностью проведения. Факт проведения документа означает, что событие, которое он отражает, повлияло на состояние учета.

До тех пор, пока документ не проведен, состояние учета неизменно, и документ - не более чем черновик, заготовка. Как только документ будет проведен, изменения, вносимые документом в учет, вступят в силу и состояние учета будет изменено.

Поскольку документ вносит изменения в состояние учета, он всегда «привязан» к конкретному моменту времени. Это позволяет отражать в базе данных фактическую последовательность событий.

Каждый документ содержит информацию, которая подробнее описывает этот документ. Набор такой информации является одинаковым для всех документов одного вида, и для описания такого набора используются реквизиты объекта конфигурации Документ, являющиеся подчиненными объектами конфигурации. Большинство реквизитов разработчик создает самостоятельно, однако у каждого объекта Документ по умолчанию существуют стандартные реквизиты. Два наиболее важных из них - это Дата и Номер. Для описания конкретной информации служат табличные части объекта Документ. В этом случае в базе данных будут созданы дополнительные таблицы для хранения табличных частей, подчиненных конкретному документу.

Формы документа

Для «визуализации» документа существует несколько основных форм, которые имеют несколько вариантов названий (табл. 4.1).

В контекстном меню и в палитре свойств	В конструкторе форм	На закладке формы
Форма объекта	Форма документа	Документа
Форма списка	Форма списка документа	Списка
Форма для выбора	Форма выбора документа	Выбора

Таблица 4.1. Основные формы документа.

Типы данных. Типообразующие объекты конфигурации

На предыдущем занятии, когда мы создавали реквизиты справочников или табличных частей, мы всегда указывали тип значения, которое может принимать этот реквизит. Это были примитивные типы данных: Число, Строка, Дата. *Примитивные типы данных изначально определены в системе, и их набор ограничен.*

Наряду с такими изначально определенными в любой конфигурации типами могут существовать типы данных, определяемые только конкретной конфигурацией.

Объекты конфигурации, которые могут образовывать новые типы данных, называются типобразующими. Эти типы данных не поддерживаются платформой изначально и существуют только в конкретном прикладном решении.

Это небольшое отступление было необходимо потому, что уже при создании первого документа мы столкнемся с использованием типов данных СправочникСсылка.Склады и СправочникСсылка.Номенклатура, которые появились в нашей конфигурации в результате создания объектов конфигурации Справочник Склады и Номенклатура.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Создание Документа «Приходная накладная»

Для отражения событий, происходящих на предприятии, в базе данных мы создадим два документа:

Приходная накладная и Оказание услуги.

Документ **Приходная накладная** будет фиксировать факт поступления в нашу организацию необходимых материалов, а документ **Оказание услуги** - фиксировать оказание услуг и расход материалов, которые используются при оказании этих услуг.

- **В режиме конфигуратор:**

Откроем конфигуратор и добавим новый объект конфигурации Документ.

- *На закладке Основные:* зададим имя документа - ПриходнаяНакладная. На основании имени платформа автоматически заполнит его синоним. Так же зададим Представление списка, как Приходные накладные.

- *На закладке Подсистемы:* список приходных накладных должен быть доступен в разделах Учет материалов и Бухгалтерия.

- На закладке *Данные*: создадим реквизит документа с именем Склад (нажать на «+»); далее выберем для реквизита ссылочный тип данных СправочникСсылка.Склады. Этот тип стал доступен в конфигурации после создания объекта конфигурации Справочник Склады. В качестве значения свойства **Значение заполнения** выберем predetermined элемент справочника Склады - Основной.

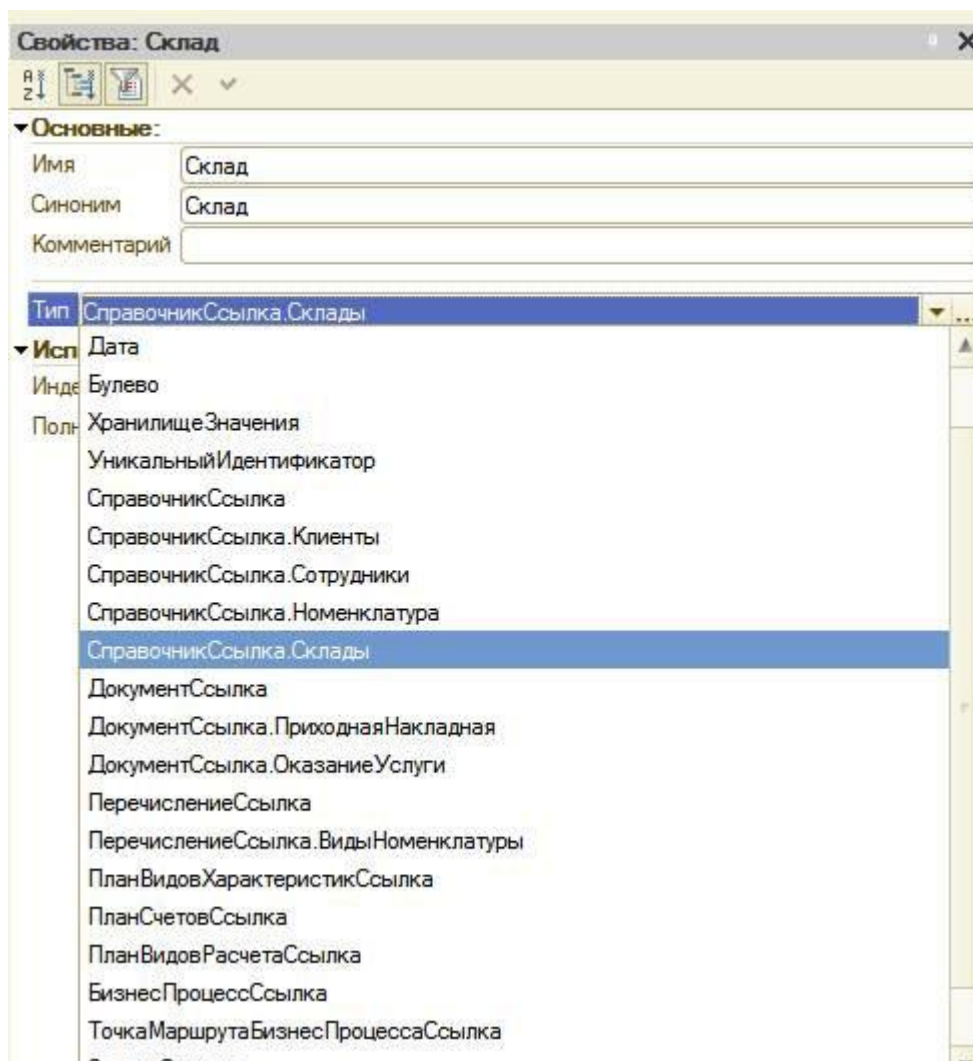


Рис. 4.1. Справочник Склады

После этого добавим в документ табличную часть с именем Материалы. Для этого нажмем кнопку *Добавить табличную часть* над списком табличных частей документа. Кроме имени табличной части установим свойство *Проверка заполнения* в значение *Выдавать ошибку*.

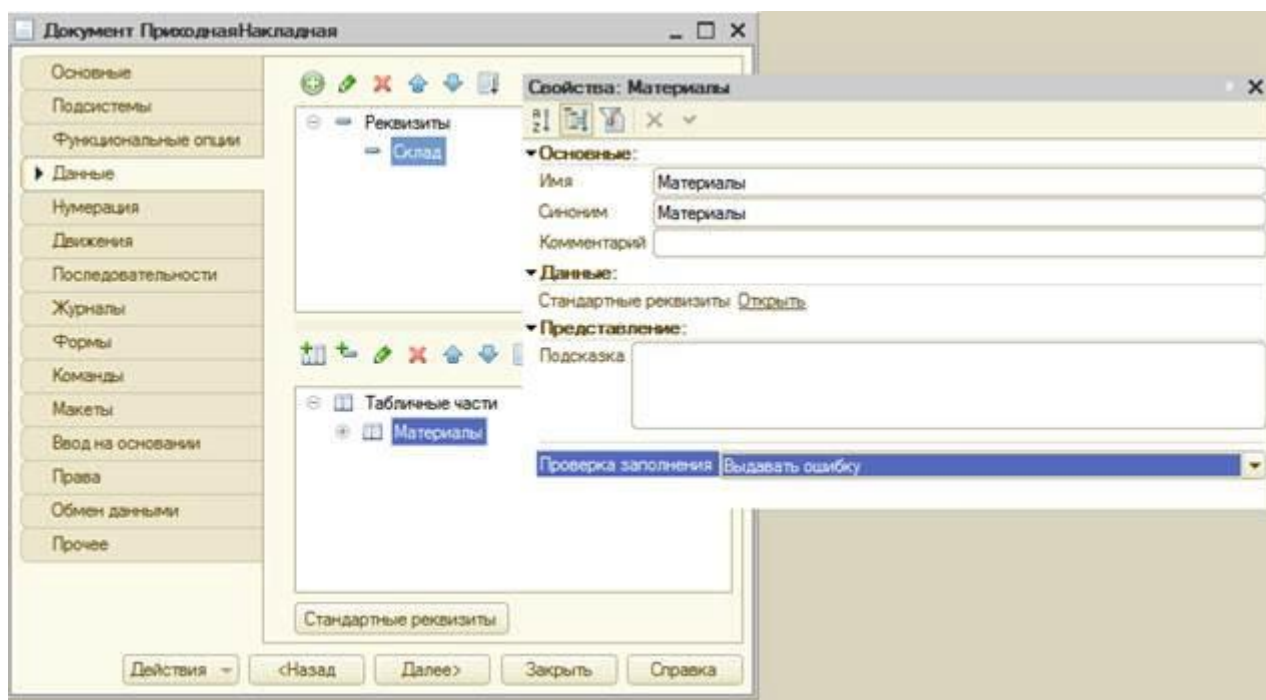


Рис. 4.2. Свойства: Материалы

Создадим реквизиты табличной части Материалы. Для этого нажмем кнопку **Добавить реквизит** в разделе описания табличных частей документа:

- **Материал**, тип **СправочникСсылка. Номенклатура**;
- **Количество**, тип **Число**, длина 15, точность 3, неотрицательное;
- **Цена**, тип **Число**, длина 15, точность 2, неотрицательное;
- **Сумма**, тип **Число**, длина 15, точность 2, неотрицательное.

Для каждого реквизита табличной части также установим свойство **Проверка заполнения** в значение **Выдавать ошибку**.

- На закладке **Нумерация**: убедимся, что свойство **Авто-нумерация** включено.

- В **дереве объектов конфигурации** выделим ветвь **Подсистемы**, вызовем ее контекстное меню и выберем пункт **Все подсистемы**. В открывшемся окне в списке **Подсистемы** выделим подсистему **УчетМатериалов**: включим видимость у команды **Приходная накладная: создать**, в группе **Панель действий.Создать**.

- **В режиме 1С:Предприятие**

Запустим **1С:Предприятие** в режиме отладки и протестируем получившийся документ. В открывшемся окне **1С:Предприятия** мы видим, что в панели навигации разделов **Бухгалтерия и Учет материалов** появилась команда **Приходные накладные** для открытия списка приходных накладных.

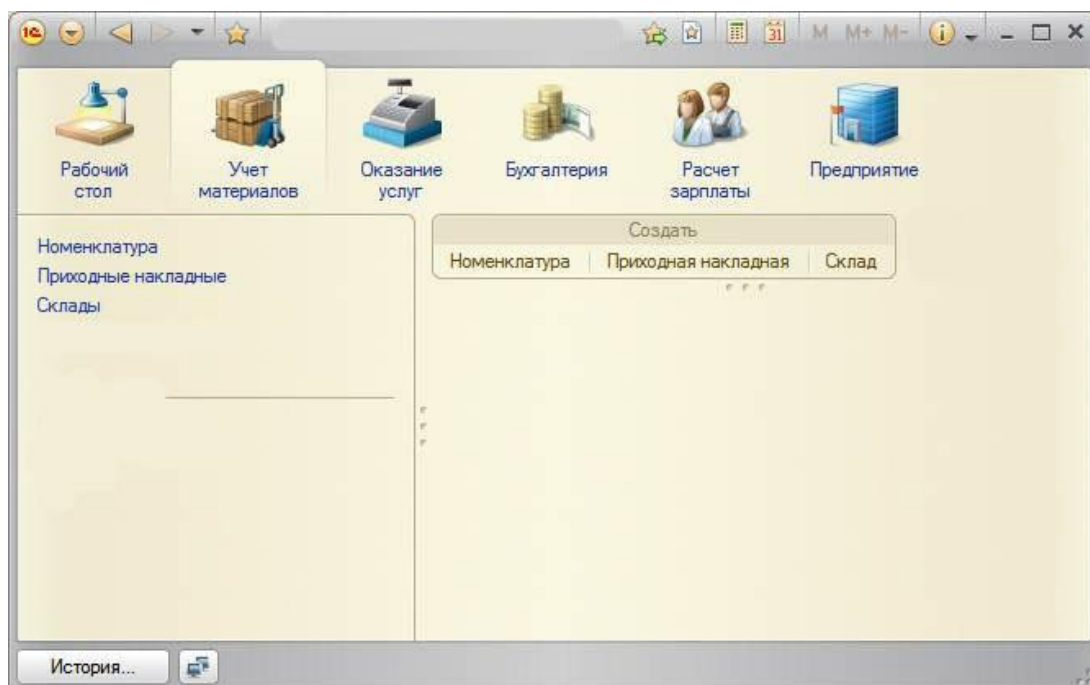


Рис. 4.3. В режиме 1С:Предприятие

Пока в БД нет ни одного документа **Приходная накладная**, поэтому выполним команду **Приходная накладная** в панели действий раздела **Учет материалов** и создадим новую приходную накладную.

Система автоматически подставит текущую дату создания документа и нулевое время, так как документ еще не проведен. В качестве времени документа при оперативном проведении ему присваивается оперативная отметка времени.

Поле **Номер** не заполнено, но система сама сгенерирует для нового документа уникальный номер, так как свойство **Автономерация** для документа включено по умолчанию. Новый номер будет сохранен в момент записи документа. *Осталось только заполнить табличную часть приходной накладной материалами (добавьте 3 любых записи в накладную). Так же система может вводить данные с клавиатуры и предлагать вам материалы по совпадающим буквам (создайте еще одну накладную таким образом)* После чего нажмем Провести и закрыть.

Документ будет сохранен и проведен, ему будет присвоен автоматически сгенерированный системой номер и текущее время проведения документа.

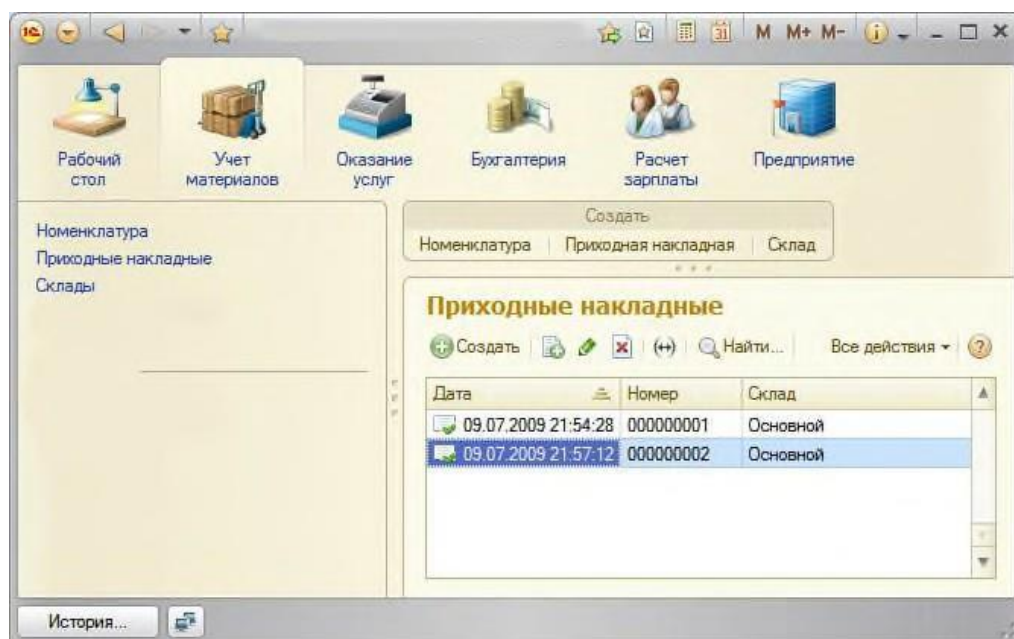


Рис. 4.4. Приходные накладные

Автоматический пересчет суммы в строках документа

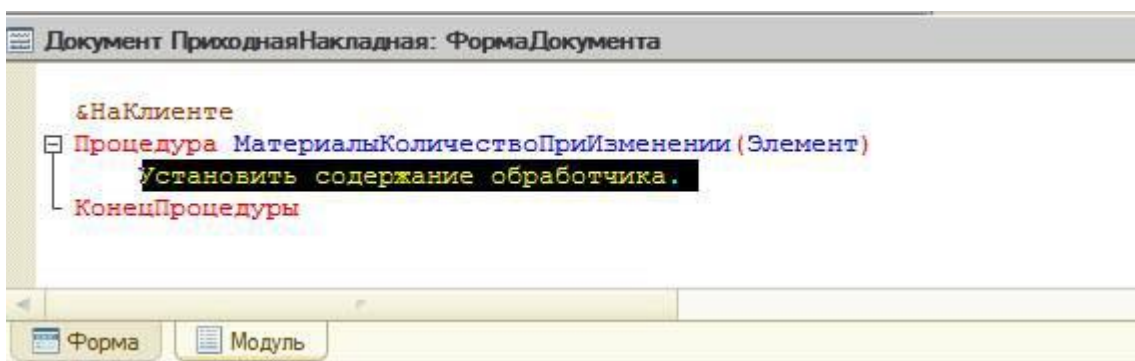
При заполнении документа приходится вводить сумму в каждой строке. Это неудобно, и возникает естественное желание автоматизировать работу документа так, чтобы сумма вычислялась автоматически каждый раз при изменении цены или количества материалов в строке.

Это совсем не сложно, и для этого нам потребуется сначала создать собственную форму документа, а затем воспользоваться возможностями встроенного языка.

- **В режиме конфигуратор:**

Откроем окно редактирования объекта конфигурации Документ **ПриходнаяНакладная**, далее пройдем на вкладку **Формы**: чтобы создать форму документа, нажмем кнопку открытия со значком лупы в поле ввода или кнопку **Добавить** над списком форм. Система вызовет полезный инструмент разработчика - **конструктор форм**, в нем выберем тип формы **Форма документа** и нажмем кнопку **Готово**. *Обращаем внимание, что в дереве объектов конфигурации у объекта конфигурации Документ ПриходнаяНакладная появилась форма ФормаДокумента, а на экране открылось окно редактора форм, содержащее эту форму.* Разработчик может через палитру свойств изменить свойства элемента, которые повлияют на его отображение в форме. Он может также изменить структуру элементов формы - создать новое поле, группу полей, добавить табличную часть, связав эти элементы с данными формы. Используя встроенный язык, разработчик может «вклиниться» в эти события и описать собственный алгоритм того, что должно происходить при наступлении этого события.

- Дважды щелкнем на элементе формы **МатериалыКоличество** правой кнопкой мыши откроем для него палитру свойств: прокрутив список до конца, мы увидим перечень событий, которые могут быть связаны с этим полем. Найдем в списке событий событие **ПриИзменении**, и нажмем на кнопку открытия (значок лупы), система создаст шаблон процедуры обработчика этого события в модуле нашей формы и откроет закладку **Модуль** редактора формы.



Листинг 4.1. Форма документа

Модуль - это «хранилище» для текста программы на встроенном языке. В модуль формы, в процедуру **МатериалыКоличествоПриИзменении()**, добавим следующий текст:

СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;

СтрокаТабличнойЧасти.Сумма =

СтрокаТабличнойЧасти.Количество*СтрокаТабличнойЧасти.Цена;

Таким образом, в результате выполнения первой строки процедуры обработчика переменная **СтрокаТабличнойЧасти** будет содержать объект **ДанныеФормыСтруктура**. Этот объект содержит данные, находящиеся в текущей строке табличной части документа

(**Элементы.Материалы.ТекущиеДанные**). Получив этот объект, мы можем обратиться к данным конкретной колонки табличной части, указав имя колонки в качестве свойства объекта.

Одна процедура для обработки нескольких событий

Теперь хотелось бы и для поля **Цена** и **Сумма** сделать то же самое.

Поэтому лучше будет поместить расчет суммы в некоторое «общедоступное» место, чтобы разные документы, имеющие аналогичные реквизиты табличной части, могли использовать этот алгоритм. Для описания таких «общедоступных» мест служат объекты конфигурации **Общий модуль**, расположенные в ветке **Общие > Общие модули**. Процедуры и функции, содержащиеся в этих модулях, могут быть доступны для любых объектов конфигурации.

- **В режиме конфигуратор:**

Добавим объект конфигурации Общий модуль. Для этого раскроем ветвь Общие в дереве объектов конфигурации, нажав на + слева от нее. Затем выделим ветвь **Общие модули** и нажмем кнопку **Добавить** в командной панели окна конфигурации:

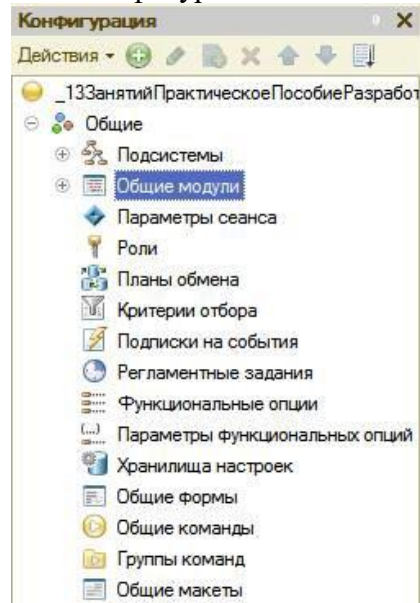
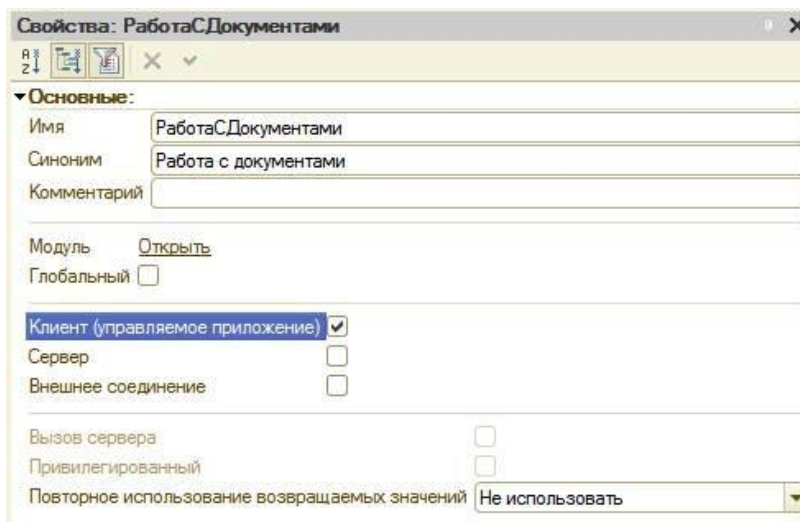


Рис. 4.5. Общие модули

Назовем его РаботаСДокументами и установим в его свойствах флажок Клиент (управляемое приложение), а флажок Сервер снимем. Это означает, что экземпляры этого модуля будут скомпилированы в контексте тонкого клиента и в контексте веб-клиента

Рис. 4.6. Свойства: РаботаСДокументами

Внесем в общий модуль следующий текст:



**Процедура РассчитатьСумму(строкаТабличнойЧасти) Экспорт
СтрокаТабличнойЧасти.Сумма =
СтрокаТабличнойЧасти.Количество*СтрокаТабличнойЧасти.Цена;
КонецПроцедуры**

Изменим текст обработчика. Для этого вернитесь в модуль формы документа, выберите поле КоличествоПриИзменении.

&НаКлиенте

**Процедура МатериалыКоличествоПриИзменении (Элемент)
СтрокаТабличнойЧасти. = Элементы.Материалы.ТекущиеДанные;
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
КонецПроцедуры**

Создадим обработчик события ПриИзменении для поля табличной части МатериалыЦена так же, как мы делали это для поля Материалы Количество, и повторим в нем вызов процедуры РассчитатьСумму из общего модуля.

- Листинг процедуры «МатериалыЦенаПриИзменении()»

&НаКлиенте

Процедура МатериалыЦенаПриИзменении (Элемент)

СтрокаТабличнойЧасти. = Элементы.Материалы.ТекущиеДанные;

РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);

КонецПроцедуры

- **В режиме 1С: Предприятие:**

Запустим 1С: Предприятие в режиме отладки и убедимся, что теперь сумма в строках табличной части документов ПриходнаяНакладная пересчитывается как при изменении количества, так и при изменении цены.

Документ «Оказание услуги»

- **В режиме конфигуратор:**

Добавим новый объект конфигурации Документ и назовем его ОказаниеУслуги.

- На закладке *Основные*: Представление объекта задавать не будем, вместо него будет использоваться Синоним объекта. Представление списка зададим как Оказание услуг.

- На закладке *Подсистемы*: отметим, что документ будет доступен в подсистемах Оказание услуг и Бухгалтерия.

- На закладке *Данные*:

создадим реквизиты документа:

- Склад, тип СправочникСсылка. Склады. Выберем для свойства Значения заполнения предопределенный элемент Основ

ка заполнения значение Выдавать ошибку.

- Мастер, тип СправочникСсылка.Сотрудники. Установим свойство Проверка заполнения значение Выдавать ошибку.

Создадим табличную часть этого документа ПереченьНоменклатуры с реквизитами:

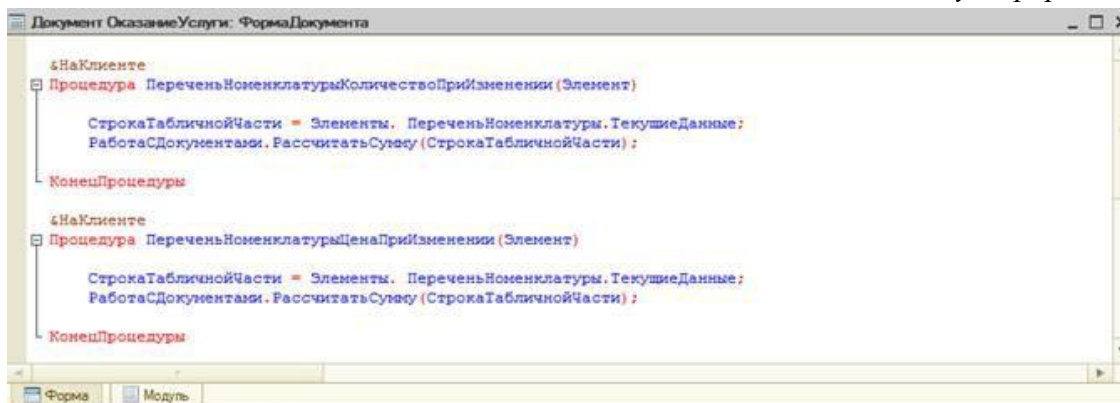
- Номенклатура, тип СправочникСсылка.Номенклатура;
- Количество, тип Число, длина 15, точность 3, неотрицательное;
- Цена, тип Число, длина 15, точность 2, неотрицательное;
- Сумма, тип Число, длина 15, точность 2, неотрицательное;

Установим для табличной части в целом и для каждого ее реквизита свойство Проверка заполнения значение Выдавать ошибку.

- На закладке *Формы*: создадим основную форму документа: Для поля ПереченьНоменклатурыКоличество создадим обработчик события ПриИзменении, в котором будем вызывать процедуру РассчитатьСумму из общего модуля РаботаСДокументами. При этом откроется модуль формы с шаблоном обработчика события ПереченьНоменклатурыКоличествоПриИзменении, который мы пока заполнять не будем, а перейдем в окно элементов формы на закладку *Форма* и аналогичным образом создадим обработчик события ПереченьНоменклатурыЦенаПриИзменении для поля ПереченьНоменклатурыЦена.

Далее модуль формы документа ОказаниеУслуги нужно заполнить следующим образом:

Листинг 4.2. Модуль формы документа



«ОказаниеУслуги»

- вной справочника Склады.

В заключение отредактируем командный интерфейс, чтобы в подсистеме Оказание услуг была доступна команда создания новых документов.

Для разнообразия воспользуемся другим способом.

Откроем окно редактирования объекта конфигурации

Подсистема ОказаниеУслуг и нажмем кнопку Командный интерфейс.

В открывшемся окне отразятся все команды выбранной подсистемы.

В группе Панель действий.Создать включим видимость у команды Оказание услуги: создать

- Клиент, тип СправочникСсылка.Клиенты. Установим свойство Провер

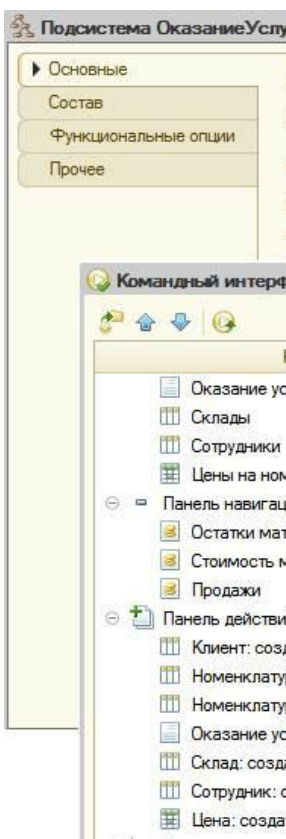


Рис. 4.7. Видимость команды оказания услуги: создать

В режиме 1С:Предприятие:

В панели действий раздела Оказание услуг вызовем команду создания документа Оказание услуги и выполним его следующим образом:

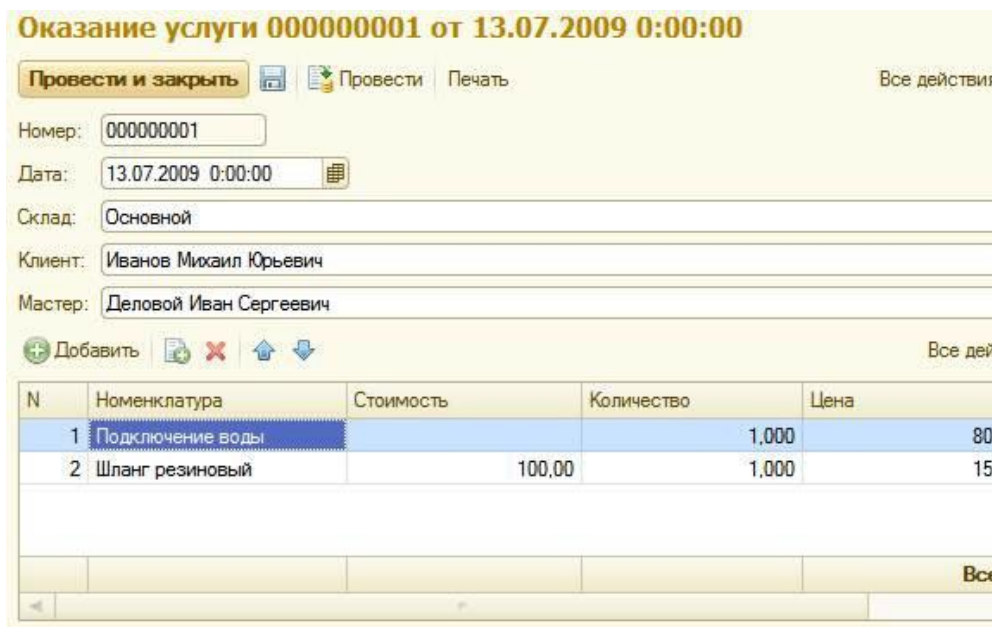


Рис. 4.8. Оказание услуги

Обратите внимание, что склад Основной подставляется по умолчанию, а для полей Мастер и Клиент выполняется проверка заполнения. А также при вводе цены и количества в табличную часть документа Оказание услуги сумма пересчитывается по нашему алгоритму.

Контрольные вопросы:

- Какими характерными особенностями обладает документ.
- Для чего предназначены реквизиты и табличные части документа.
- Какие существуют основные формы документа.
- Что такое проведение документа.
- Как создать объект конфигурации Документ и описать его основную структуру.
- Как создать новый документ и заполнить его данными.
- Как создать собственную форму документа.
- Что такое конструктор форм.
- Что такое редактор форм.
- Что такое элементы формы.
- Что такое события и с чем они связаны.
- Что такое обработчик события и как его создать.
- Что такое модуль и для чего он нужен.
- Зачем нужны общие модули.
- Что такое типоброзирующие объекты.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Лабораторная работа № 5

Чтобы не усложнять восприятие все теоретические моменты были вынесены в это отдельное занятие. На нем будут подробно рассмотрены вопросы, которые мы опускали или бегло проходили ранее. Эта теория очень важна для более глубокого понимания, почему мы делали так, а не иначе. Она необходима читателям, желающим в будущем самостоятельно разрабатывать конфигурации, а не просто шаг за шагом следовать данному пособию.

Механизм основных форм

На предыдущем занятии мы создали форму документа Приходная накладная и назначили эту форму основной. Что это значит?

У всех прикладных объектов конфигурации существует некоторое количество основных форм. Они служат для отображения данных объекта в том или ином виде.

Если разработчик не назначит в качестве основных форм объекта свои собственные, система будет генерировать необходимые формы объекта самостоятельно, в те моменты, когда к ним происходит обращение.

Наличие такого механизма позволяет разработчику не тратить время на создание форм для тестирования своей разработки, а воспользоваться тем, что платформа создаст по умолчанию.

Создание этих форм происходит динамически, в процессе работы системы. Форма создается в тот момент, когда к ней происходит обращение. Причем не важно, интерактивное это обращение или программное.

Также примечательным фактом является то, что состав основных форм, определенных для объекта конфигурации, может не совпадать с перечнем тех форм, которые вообще можно создать для данного объекта, используя конструктор формы.

Например, для большинства регистров в конфигураторе можно задать основную форму списка.

Однако если открыть конструктор форм для регистра, вы увидите, что кроме формы списка предлагается создать и форму набора записей регистра, которая отсутствует в перечне основных форм.

Обработчики событий

При работе с событиями в платформе 1С:Предприятие 8 следует различать два типа событий: события, связанные с формой и ее элементами, и все остальные.

Разница заключается в том, что обработчики событий, связанных с формой и ее элементами, - назначаемые, а обработчики всех остальных событий - фиксированные.

Фиксированный обработчик события должен иметь имя, совпадающее с именем события. Только в этом случае он будет вызываться при возникновении соответствующего события.

Назначаемый обработчик может иметь произвольное имя. Если имя процедуры совпадает с именем события формы или ее элемента, этого совсем недостаточно для вызова процедуры обработки события с таким именем. Требуется явное назначение процедуры обработчиком этого события в палитре свойств, в соответствующем событии.

Назначение обработчика может выполняться интерактивно, при работе с формой в конфигураторе, или программно, используя методы формы и ее элементов - УстановитьДействие().

Модули

Виды модулей:

- *Модуль управляемого приложения.* Модуль управляемого приложения выполняется при старте системы 1С:Предприятие в режимах тонкого клиента и веб-клиента.

В нем возможно объявление переменных, а также объявление и описание процедур и функций, которые будут доступны в любом модуле конфигурации (кроме модуля внешнего соединения). Их доступность также обеспечивается для неглобальных общих модулей с установленным свойством Клиент (управляемое приложение). В контексте модуля управляемого приложения доступны экспортируемые процедуры и функции общих модулей. Чтобы открыть модуль управляемого приложения, нужно выделить корень дерева объектов конфигурации и вызвать из контекстного меню команду Открыть модуль управляемого приложения

- *Общие модули.* В общих модулях хранятся процедуры и функции, которые вызываются из других модулей системы. Сам по себе общий модуль не исполняется. Исполняются отдельные его процедуры/ функции в момент их вызова из других модулей. Чтобы открыть общий модуль, нужно раскрыть ветвь Общие в дереве объектов конфигурации, затем раскрыть ветвь Общие модули и дважды щелкнуть мышью на нужном модуле.
- *Модули объектов.* Модули объектов - это, например, модуль элемента справочника или модуль документа. Эти модули вызываются тогда, когда либо программно создается этот объект средствами встроенного языка, например, методами СоздатьЭлемент() у менеджеров справочников или СоздатьДокумент() менеджеров документов, либо когда пользователь создает новый элемент справочника или документ интерактивно. Чтобы открыть модуль объекта, нужно в окне редактирования объекта конфигурации перейти на закладку Прочее и нажать кнопку Модуль объекта. Или, выделив нужный объект в дереве объектов конфигурации, вызвать из контекстного меню команду Открыть модуль объекта.

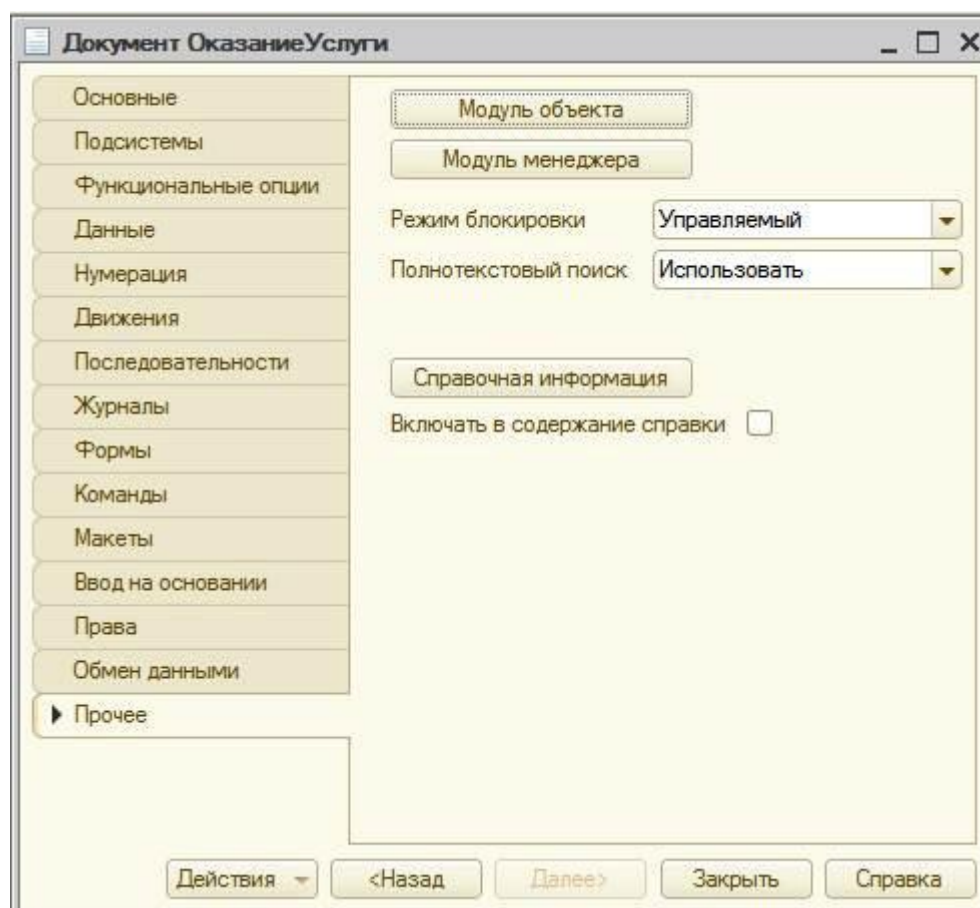
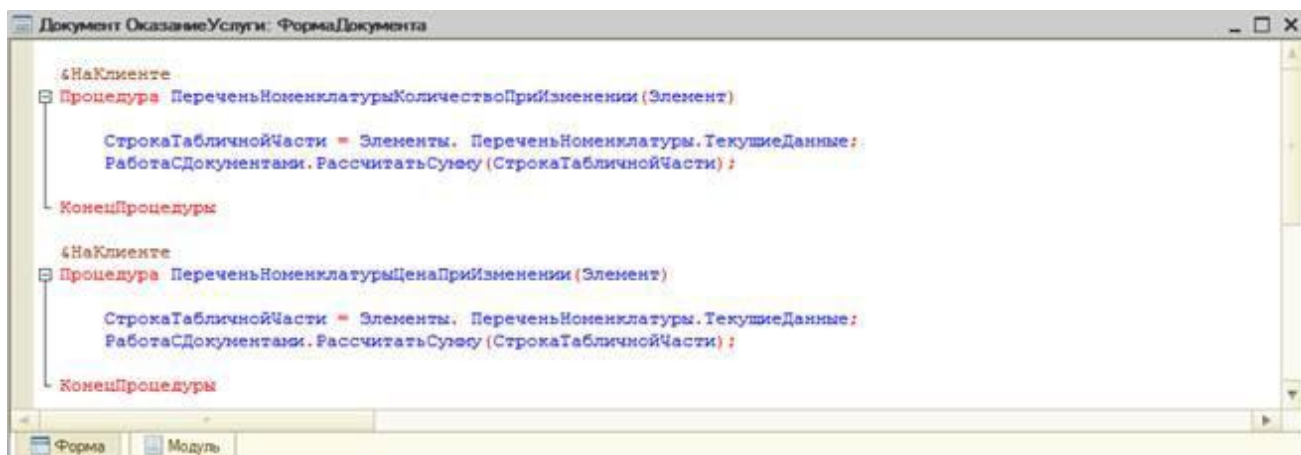


Рис.5.1. Открытие модуля объекта

- *Модули форм.* Каждая форма, определенная в конфигурации, имеет свой собственный модуль. Этот модуль исполняется при создании объекта УправляемаяФорма встроенного языка. Этот объект создается системой в режиме 1С предприятие в тот момент, когда мы программно (методами ПолучитьФорму() или ОткрытьФорму()). Чтобы открыть модуль формы, нужно открыть нужный объект конфигурации Форма объекта и в окне редактора форм перейти на

закладку Модуль .



Листинг 5.1. Открытие модуля формы

- **Модуль сеанса.** Модулем сеанса называется модуль, который автоматически выполняется при старте системы 1С:Предприятие в момент загрузки конфигурации. Модуль сеанса предназначен для инициализации параметров сеанса и обработки действий, связанных с сеансом работы. Модуль сеанса не содержит экспортируемых процедур и функций и может использовать процедуры из общих модулей конфигурации. Чтобы открыть модуль сеанса, нужно выделить корень дерева объектов конфигурации (строку Фамилия) и вызвать из контекстного меню команду Открыть модуль сеанса.
- **Модуль внешнего соединения** предназначен для размещения в нем текстов функций и процедур, которые могут вызываться в сессии внешнего соединения. Чтобы открыть модуль сеанса, нужно выделить корень дерева объектов конфигурации (строку ПособиеДляНачинающих) и вызвать из контекстного меню команду Открыть модуль внешнего соединения.
- **Модуль менеджеров.** Для каждого прикладного объекта существует менеджер, предназначенный для управления этим объектом как объектом конфигурации. С помощью менеджера можно создавать объекты, работать с формами и макетами. Модуль менеджера позволяет расширить функциональность менеджеров, предоставляемых системой, за счет написания процедур и функций на встроенном языке. Чтобы открыть модуль менеджера, нужно в окне редактирования объекта конфигурации перейти на закладку Прочее и нажать кнопку Модуль менеджера.
- **Модуль команды.** Как в самой конфигурации, так и у многих прикладных объектов могут существовать подчиненные объекты конфигурации - Команды. У каждой команды существует модуль команды, в котором можно написать предопределенную процедуру ОбработкаКоманды() для выполнения этой команды. Чтобы открыть модуль команды, подчиненной некоторому объекту конфигурации, нужно в окне редактирования объекта конфигурации перейти на закладку Команды и дважды щелкнуть мышью на нужной команде. Или, выделив нужную команду в дереве объектов конфигурации, вызвать из контекстного меню команду Открыть модуль команды.

Контекст модуля формы

Каждый модуль связан с остальной частью конфигурации. Эта связь называется контекстом модуля. Контекст модуля определяет набор доступных во время выполнения модуля объектов, переменных, процедур и функций.

Контекст модуля формы образуется:

- локальным контекстом самого модуля формы;
- реквизитами формы, которой ;принадлежит; модуль;
- свойствами и методами объекта УправляемаяФорма встроенного языка;
- свойствами и методами расширения формы, определяемого типом того объекта, данные которого содержатся в основном реквизите формы;

- глобальным контекстом, в том числе неглобальными общими модулями и экспортируемыми функциями и процедурами глобальных общих модулей;
- экспортируемыми переменными, процедурами и функциями модуля управляемого приложения;

Форма как программный объект

Помимо того что форма;внутри; своего модуля предоставляет доступ к различным частям конфигурации, она также доступна из других частей конфигурации как программный объект. При этом помимо стандартных свойств и методов объекта встроенного языка УправляемаяФорма, у нее могут существовать и другие свойства и методы, определенные разработчиком.

```

&НаКлиенте
□ Процедура МатериалыКоличествоПриИзменении (Элемент)
    СтрокаТабличнойЧасти = Элементы.Материалы.ТекущиеДанные;
    РаботаСДокументами.РассчитатьСумму (СтрокаТабличнойЧасти);
КонецПроцедуры
  
```

Листинг 5.2.

Например, если в модуле формы ФормаДокумента документа ПриходнаяНакладная описана экспортируемая процедура, то может быть использован следующий вызов этой процедуры:

```

форма=ПолучитФорму ("Документ.ПриходнаяНакладная.форма.формаДокумента");
форма.МатериалыКоличествоПриИзменении ();
  
```

Листинг 5.3.

Как понять работу кода на встроенном языке

Теперь мы покажем два способа, как самому разобраться с множеством незнакомых свойств и методов объектов конфигурации, чтобы в будущем самостоятельно изучать фрагменты кода или создавать свои собственные процедуры на встроенном языке.

Синтакс-помощник — инструмент, созданный для помощи разработчику, содержащий описание всех программных объектов, которые использует система, их методов, свойств, событий и пр.

Чтобы открыть синтакс-помощник, нужно нажать соответствующую кнопку на панели инструментов конфигуратора или выполнить команду главного меню Справка > Синтакс-помощник, или с помощью горячей клавиши (Ctrl + F1).

Анализ кода с помощью синтакс-помощника:

Пользоваться синтакс-помощником удобно в тех случаях, когда нужно разобраться в уже написанном незнакомом коде.

Первый способ - найти нужный раздел в содержании и спускаться вниз «по дереву», раскрывая нужные подразделы, свойства, ссылки и т.п.

Второй способ - воспользоваться контекстной помощью синтакс-помощника. Для этого нужно открыть программный модуль, установить курсор на интересующую вас конструкцию встроенного языка и нажать Ctrl+ F1.

Есть также еще одна полезная возможность использования синтакс-помощника. Можно ограничить состав объектов, которые будут отображаться в нем. Так как мы находимся на клиенте, в форме, имеет смысл ограничиться только объектами встроенного языка, доступными в режимах Тонкий клиент и Веб-клиент. Для этого нужно выполнить команду главного меню Сервис > Параметры или нажать кнопку Открыть режим настройки параметров, находящуюся над окном описания объектов синтакс-помощника. На закладке Справка окна Параметры можно снять или поставить отметку у нужных режимов исполнения.

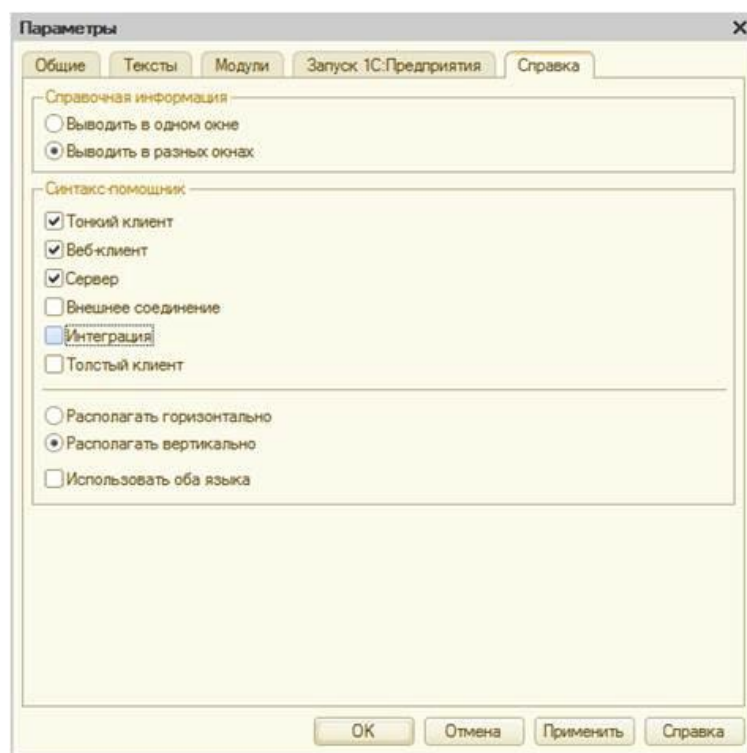


Рис. 5.2. Параметры

Анализ кода с помощью отладчика

Пользоваться отладчиком наиболее удобно в тех случаях, когда нужно написать какой-то собственный код. Можно просто остановиться в конкретном месте программы и посмотреть, какие же свойства здесь доступны или какие программные объекты здесь используются.

Отладчик - вспомогательный инструмент, облегчающий разработку и отладку программных модулей системы 1С:Предприятие. Отладчик предоставляет следующие возможности:

- пошаговое выполнение модуля,
- расстановка точек останова,
- прерывание и продолжение выполнения модуля,
- возможность отладки нескольких модулей одновременно,
- вычисление выражений для анализа состояния переменных,
- просмотр стека вызовов процедур и функций,
- возможность остановки по возникновению ошибки,
- возможность редактирования модуля в процессе отладки.

Объекты

Как правило, термин объект употребляется в одном из трех контекстов:

- конфигурация,
- база данных,
- встроенный язык.

Говоря о конфигурации, термином объект конфигурации мы обозначаем некоторую совокупность описания данных и алгоритмов работы с этими данными. Например, в конфигурации может существовать объект Справочник Сотрудники.

Когда мы говорим о базе данных, термином *объект* мы обозначаем всего лишь некий элемент такой информационной структуры. Характерной особенностью такого элемента является то, что на него (как на совокупность данных) существует ссылка, которая может являться значением какого-либо поля другой информационной структуры.

Если же мы начинаем говорить о встроенном языке и о том, каким образом средствами встроенного языка работать со справочниками, то термином *объект* мы обозначаем тип данных, позволяющий получить доступ к данным и обладающий набором свойств и методов.

Сервер и клиенты

Система 1С:Предприятие поддерживает два варианта работы системы: файловый и клиент-серверный.

Файловый вариант работы с информационной базой рассчитан на персональную работу одного пользователя или работу небольшого количества пользователей в локальной сети. В этом варианте все данные информационной базы (конфигурация, база данных, административная информация) располагаются в одном файле.

Рис. 5.3. Файловый вариант

Основное назначение файлового варианта - быстро и легко установить систему и работать с ней. Например, чтобы что-то посмотреть или доработать дома или на ноутбуке. В файловом варианте тоже можно вести реальную учетную работу, но при этом нужно понимать, что он не предоставляет абсолютно всех тех же возможностей по масштабируемости, защите данных, какие имеет клиент-серверный вариант.

Клиент-серверный вариант предназначен для использования в рабочих группах или в масштабе предприятия - это основной вариант для работы в многопользовательской среде с большим объемом данных. Он предоставляет абсолютно все возможности по масштабируемости, администрированию и защите данных. Однако он требует значительных усилий по установке и администрированию.

Система 1С:Предприятие изначально рассчитана на клиент-серверный вариант работы. Хотя сейчас вы разрабатываете свою учебную конфигурацию в файловом варианте работы, она будет работать и в клиент-серверном варианте без ваших дополнительных усилий.

Прикладные решения разрабатываются один раз и одинаково работают, что в одном, что в другом варианте. То есть переход с одного варианта на другой не требует переделки конфигурации.

Рис. 5.4. Клиент-серверный вариант



Клиент-серверная архитектура разделяет всю работающую систему на три различные части, определенным образом взаимодействующие между собой, - Клиент, Сервер 1С: Предприятия и Сервер баз данных.

Клиентское приложение - это программа, часть системы 1С: Предприятие. Основное ее назначение - организация пользовательского интерфейса, отображение данных с возможностью их изменения. Кроме этого, клиентское приложение может исполнять код на встроенном языке. Такой подход позволяет клиентскому приложению быть очень «легким», не требовать много ресурсов, «путешествовать» по Интернету и работать даже в среде веб-браузеров.

Сервер баз данных - это тоже программа. Она уже не является частью системы 1С:Предприятие, это специализированная программа, поставляемая сторонними производителями. Ее основное назначение - это организация и ведение баз данных - структурированных организованных наборов данных, описывающих характеристики каких-либо физических или виртуальных систем.

В настоящее время система 1С:Предприятие может работать со следующими серверами баз данных:

- *MS SQL Server;*
- *Posteg SQL;*
- *IBM DB2;*
- *Oracle Database.*

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Лабораторная работа № 6 Регистры накопления.

Зачем нужен регистр накопления

Казалось бы, все необходимое мы с вами уже создали: у нас есть что расходовать и приходовать (справочники), и у нас есть чем расходовать и приходовать (документы). Осталось только построить несколько отчетов, и автоматизация предприятия будет закончена. Однако это не так.

Во-первых, путем анализа документов можно, конечно, получить требуемые нам выходные данные. Но представьте, что завтра предприятие решит немного изменить свои бизнес-планы, и нам потребуется ввести в конфигурацию еще один документ.

Во-вторых, отчеты, анализирующие документы, будут работать довольно медленно, что будет вызывать раздражение пользователей и недовольство руководителей.

Поэтому в системе 1С Предприятие есть несколько объектов конфигурации, которые позволяют создавать в базе данных структуры, предназначенные для накопления информации в удобном для последующего анализа виде. Использование таких «хранилищ» данных позволяет нам, с одной стороны, накапливать в них данные, а с другой стороны, легко создавать нужные нам отчеты или использовать эти данные в алгоритмах работы конфигурации.

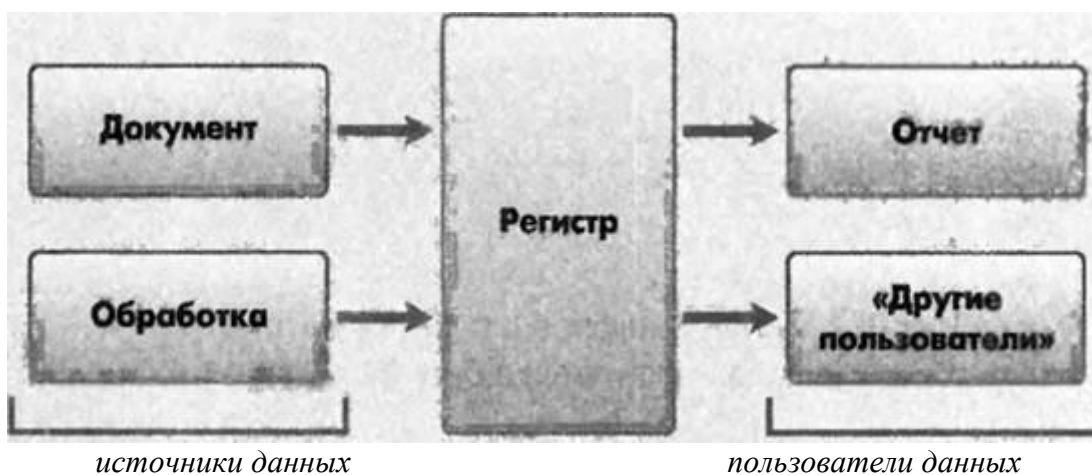


Рис. 6.1. Алгоритм работы конфигурации

В конфигурации существует несколько объектов, называемых регистрами, для описания подобных «хранилищ».

Что такое регистр накопления

Объект конфигурации Регистр накопления предназначен для описания структуры накопления данных. На основе объекта конфигурации Регистр накопления платформа создает в базе данных таблицы, в которых будут накапливаться данные, поставляемые различными объектами базы данных. Эти данные будут храниться в таблицах в виде отдельных записей, каждая из которых имеет одинаковую, заданную в конфигураторе структуру.

Отличительной особенностью регистра накопления является то, что он не предназначен для интерактивного редактирования пользователем.

Основным назначением регистра накопления является накопление числовой информации в разрезе нескольких измерений, которые описываются разработчиком в соответствующем объекте конфигурации Регистр накопления и являются подчиненными объектами конфигурации.

Виды числовой информации, накапливаемой регистром накопления, называются ресурсами, также

являются подчиненными объектами и описываются в конфигураторе.

Изменение состояния регистра накопления происходит, как правило, при проведении документа и заключается в том, что в регистр добавляется некоторое количество записей. Каждая запись содержит значения измерений, значения приращений ресурсов, ссылку на документ, который вызвал эти

изменения (регистратор), и направление приращения (приход или расход). Такой набор записей называется движениями регистра накопления. Каждому движению регистра накопления всегда должен соответствовать регистратор - объект информационной базы (как правило, документ), который произвел эти движения.

Кроме этого, регистр накопления может хранить дополнительную информацию, описывающую каждое движение. Набор такой дополнительной информации задается разработчиком при помощи реквизитов объекта конфигурации Регистр накопления.

Движения документа

Движения документа — это записи в регистрах, которые создаются в процессе проведения документа и отражают изменения, производимые документом.

Способы работы с коллекцией

В процессе формирования движений документов, когда в цикле обходили табличные части документов Приходная Накладная и Оказание Услуги, вы столкнетесь с одним из объектов встроенного языка, который является коллекцией.

Многие объекты встроенного языка являются коллекциями. Коллекция представляет собой совокупность объектов. Существуют общие принципы работы с любой коллекцией.

Во-первых, доступ к каждому объекту коллекции возможен путем перебора элементов коллекции в цикле. Для этого используется конструкция языка Для Каждого Из... Цикл ...

Листинг 6.1: Перебор элементов коллекции в цикле

```
Для Каждого СтрокаТабличнойЧасти из ТабличнаяЧасть Цикл  
Сообщить(СтрокаТабличнойЧасти.Услуга);  
КонецЦикла;
```

В этом примере ТабличнаяЧасть - это коллекция строк табличной части объекта конфигурации. При каждом проходе цикла в переменной СтрокаТабличнойЧасти будет содержаться очередная строка из этой коллекции.

Во-вторых, существует доступ напрямую к элементу коллекции, без перебора коллекции в цикле. Здесь возможны различные комбинации двух обращений.

1. Во встроенном языке бывают именованные коллекции. То есть коллекции, в которых каждый элемент имеет некоторое уникальное имя. В этом случае обращение к элементу коллекции возможно по этому имени.

Листинг 6.2: Обращение к элементу коллекции

```
Справочники.Сотрудники;  
Справочники["Сотрудники"];
```

2. Если нет смысла в «персонификации» элементов коллекции (коллекция неименованная), тогда обращение к элементу коллекции возможно по индексу (индекс первого элемента коллекции - ноль).

Листинг 6.3: Обращение к элементу коллекции по индексу

ТабличнаяЧасть[0];

В этом примере *ТабличнаяЧасть* - это коллекция строк табличной части объекта конфигурации. И мы обращаемся к первому элементу этой коллекции, указывая его индекс - 0. Следует отметить, что существуют коллекции, сочетающие оба вида обращений. Например, к коллекции колонок таблицы значений можно обращаться как по именам колонок, так и по индексу.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Добавление регистра накопления

- **В режиме конфигуратор**

Откроем в конфигураторе нашу учебную конфигурацию и добавим новый объект конфигурации Регистр накопления. Для этого выделим в дереве объектов конфигурации ветвь Регистры накопления и нажмем кнопку **Добавить** в командной панели окна конфигурации.

- *На закладке Основные:* имя регистра - *ОстаткиМатериалов*. Расширенное представление списка как Движения по регистру *Остатки материалов*. Этот заголовок будет отображаться в окне списка записей регистра.

- *На закладке Подсистемы:* отметим в списке следующие подсистемы: *Учет материалов, Оказание услуг и Бухгалтерия*.

- *На закладке Данные:* нажмем на «+» и добавим следующие:

- *Материал*, тип *СправочникСсылка.Номенклатура*;
- *Склад*, тип *СправочникСсылка.Склады*.
- ресурс **Количество** с длиной 15 и точностью 3.

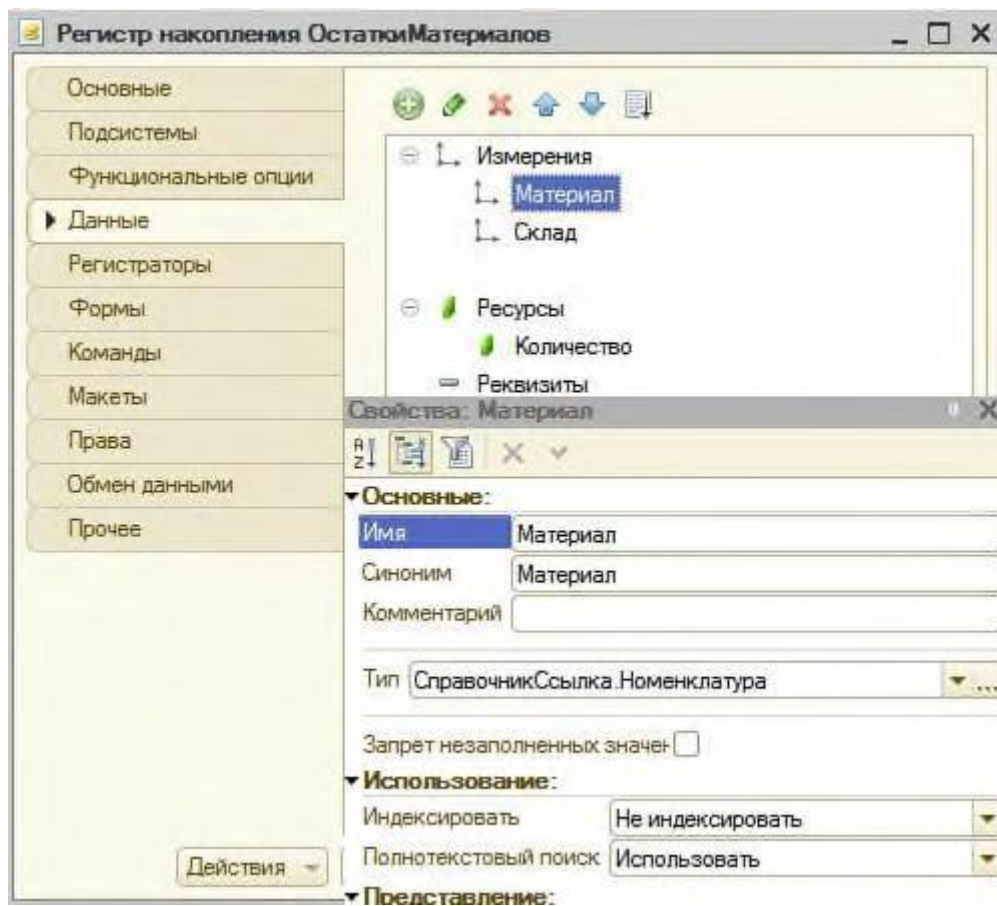


Рис. 6.2. Создание измерений регистра

В результате этих действий регистр **ОстаткиМатериалов** должен иметь следующий вид:

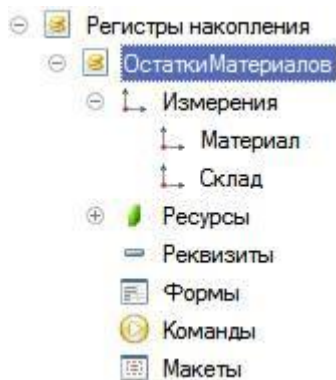


Рис. 6.3. ОстаткиМатериалов

Если вы сейчас попытаетесь запустить 1С : Предприятие в режиме отладки, то система выдаст сообщение об ошибке: «**РегистрНакопления.ОстаткиМатериалов: Ни один из документов не является регистратором для регистра**». Это сообщение еще раз подтверждает тот факт, что назначение регистра накопления в том, чтобы аккумулировать данные, поставляемые различными документами.

Поэтому мы сформируем движения регистра накопления **ОстаткиМатериалов** в процессе проведения двух созданных нами документов **ПриходнаяНакладная** и **ОказаниеУслуг**.

Откроем окно редактирования объекта конфигурации **Документ ПриходнаяНакладная**. Перейдем на закладку **Движения**, раскроем список **Регистры накопления** и отметим регистр накопления **ОстаткиМатериалов**.

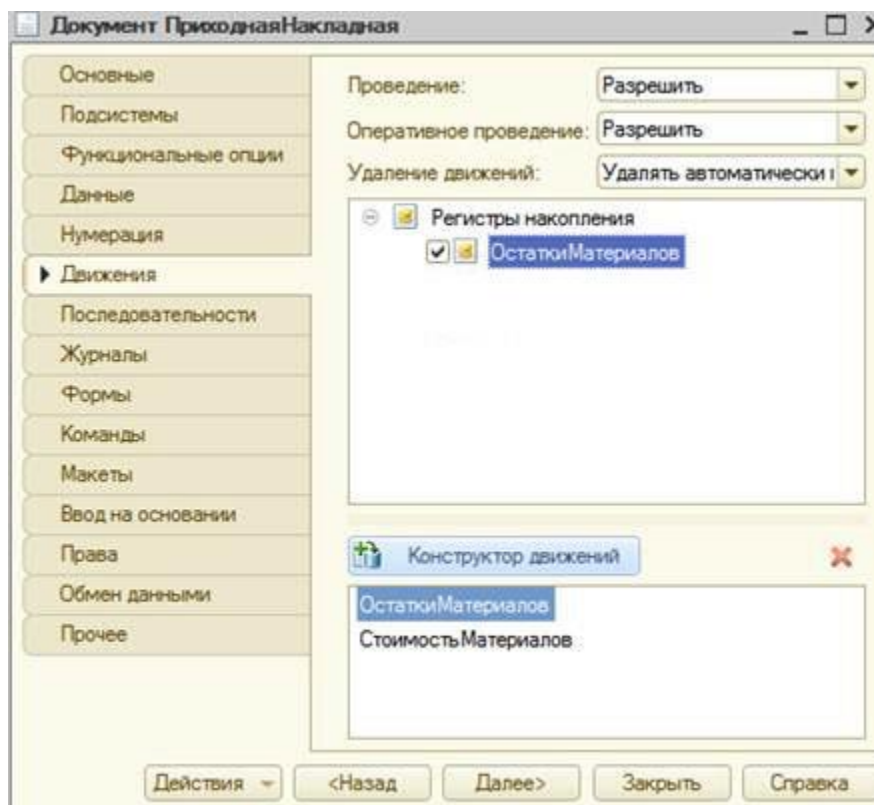


Рис. 6.4. Создание движения документа

После чего активизируется **Конструктор движений**, воспользуемся этим конструктором.

Конструктор устроен просто. В списке **Регистры** перечислены регистры, в которых документ может создавать движения. В нашем случае там пока один регистр **ОстаткиМатериалов**.

В списке **Реквизиты документа** должны находиться исходные данные для создания движений - реквизиты документа **ПриходнаяНакладная**. А в таблице **Поле - Выражение** должны быть заданы

формулы, по которым будут вычисляться значения измерений и ресурсов регистра при записи движений.

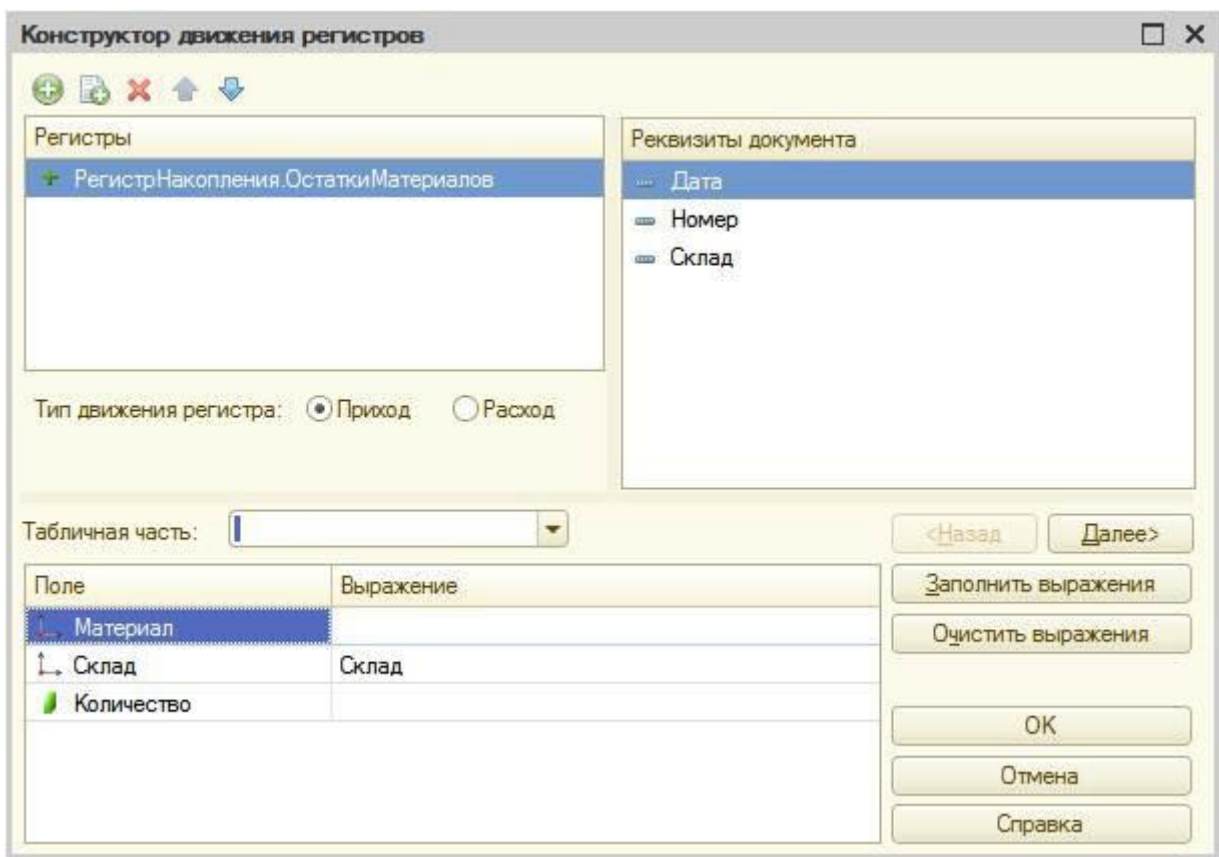


Рис. 6.5. Конструктор движения регистров

В поле выбора **Табличная часть** выберем табличную часть нашего документа - **Материалы**.

Список реквизитов документа, который уже заполнен реквизитами шапки документа, автоматически дополнится реквизитами нашей табличной части.

Теперь нажмем кнопку **Заполнить выражения**.

В нижнем окне сформируется соответствие полей (измерений и ресурсов) регистра и выражений для их расчета.

Нажмем кнопку ОК и посмотрим, какой текст сформировал конструктор в модуле документа

Приходная Накладная .

*Конструктор создал обработчик события **ОбработкаПроведения** объекта конфигурации **Документ Приходная Накладная**, поместил его в модуль объекта и открыл текст модуля.*

В заключение отредактируем командный интерфейс, чтобы в подсистемах **Бухгалтерия, Оказание услуг и Учет материалов** была доступна ссылка для просмотра записей нашего регистра накопления.

- В дереве объектов конфигурации выделим ветвь **Подсистемы**, вызовем ее контекстное меню и выберем пункт **Все подсистемы**:

- В списке **Подсистемы** выделим подсистему **УчетМатериалов**.
- В группе **Панель навигации.Обычное** включим видимость у команды **Остатки материалов** и мышью перетащим ее **См.также.** панели навигации

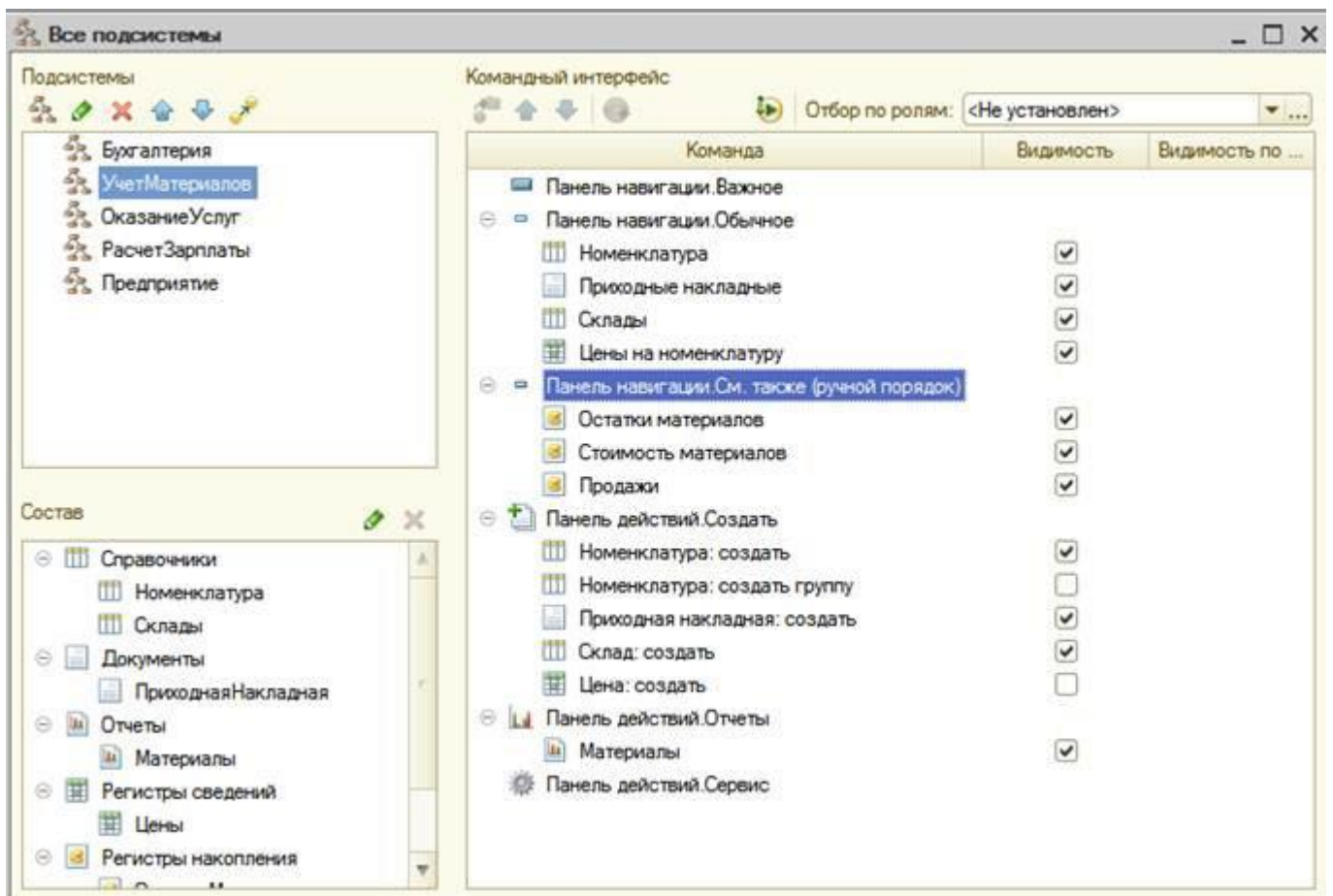


Рис. 6.6. Панель навигации См. также

- Аналогично, выделив подсистемы **ОказаниеУслуг** и **Бухгалтерия**, в панели навигации в группе **Обычное** включим видимость у команды **Остатки материалов** и перенесем ее в группу **См.также**. панели навигации
- В режиме 1С:Предприятие

В открывшемся окне 1С предприятия мы видим, что в панели навигации в группе **См.также** разделов **Бухгалтерия**, **Оказание услуг** и **Учет материалов** появилась команда для открытия списка регистра **Остатки материалов**.

Чтобы проследить связь между проведением документа и накоплением информации в регистре, откроем список приходных накладных, выполнив команду **Приходные накладные** разделе **Бухгалтерия**.

Откроем **Приходную накладную № 1** и нажмем **Провести и закрыть**, то есть перепроведем ее. То же самое сделаем для **Приходной накладной № 2**.

Перепровести документы можно и не открывая документов. Для этого нужно выделить нужный документ в списке, нажать кнопку **Все действия** в командной панели формы списка и выбрать пункт **Провести**.

Теперь выполним команду **Остатки материалов** и откроем список нашего регистра накопления:

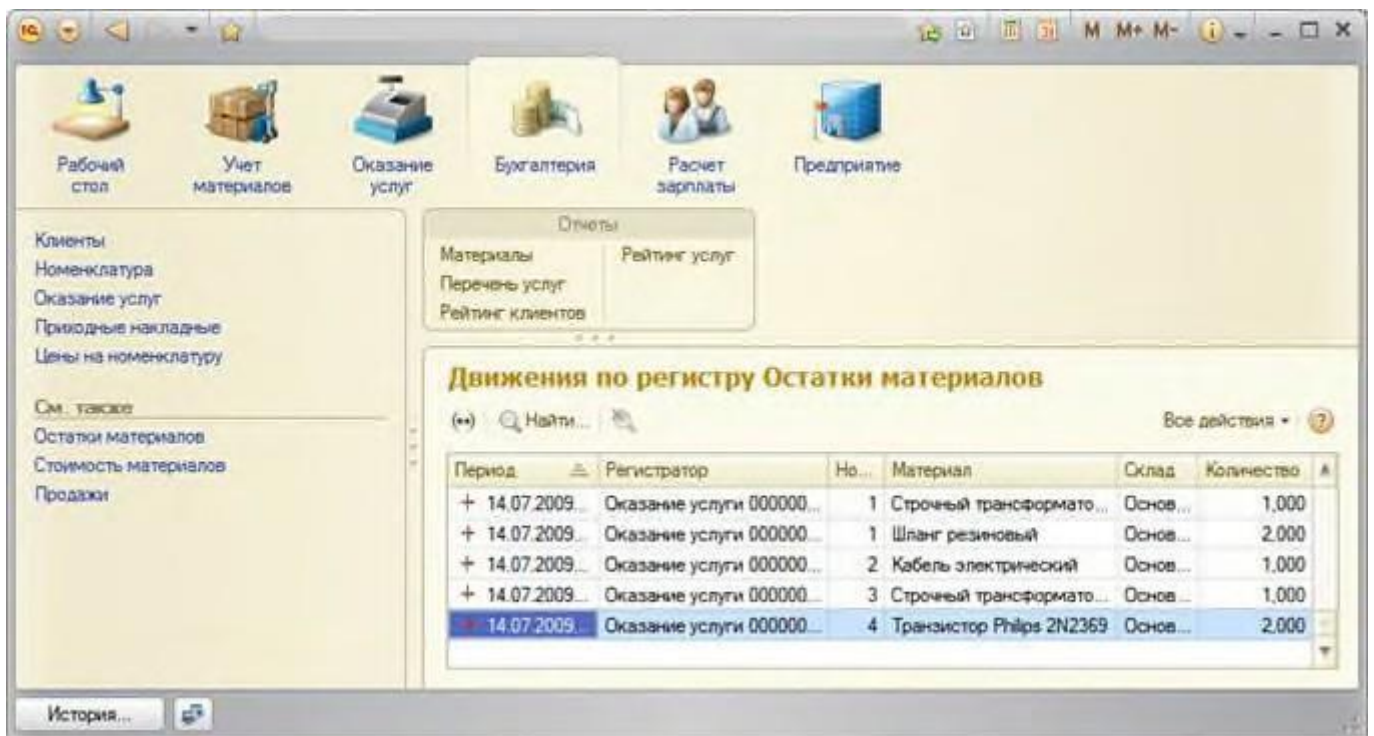


Рис. 6.7. Движения по регистру Остатки материалов

Команда перехода к движениям в форме документа

- В режиме Конфигуратор

При реальной работе записей в регистре **ОстаткиМатериалов** будет много, и будет трудно понять, какие записи относятся к определенному документу.

Поэтому наряду с общим списком регистра хотелось бы иметь возможность вызывать из формы документа список регистра, в котором показаны движения, произведенные только этим документом. Чтобы реализовать такую возможность, вернемся в конфигуратор и откроем форму документа **ПриходнаяНакладная**.

В левом верхнем окне перейдем на закладку **Командный интерфейс**: в разделе Панель навигации раскроем группу перейти, установим свойство **Видимость** для этой команды

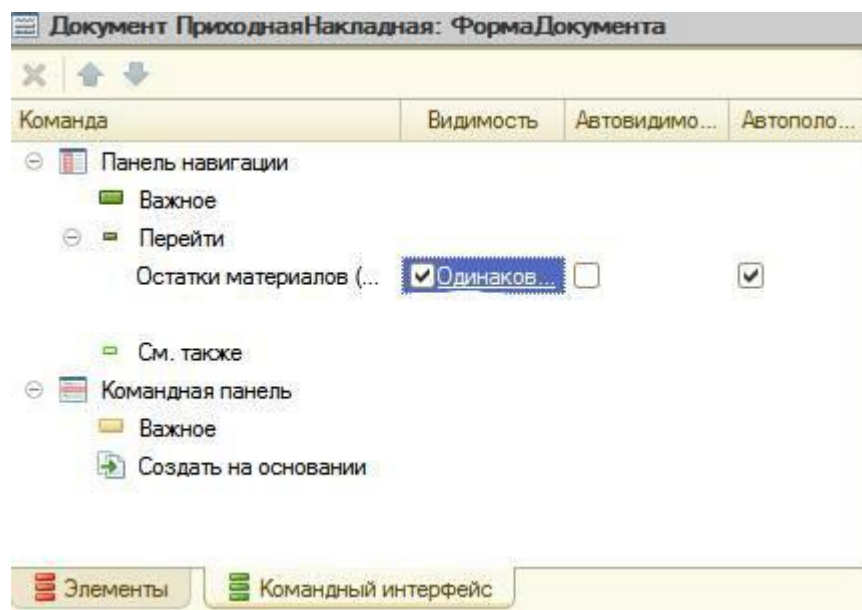


Рис. 6.8. Командный интерфейс

- **В режиме 1С:Предприятие**

Запустим 1С: Предприятие в режиме отладки и откроем **Приходную накладную № 2**.

В форме документа появилась панель навигации, в которой мы можем переходить к списку записей регистра **Остатки Материалов**, связанному с документом, и обратно к содержимому документа.

Движения документа «Оказание услуги»

- **В режиме Конфигуратор**

Откроем окно редактирования объекта конфигурации Документ **ОказаниеУслуги**.

Перейдем на закладку Движения и в списке регистров конфигурации отметим регистр накопления

ОстаткиМатериалов.

- Нажмем кнопку **Конструктор движений**:

1. Тип движения регистра - **Расход**

2. В поле выбора **Табличная часть** выберем табличную часть нашего документа -

ПереченьНоменклатуры.

3. Нажмем кнопку **Заполнить выражения**.

4. Выделим поле регистра **Материал** и в окне **Реквизиты документа** дважды щелкнуть по строке

ТекСтрокаПереченьНоменклатуры.Номенклатура., нажмем «ОК»

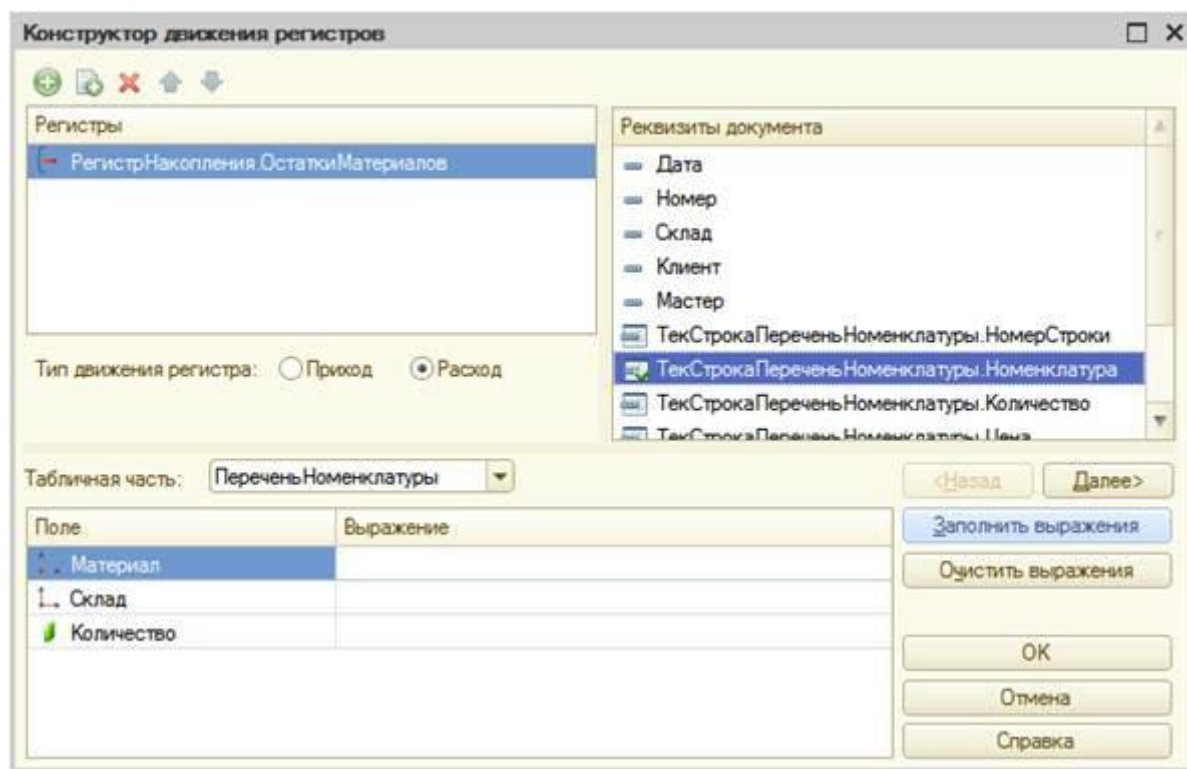


Рис. 6.9. Конструктор движения регистров

В заключение отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра *Остатки Материалов*, связанному с документом.

Для этого откроем форму документа **ОказаниеУслуги**.

В левом верхнем окне перейдем на закладку **Командный интерфейс**.

В разделе **Панель навигации** раскроем группу **Перейти** и установим видимость для команды открытия регистра накопления **Остатки материалов**.

- **В режиме 1С:Предприятие**

Запустим 1С:Предприятие в режиме отладки и в разделе **Оказание услуг** откроем документ **Оказание услуги № 1** и нажмем **Провести и закрыть**, то есть перепроведем его.

Теперь выполним команду **Остатки материалов** и откроем список нашего регистра накопления:

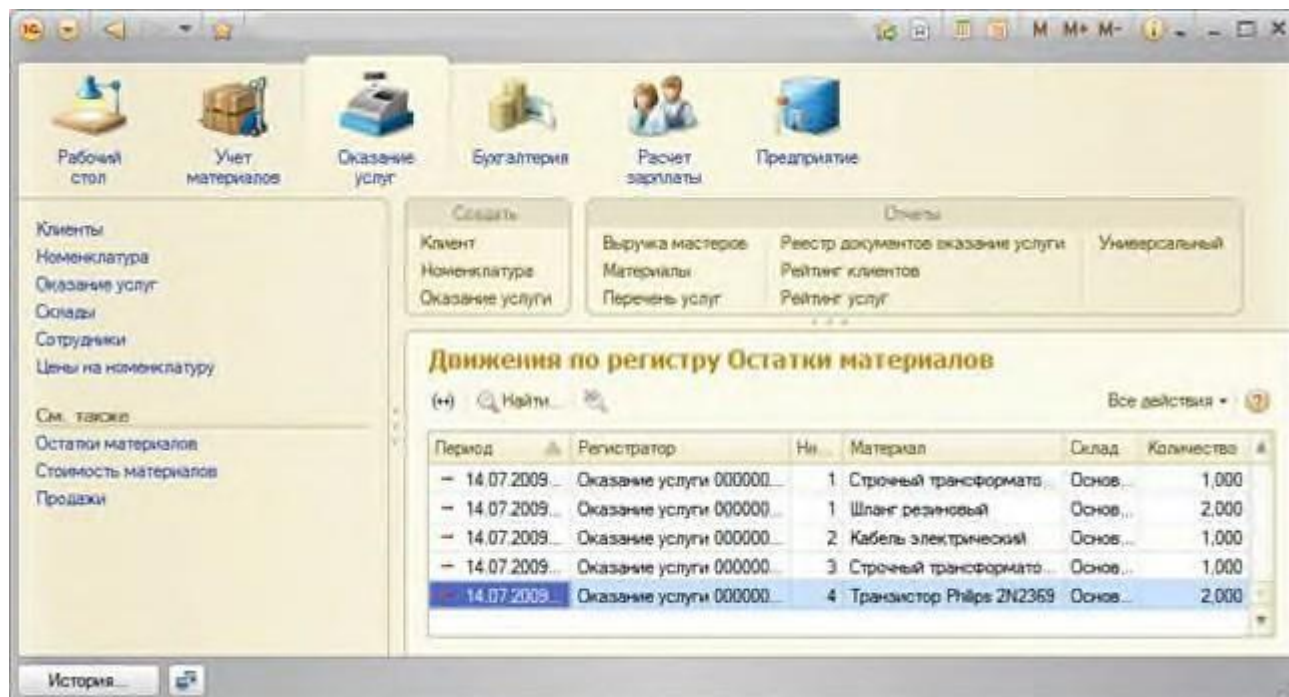


Рис. 6.10. Движения по регистру Остатки материалов

Мы видим, что в регистре накопления Остатки материалов появилась еще одна запись, что соответствует количеству строк в табличной части проведенного документа.

Все поля регистра заполнились данными документа так, как мы задали в обработчике проведения документа **Оказание услуги**.

Пиктограмма со знаком «минус» слева от записи указывает на тип движения – Расход.

Сейчас мы видим весь список движений регистра. Открыв этот список из формы документа, мы можем отфильтровать движения по документу-регистратору.

Для этого откроем еще раз документ **Оказание услуги № 1**.

В форме документа появилась панель навигации, в которой мы можем переходить к списку записей регистра Остатки материалов, связанному с документом, и обратно к содержимому документа.

Сформированные таким образом движения этого документа будут не совсем правильными.

Дело в том, что в документе **Оказание услуги**, в отличие от документа Приходная накладная, могут содержаться не только расходуемые материалы, но и услуги. Поэтому в регистр Остатки материалов будут попадать записи и о расходуемых услугах, что неправильно.

Пока мы ничего не будем делать с движениями, которые сформировал конструктор, но как только познакомимся с перечислениями, мы внесем в обработчик проведения необходимые изменения.

Контрольные вопросы

- Для чего предназначен объект конфигурации Регистр накопления.
- Почему следует использовать регистры, хотя необходимая информация содержится в других объектах.
- Для чего нужны измерения регистра, ресурсы и реквизиты.
- Что такое движения регистра и что такое регистратор.
- Как создать новый регистр накопления и описать его структуру.
- Как создать движения документа с помощью конструктора движений.
- Как средствами встроенного языка обойти табличную часть документа и обратиться к ее данным.
- Как показать команды открытия списка регистра в интерфейсе конфигурации и в интерфейсе формы.

Лабораторная работа № 7 Простые отчеты.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Что такое отчет

Объект конфигурации Отчет предназначен для описания алгоритмов, при помощи которых пользователь сможет получать необходимые ему выходные данные. Алгоритм формирования выходных данных описывается при помощи визуальных средств или с использованием встроенного языка. В реальной жизни объектам конфигурации Отчет соответствуют всевозможные таблицы выходных данных, сводных данных, диаграммы и пр.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Добавление отчета

- **В режиме Конфигуратор**

Приступим к созданию отчета, который будет показывать нам приход, расход и остатки материалов.

Откроем в конфигураторе нашу учебную конфигурацию и добавим новый объект конфигурации **Отчет**.

Для этого выделим в дереве объектов конфигурации ветвь Отчеты и нажмем кнопку **Добавить** в командной панели окна конфигурации.

- На закладке **Основные**: имя отчета - **Материалы**. Нажмем кнопку **Открыть схему компоновки данных** или кнопку открытия со значком лупы.

Макет

Так как у отчета, который мы создаем, еще не существует схемы компоновки данных, платформа предложит создать новую схему. Схема компоновки данных с точки зрения конфигурации является макетом, поэтому будет открыт конструктор макета, предлагающий выбрать единственный тип макета - **Схема компоновки данных** и нажмем кнопку **Готово**:

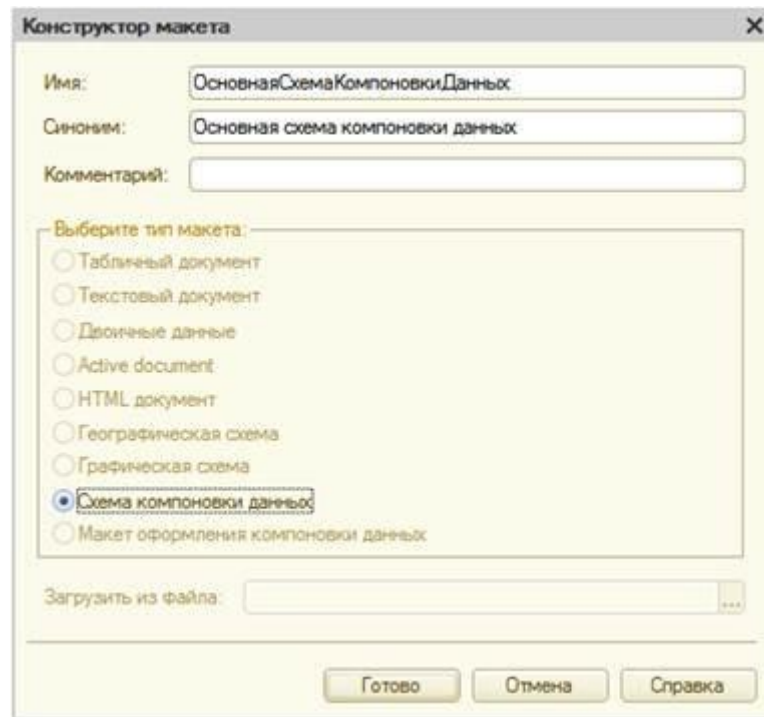


Рис. 7.1. Конструктор макета

Схема компоновки данных

Платформа создаст новый макет, содержащий схему компоновки данных, и сразу же откроет конструктор схемы компоновки данных.

- Добавим новый **набор данных - запрос**. Для этого нажмем кнопку **Добавить** и выберем соответствующий вариант.
- Добавим текст запроса, запустим конструктор запроса нажмем кнопку

Конструктор запроса.

Конструктор запроса - инструмент, созданный для помощи разработчику, позволяющий визуально конструировать запрос. Даже пользователь, не знакомый с языком запросов, может с помощью конструктора создать синтаксически правильный запрос.

В списке **База данных** представлены таблицы для создания запроса. На основе их данных мы имеем возможность построить отчет.

Если раскрыть ветку **РегистрыНакопления**, то мы увидим, что кроме таблицы регистра **ОстаткиМатериалов** в этой ветке присутствуют еще несколько виртуальных таблиц, которые формирует система.

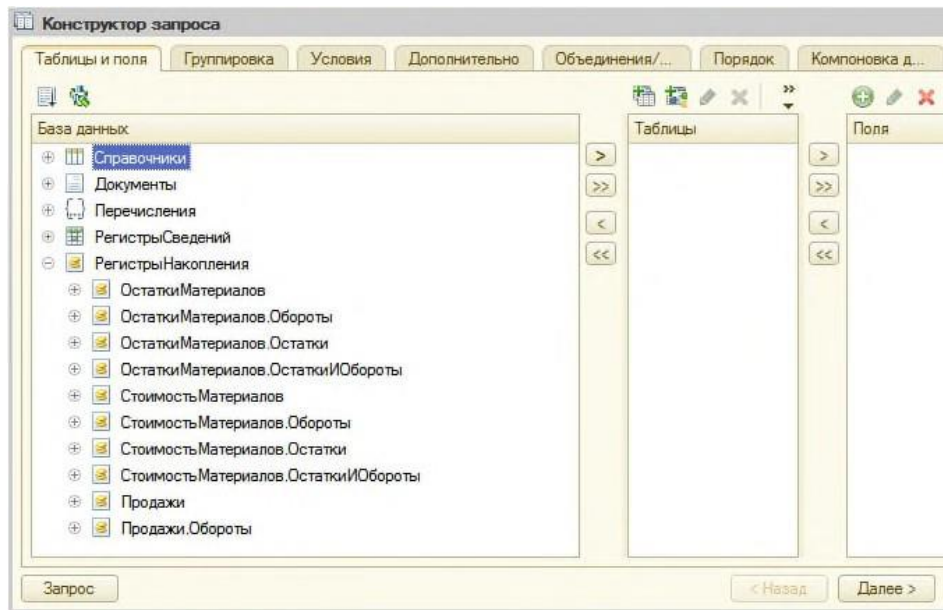


Рис. 7.2. Конструктор запроса

Поскольку мы хотим видеть как остатки материалов, так и информацию об их поступлении и расходовании, нас будет интересовать виртуальная таблица **ОстаткиМатериалов.ОстаткиИОбороты**. Раскроем ее.

Начнем выбирать поля таблицы в нужном нам порядке двойным щелчком мыши. Сначала выберем **Склад** и **Материал**, потом отметим **КоличествоНачальныйОстаток**, **КоличествоПриход**, **КоличествоРасход** и **КоличествоКонечныйОстаток**. Нажмем ОК и вернемся в конструктор схемы компоновки данных.

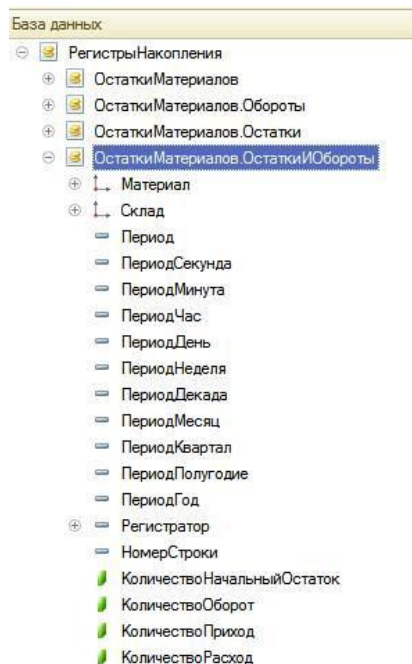


Рис. 7.3. Выбор баз данных

Текст запроса, который был создан с помощью конструктора, платформа поместит в поле

Запрос.

Это поле представляет собой текстовый редактор, в котором можно вручную отредактировать существующий запрос. Кроме того, можно снова вызвать конструктор запроса и отредактировать запрос при помощи него.

Настройки отчета

- На закладке **Настройки**:

- выделим в дереве структуры отчета корневой элемент **Отчет** и вызовем его контекстное меню. Можно также нажать кнопку **Добавить** в командной панели окна или нажать клавишу **Ins**.
- Добавим в отчет группировку - **Новая группировка**. При этом не станем указывать поле группировки, а просто нажмем ОК.

В структуре отчета появится группировка **Детальные записи**.

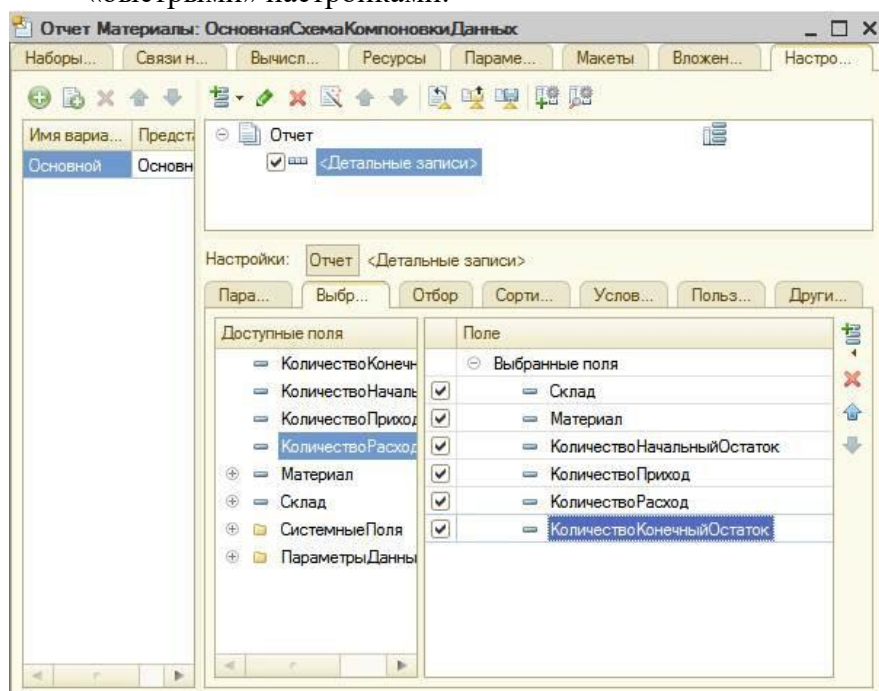
Теперь настроим поля, которые будут выводиться в результат отчета.

- Перейдем в нижнем окне настроек на закладку **Выбранные поля** и перенесем мышью из списка доступных полей:
- Склад,
- Материал,
- КоличествоНачальныйОстаток,
- КоличествоПриход,
- КоличествоРасход,
- КоличествоКонечныйОстаток.

В результате окно настроек отчета должно иметь вид:

Рис. 7.4. Окно настроек отчета

- Перейдем на закладку **Параметры** и укажем, что параметры отчета **Дата начала** и **Дата окончания** будут включены в состав пользовательских настроек, и эти настройки будут находиться непосредственно в форме отчета, то есть будут «быстрыми» настройками.



Сначала укажем, что оба эти параметра будут использоваться в отчете - установим флажки в первой колонке.

Затем выделим каждый из параметров, нажмем кнопку **Свойства элемента пользовательских настроек** и поставим флажок **Включать в пользовательские настройки**.

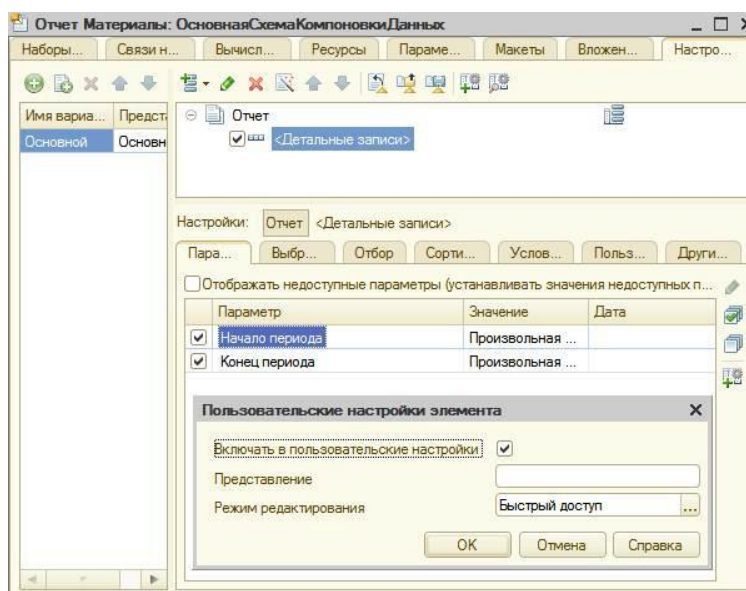


Рис. 7.5. Отчет Материалы

После проделанных действий перейдем по вкладке **Подсистемы**: отметим **Бухгалтерия**, **Учет Материалов**, **Оказание Услуг**.

- **В режиме 1С:Предприятие**

Запустим 1С:Предприятие в режиме отладки и посмотрим, как работает отчет. В открывшемся окне 1С предприятия мы видим, что в панели действий разделов **Бухгалтерия, Оказание услуг и Учет материалов** появилась новая группа команд для выполнения отчетов и в ней команда для формирования отчета **Материалы**. Выполним ее.

Контрольные вопросы

- *Для чего предназначен объект конфигурации Отчет.*
- *Как создать отчет с помощью конструктора схемы компоновки данных.*
- *Как отобразить отчет в разделах прикладного решения.*

Лабораторная работа № 8

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Что такое макет

Объект конфигурации Макет предназначен для хранения различных форм представления данных, различных данных, которые могут потребоваться каким-либо объектам конфигурации или всему прикладному решению в целом.

Макет может содержать табличный или текстовый документ, двоичные данные, HTML-документ или Active Document, графическую или географическую схему, схему компоновки данных или макет оформления схемы компоновки данных.

Макеты могут существовать как сами по себе, так и быть подчинены какому-либо объекту конфигурации.

Одно из предназначений макета, подчиненного объекту конфигурации и содержащего табличный документ, - создание печатной формы этого объекта. Создание печатной формы заключается в конструировании ее составных частей - именованных областей, из которых затем «собирается» готовая печатная форма.

Порядок заполнения областей данными и вывода их в итоговую форму описывается при помощи встроенного языка. Печатная форма может включать в себя различные графические объекты: картинки, OLE-объекты, диаграммы и т. д.

Помимо создания макета «вручную» конструктор предоставляет разработчику возможность воспользоваться специальным инструментом - конструктором печати, который берет на себя большинство рутинной работы по созданию макета.

Макет печатной формы

- **В режиме Конструктор**

Откроем в конструкторе окно редактирования объекта конфигурации **Документ**

ОказаниеУслуги.

- На закладке **Макеты**: запустим конструктор печати.

- В окне **конструктора печати документа** на первом шаге укажем: «Создать новую команду», имя команды: «Печать».

-Нажмем «Далее»-

- На втором шаге нажатием кнопки >> определим, что все реквизиты нашего документа будут отображены в шапке печатной формы:

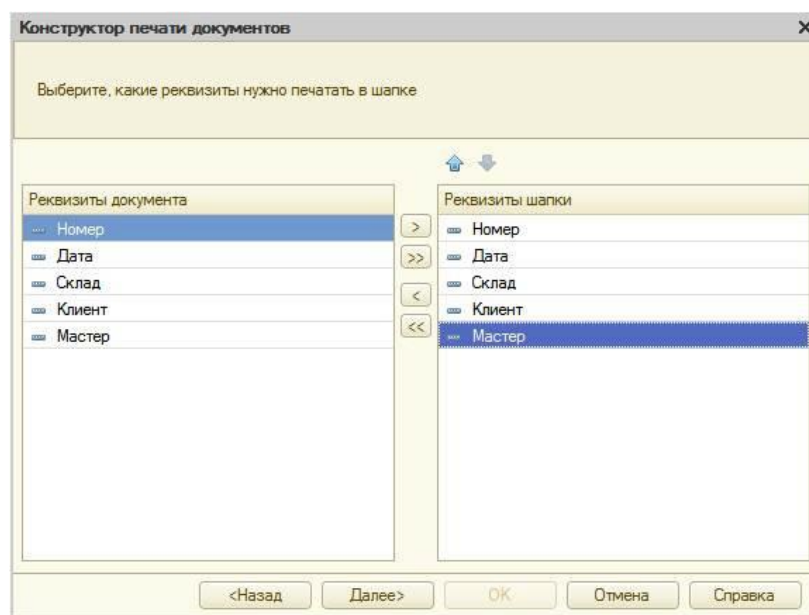


Рис. 8.1. Конструктор печати. Шаг 2

-Нажмем «Далее»-

- На третьем шаге точно так же определим, что все реквизиты табличной части документа будут отображены в печатной форме:

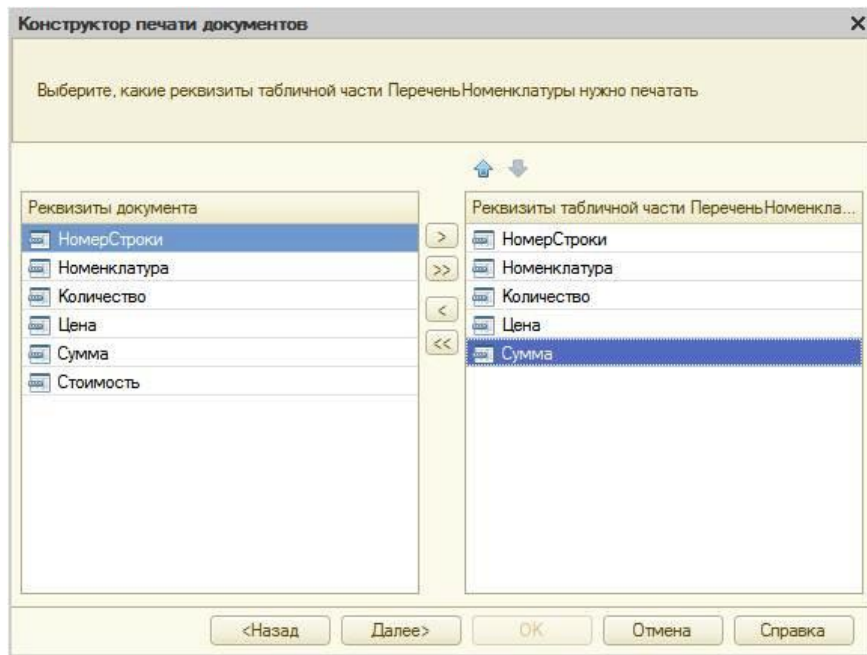


Рис. 8.2. Конструктор печати. Шаг 3

-Нажмем «Далее»-

- На четвертом шаге и пятом шагах указывать ничего не будем. Тем самым согласимся с тем, что команда для вызова процедуры формирования печатной формы будет помещена в командную панель формы, в раздел Важное.

-Нажмем «ОК»-

В конфигураторе откроется модуль команды Печать, модуль менеджера документа **ОказаниеУслуги** и макет этого документа:

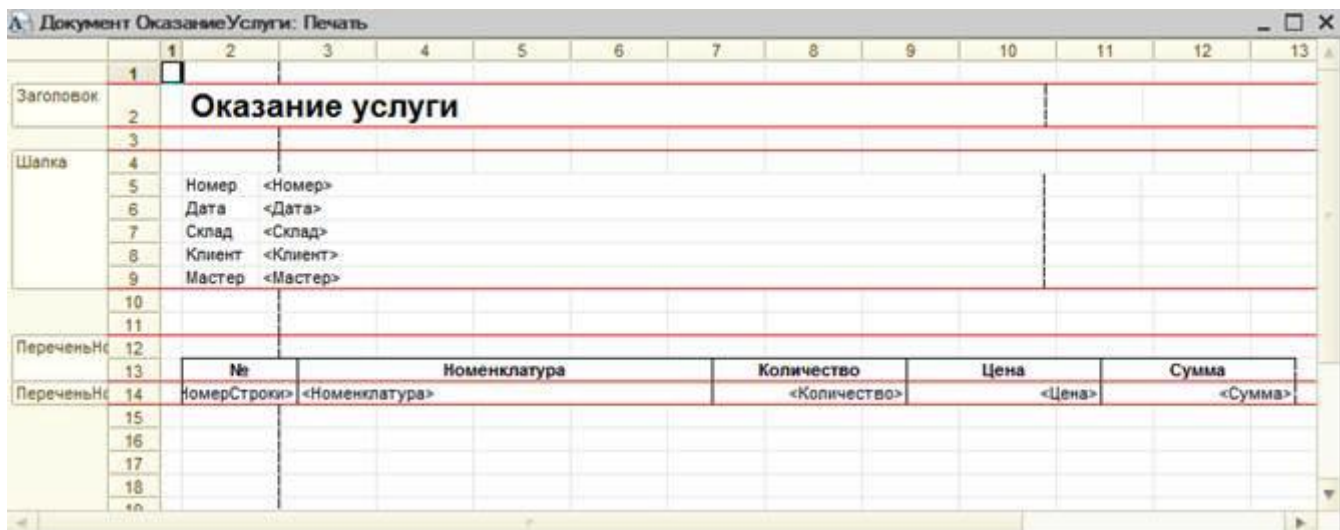


Рис. 8.3. Макет документа "Оказание услуги"

Заметим, что разработчик может создать макет печатной формы с нуля и для ее вывода создать соответствующую команду и кнопку в форме документа, но в данном случае всю работу сделал за нас конструктор печати.

- **В режиме 1с: Предприятие**

Запустим 1С: Предприятие в режиме отладки и откроем документ **Оказание услуги № 1**. Обратите внимание, что в командной панели документа появилась новая кнопка **Печать**. Нажмем на нее и увидим печатную форму нашего документа

Как видите, конструктор сформировал вполне подходящую печатную форму для нашего документа. Единственное, чего не хватает в данной форме, - это итоговой суммы документа.

В следующем разделе, на примере добавления итоговой суммы документа, мы познакомимся с тем, как можно редактировать макеты и формы объектов конфигурации.

Редактирование макета

- **В режиме Конфигуратор**

- Добавим итоговую сумму в печатную форму документа **ОказаниеУслуги**:

1) Откроем конфигуратор, раскроем дерево документа **ОказаниеУслуги** и дважды щелкнем на макете **Печать**.

2) Добавим новую область для вывода итоговой суммы документа. Выделим мышью две пустые строки под табличной частью документа и выполним **Таблица > Имена > Назначить имя...**

3) Назовем область **«Всего»**, нажмем **«ОК»**.

Чтобы формат добавленных нами строк совпадал с имеющимся форматом заголовка и табличной части документа, изменим ширину колонок.

4) Аналогичные действия выполним и для колонок 3,4, 5 и 6.

5) В созданной области, в колонке **Цена**, напишем **ВСЕГО:**, а в колонке **Сумма** напишем

ВсегоПоДокументу.

6) Вызвав палитру свойств для последней заполненной нами ячейки (контекстное меню - Свойства), в свойстве **Заполнение** укажем, что в этой ячейке будет находиться не текст, а параметр.

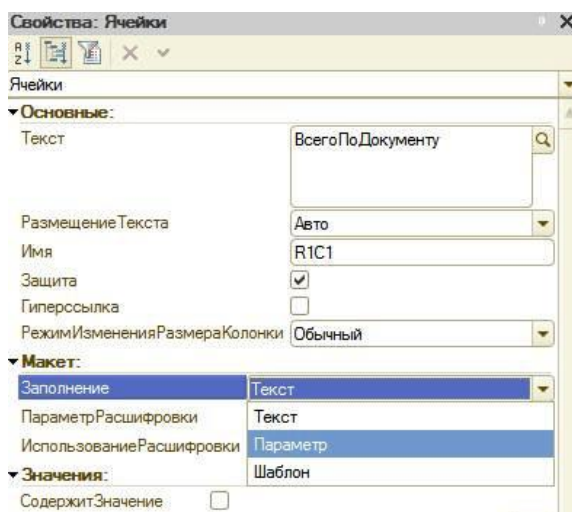


Рис. 8.4. Свойства ячейки "ВсегоПоДокументу"

Каждая ячейка редактируемого нами табличного документа может содержать либо текст, либо некоторый параметр, либо шаблон.

Текст, содержащийся в ячейке, будет показан на экране.

Параметр будет заменен некоторым значением, которое может быть присвоено ему средствами встроенного языка. Текст, содержащийся в ячейке, является именем этого параметра.

Шаблон представляет собой текстовую строку, в определенные места которой будут вставлены значения параметров.

Поэтому, указав для ячейки в качестве заполнения Параметр, мы определили параметр области с именем **ВсегоПоДокументу**, которому присвоим нужное нам значение при формировании печатной формы.

- Откроем модуль менеджера документа **ОказаниеУслуги**:

1) Для этого перейдем на закладку **Прочее** окна редактирования объекта конфигурации Документ **ОказаниеУслуги** и нажмем кнопку Модуль менеджера

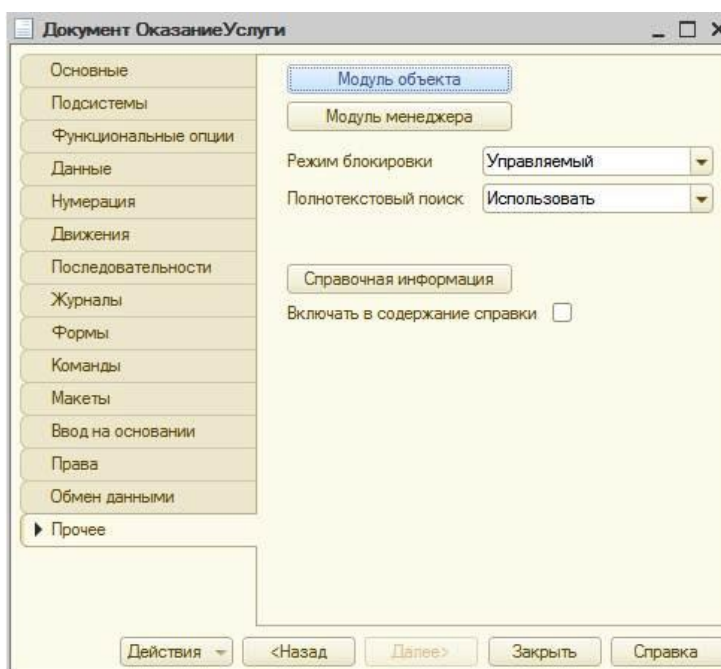


Рис.8.5. Документ ОказаниеУслуги

Найдем в нем процедуру **Печать** и отредактируем ее:

```

Запрос.Параметры.Вставить("Ссылка", Ссылка);
Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");
Шапка = Макет.ПолучитьОбласть("Шапка");
ОбластьПереченьНоменклатурыШапка = Макет.ПолучитьОбласть("ПереченьНоменклатурыШапка");
ОбластьПереченьНоменклатуры = Макет.ПолучитьОбласть("ПереченьНоменклатуры");
ОбластьИтог = Макет.ПолучитьОбласть("Всего");
ТабДок.Счистить();

ВставитьРазделительСтраниц = Ложь;
Пока Выборка.Следующий() Цикл
    Если ВставитьРазделительСтраниц Тогда
        ТабДок.ВывестиГоризонтальныйРазделительСтраниц();
    КонецЕсли;

    ТабДок.Вывести(ОбластьЗаголовок);

    Шапка.Параметры.Заполнить(Выборка);
    ТабДок.Вывести(Шапка, Выборка.Уровень());

    ТабДок.Вывести(ОбластьПереченьНоменклатурыШапка);
    ВыборкаПереченьНоменклатуры = Выборка.ПереченьНоменклатуры.Выбрать();
    СуммаИтог = 0;
    Пока ВыборкаПереченьНоменклатуры.Следующий() Цикл
        ОбластьПереченьНоменклатуры.Параметры.Заполнить(ВыборкаПереченьНоменклатуры);
        ТабДок.Вывести(ОбластьПереченьНоменклатуры, ВыборкаПереченьНоменклатуры.Уровень());
        СуммаИтог = СуммаИтог + ВыборкаПереченьНоменклатуры.Сумма;
    КонецЦикла;

    ОбластьИтог.Параметры.ВсегоПоДокументу = СуммаИтог;
    ТабДок.Вывести(ОбластьИтог);

ВставитьРазделительСтраниц = Истина;
КонецЦикла;

```

Листинг 8.1. Редактируем процедуру Печать

Смысл добавленного фрагмента прост. Мы обращаемся к макету документа **ОказаниеУслуги** по его имени - **Макет**. Используя его метод **ПолучитьОбласть()**, получаем область **Всего** (ту, которую мы только что добавили к макету) и сохраняем её в переменной **ОбластьИтог**. В цикле обхода строк табличной части документа, полученных в результате выполнения запроса, мы накапливаем в переменной **СуммаИтог** значение суммы табличной части документа по колонке **Сумма**.

Затем мы обращаемся к параметру **ВсегоПоДокументу** (**ОбластьИтог.Параметры.ВсегоПоДокументу**), находящемуся в области **Всего**, и присваиваем ему значение переменной **СуммаИтог**. В заключение мы выводим итоговую область в табличный документ, который будет показан на экране и распечатан пользователем - **ТабДок.Вывести(ОбластьИтог)**.

Отображение табличного документа на экране выполняется в обработчике команды **Печать**, в модуле этой команды на клиенте, в то время как сама процедура печати, описанная в модуле менеджера документа, выполняется на сервере.

- **В режиме 1С:Предприятие**

Запустим 1С:Предприятие в режиме отладки и проверим результат наших изменений. Подобным образом, создавая именованные области и ячейки макета, используя их свойства и управляя порядком их вывода с помощью встроенного языка, разработчик имеет возможность создать печатную форму любого дизайна.

А теперь, для того чтобы наш документ **ОказаниеУслуги**, выглядел вполне законченным, добавим итоговую сумму по документу и на экранную форму, чтобы пользователь мог видеть ее в процессе заполнения табличной части документа.

Редактирование формы

- **В режиме Конфигуратор**

После того как мы вывели итоговую сумму по табличной части в печатную форму документа, возникло естественное желание видеть такую же итоговую сумму и в форме документа. Чтобы в процессе его создания можно было оперативно, не печатая документ, знать итоговую сумму по документу.

Для этого мы внесем небольшие изменения в форму документа **ОказаниеУслуги**.

Для редактирования формы документа откроем конфигуратор, раскроем дерево документа **ОказаниеУслуги** и дважды щелкнем на форме **ФормаДокумента**.

1) Дважды щелкнем на элементе **ПереченьНоменклатуры** и установим свойство **Подвал**, которое определяет наличие подвала у таблицы формы.

2) Затем откроем свойства элемента формы **ПереченьНоменклатурыЦена** и установим:

- Текст подвала – **Всего**;
- Горизонтальное положение в подвале – **Право**;
- Свойство шрифт подвала изменим начертание на **Жирный**.

3) Откроем свойства элемента **ПереченьНоменклатурыСумма**, установим:

- Горизонтальное положение в подвале - **Право**.
- Свойство Шрифт подвала - **Жирный**.

Для того чтобы в подвале колонки Сумма отображался итог по ней, нажмем кнопку выбора в поле **ПутьКДаннымПодвала**. Раскроем дерево реквизитов объекта и выберем элемент **ИтогСумма**:

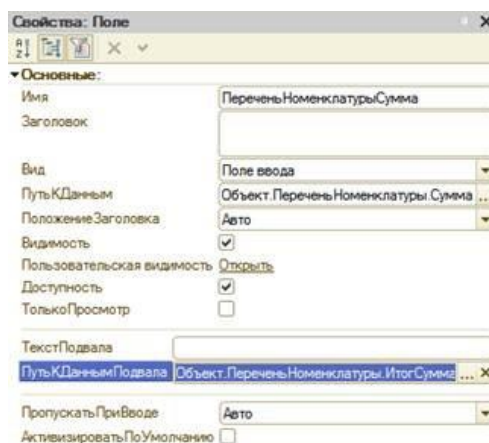
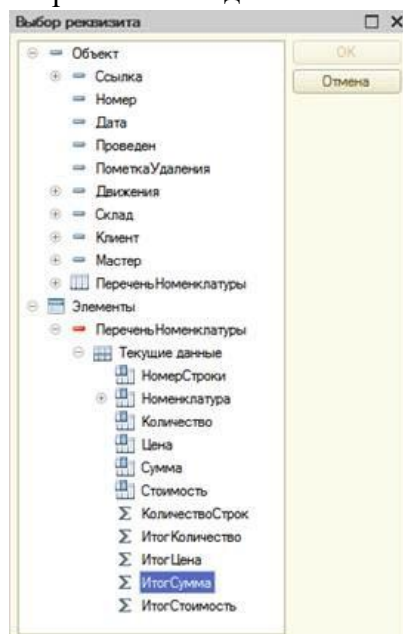


Рис. 8.6. Свойства Поле

Рис. 8.7. Выбор объекта ИтогСумма

- **В режиме 1С:Предприятие**

Запустим 1С:Предприятие в режиме отладки и посмотрим, как теперь выглядит форма



документа Оказание услуги № 1.

Мы видим, что по колонке Сумма в табличной части документа, подсчитывается общий итог документа.

Контрольные вопросы

- Для чего предназначен объект конфигурации Макет.
- Что такое конструктор печати.
- Как создать макет с помощью конструктора печати.
- Как изменить табличный документ.
- Какая разница в заполнении ячейки табличного документа текстом, параметром и шаблоном.
- Как с помощью встроенного языка вывести в табличный документ новую область.
- Как изменить внешний вид и поведение элемента формы.
- Как отобразить сумму по колонке таблицы.

Лабораторная работа № 9 Периодические регистры сведений.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Зачем нужен периодический регистр сведений

Начнем мы с того, что обратим, ваше внимание на документ **Оказание услуги**. Как вы помните, в этом документе мы выбираем услугу, которая оказывается, и затем указываем цену.

Очевидно, что в компании существует перечень услуг, который определяет стоимость каждой услуги. Казалось бы, стоимость услуги является неотъемлемым свойством самой услуги, и поэтому ее следует добавить в качестве реквизита справочника **Номенклатура**. Однако стоимость услуг имеет особенность меняться со временем. И может сложиться такая ситуация, когда нам потребуется внести изменения или уточнения в один из ранее проведенных документов **Оказание услуги**. В этом случае мы не сможем получить

правильную стоимость услуги, поскольку в реквизите справочника будет храниться последнее введенное значение.

Для хранения стоимости услуг мы используем новый объект - **Регистр сведений**.

Что такое регистр сведений

Объект конфигурации Регистр сведений предназначен для описания структуры хранения данных в разрезе нескольких измерений. На основе объекта конфигурации Регистр сведений платформа создает в базе данных таблицу, в которой может храниться произвольная информация, «привязанная» к набору измерений.

Принципиальное отличие регистра сведений от регистра накопления заключается в том, что каждое движение регистра сведений устанавливает новое значение ресурса, в то время как движение регистра накопления изменяет существующее значение ресурса. По этой причине регистр сведений может хранить любые данные.

Следующей важной особенностью регистра сведений является его способность хранить данные с привязкой ко времени. Благодаря этому регистр сведений может хранить не только актуальные значения данных, но и историю их изменения во времени - периодический регистр сведений.

Периодичность регистра сведений можно определить одним из следующих значений:

- в пределах секунды,
- в пределах дня,
- в пределах месяца,
- в пределах квартала,
- в пределах года,
- в пределах регистратора .

Периодический регистр сведений всегда содержит служебное поле **Период**, добавляемое системой автоматически. Оно имеет тип **Дата** и служит для указания факта принадлежности записи к какому-либо периоду. При записи данных в регистр платформа всегда приводит значение этого поля к началу того периода, в который он попадает. Например, если в регистр сведений с периодичностью в пределах месяца записать данные, в которых период указан как 08.04.2004, то регистр сохранит эти данные со значением периода, равным 01.04.2004. Как и для других регистров, система контролирует уникальность записей для регистрасведений. Однако если для прочих регистров уникальным идентификатором записи является регистратор и номер строки, то для регистра сведений применяется другой принцип формирования ключевого значения.

Ключом записи, однозначно идентифицирующим запись, является в данном случае совокупность значений измерений регистра и периода (в случае если регистр сведений периодический). Регистр сведений не может содержать несколько записей с одинаковыми ключами.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Добавление периодического регистра сведений

- **В режиме Конфигуратор**

Откроем в конфигураторе конфигурацию и добавим новый объект конфигурации **Регистр**

сведений.

Для этого выделим в дереве объектов конфигурации ветвь **Регистры сведений** и нажмем кнопку **Добавить**.

- На закладке **Основные**: Имя регистра - **Цены**. Периодичность регистра - **В пределах секунды**. Представление записи - **Цена**, а Представление списка - **Цены на номенклатуру**. Режим записи – **Независимый**.

Нажмем «Далее»

- На закладку **Подсистемы**: отметим в списке следующие разделы: **Учет материалов, Оказание услуг и Бухгалтерия**.

Нажмем «Далее»

- На закладке **Данные**: а) создадим измерение **Номенклатура** с типом **СправочникСсылка.Номенклатура**. Укажем, что это измерение будет **Ведущим**.

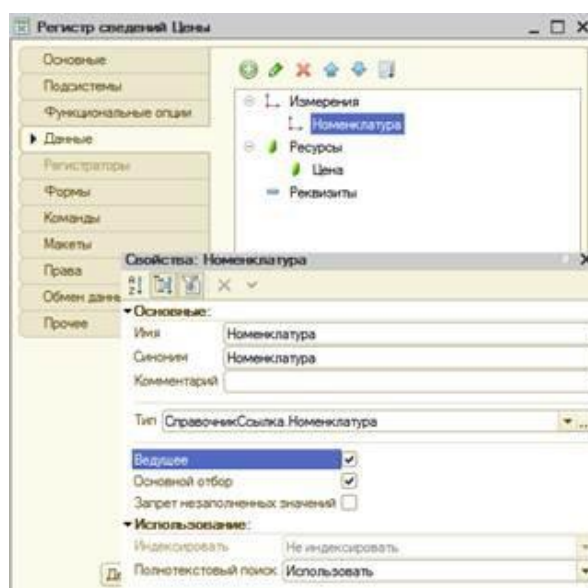


Рис. 9.1. Измерение Номенклатура будет Ведущим

Свойство **Ведущее** имеет смысл использовать лишь тогда, когда измерение имеет тип ссылки на объект базы данных. Установка свойства **Ведущее** будет говорить о том, что запись регистра сведений представляет интерес, пока существует тот объект, ссылка на который выбрана в качестве значения этого измерения в этой записи. При удалении объекта, все записи регистра сведений по этому объекту тоже будут автоматически удалены.

б) создадим ресурс **Цена**, тип **Число**, длина 15, точность 2, неотрицательное.

Для этого выделим ветвь **Ресурсы** и нажмем кнопку **Добавить** в командной панели окна:

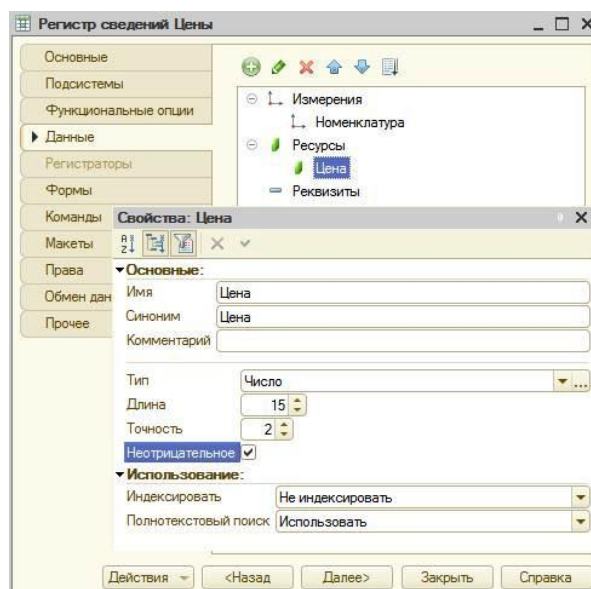


Рис. 9.2. Свойства Цена

- **В режиме 1С: Предприятие**

Теперь запустим 1С: Предприятие в режиме отладки и посмотрим, как работает наш периодический регистр сведений **Цены**.

В открывшемся окне 1С: Предприятия мы видим, что в панели навигации разделов **Бухгалтерия, Оказание услуг и Учет материалов** появилась команда для открытия списка регистра **Цены на номенклатуру**.

Создание записей в регистре сведений

Чтобы добавить новую запись в регистр сведений, нажмем кнопку **Создать**.

Зададим стоимость услуг. При этом период зададим прошлым числом, так как он должен быть меньше или равен дате создания документа об оказании услуг.

Автоматическая подстановка цены в документ при выборе номенклатуры

Для этого нужно сделать две вещи:

- Написать некую функцию, которая будет возвращать актуальную цену номенклатуры.
- Вызвать эту функцию в тот момент, когда в документ добавляется номенклатура, и подставить в документ цену номенклатуры, которую вернет функция.

Необходимо данную функцию разместить в общедоступном, для других документов, месте.

- **В режиме Конфигуратор**

*Создадим функцию **РозничнаяЦена()**, которая будет возвращать нам актуальную розничную цену номенклатуры, и поместим ее в общий модуль конфигурации:*

Откроем конфигуратор, в ветке **Общие > Общие модули** добавим новый объект конфигурации **Модуль** и назовем его **РаботаСоСправочниками**.

Установим флажок **Вызов сервера** для того, чтобы экспортные процедуры и функции

этого модуля можно было вызывать с клиента:

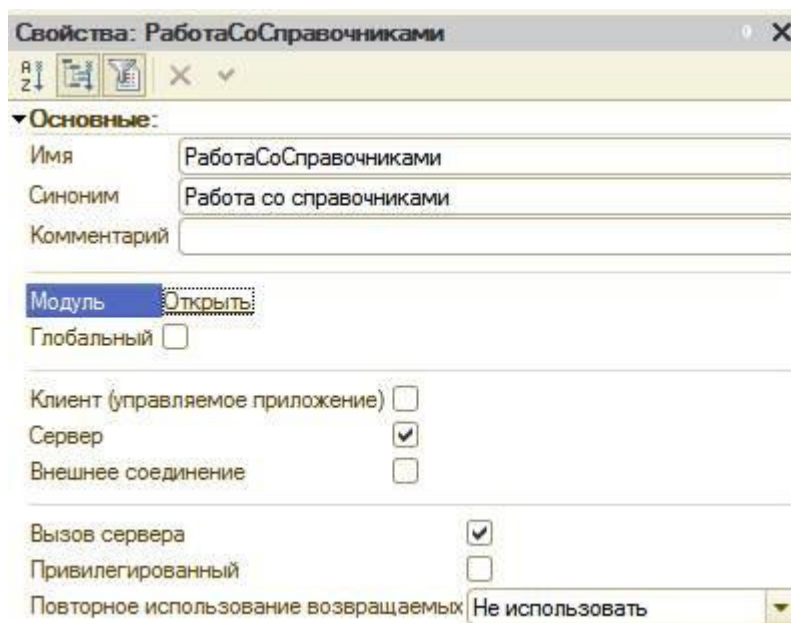
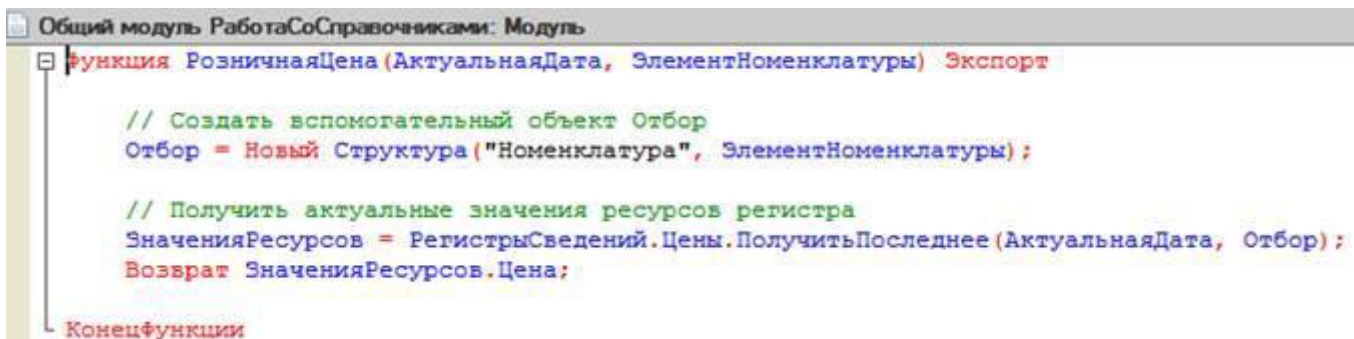


Рис. 9.3. Свойства РаботаСоСправочниками

Затем поместим в модуль следующий текст (см. Листинг 1):



Листинг 9.1. Редактируем Общий Модуль РаботаСоСправочниками

Для получения розничной цены мы будем передавать в функцию два параметра:

- **АктуальнаяДата** - параметр типа Дата, определяет точку на оси времени, в которой нас интересует значение розничной цены.
- **ЭлементНоменклатуры** - ссылка на элемент справочника Номенклатура, для которого мы хотим получить розничную цену.

Теперь эту функцию нужно вызвать в некоторый момент работы документа.

Вызов функции при выборе номенклатуры и заполнение цены в документе

1 - Найдем в конфигураторе документ **ОказаниеУслуги** и откроем его форму

Форма Документа.

2 - Дважды щелкнем на элементе формы **ПереченьНоменклатурыНоменклатура**.

Прокрутим список до конца, найдем событие **ПриИзменении**, которое возникает после изменения значения поля.

3 - Нажмем кнопку открытия со значком лупы в поле ввода. Система создаст шаблон процедуры обработчика этого события в модуле нашей формы и откроет закладку Модуль редактора формы. Внесем в него следующий текст (см. Листинг 2):

```
// Получить текущую строку табличной части
СтрокаТабличнойЧасти = Элементы.ПереченьНоменклатуры.ТекущиеДанные;

// Установить цену
СтрокаТабличнойЧасти.Цена = РаботаСоСправочниками.РозничнаяЦена(Объект.Дата, СтрокаТабличнойЧасти.Номенклатура);

// Пересчитать сумму строки
РаботаСДокументами.РассчитатьСумму(СтрокаТабличнойЧасти);
```

Листинг 9.2. Модуль редактора формы

- **В режиме 1С: Предприятие**

Запустим 1С:Предприятие в режиме отладки и откроем регистр сведений **Цены**. Для элемента таблицы добавим другим числом новую цену.

Теперь откроем документ **Оказание услуги № 1**. Оставим дату документа без изменения и повторим выбор транзистора в колонке **Номенклатура** табличной части документа. Автоматически установится последнее значение цены товара.

Контрольные вопросы

- Для чего предназначен объект конфигурации Регистр сведений.
- Какими особенностями обладает объект конфигурации Регистр сведений.
- В чем главные отличия регистра сведений от регистра накопления.
- Какие поля определяют ключ уникальности регистра накопления.
- Что такое периодический регистр сведений и что такое независимый регистр сведений.
- Как создать периодический регистр сведений.
- Что такое ведущее измерение регистра.
- Как получить значения ресурсов наиболее поздних записей регистра средствами встроенного языка.

Лабораторная работа № 10 Перечисления.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Что такое перечисление

Объект конфигурации Перечисление предназначен для описания структуры хранения постоянных наборов значений, не изменяемых в процессе работы конфигурации. На основе объекта конфигурации Перечисление платформа создает в базе данных таблицу, в которой может храниться набор некоторых постоянных значений.

В реальной жизни этому объекту может соответствовать, например, перечисление вариантов указания цены («включая НДС», «без НДС»). Набор всех возможных значений, которые содержит перечисление, задается при конфигурировании системы, и пользователь не может изменять их, удалять или добавлять новые.

Особенность перечисления: значения перечисления не «обезличены» для конфигурации, на них могут опираться алгоритмы работы программы.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Добавление перечисления

- В режиме Конфигуратор

Откроем конфигуратор и создадим сначала новый объект конфигурации **Перечисление** с именем **ВидыНоменклатуры**.

- На закладке **Данные**: нажмем на «+» и добавим два значения перечисления: **Материал** и **Услуга**.

Привязка номенклатуры к значениям перечисления «ВидНоменклатуры»

Чтобы привязать номенклатуру к значениям перечисления, мы сделаем следующее:

- В режиме Конфигуратор создадим у справочника **Номенклатура** реквизит, который будет хранить значение перечисления.
- В режиме 1С: Предприятие проставим нужные значения этого реквизита для всех элементов справочника **Номенклатура**.
- **В режиме Конфигуратор**

- На закладке **Данные**: добавим в справочник **Номенклатура** новый реквизит

ВидНоменклатуры с типом **ПеречислениеСсылка.ВидыНоменклатуры**.

- **В режиме 1С:Предприятие**

После этого запустим 1С: Предприятие в режиме отладки.

Платформа выдаст предупреждающее сообщение о том, что наше перечисление не включено ни в одну подсистему. Проигнорируем его и примем изменения конфигурации. В режиме 1С: Предприятие зададим для каждого элемента справочника **Номенклатура** соответствующее значение реквизита **ВидНоменклатуры**:

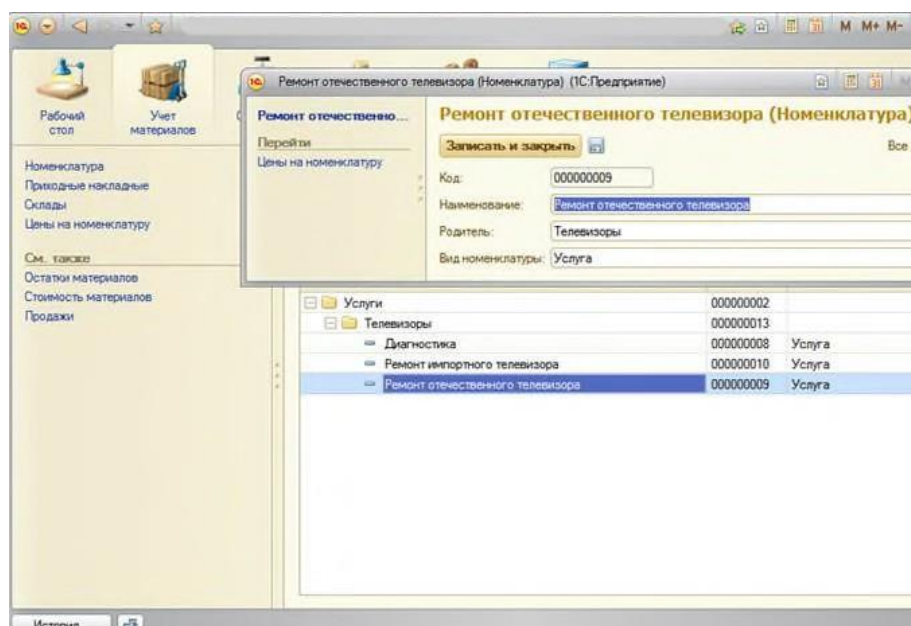


Рис. 10.1. Зададим значение реквизита ВидНоменклатуры

Регистрация расхода только той номенклатуры, которая является материалом

Сначала в режиме Конфигуратор изменим процедуру проведения документа так, чтобы в регистр попадали записи только о той номенклатуре, которая является материалом, а потом в режиме 1С:Предприятие заново проведем (перепроведем) все документы **Оказание услуги**, чтобы данные в регистре изменились в соответствии с новым алгоритмом проведения документа.

Эта доработка будет не совсем эффективна с точки зрения производительности, зато позволит нам получить нужные данные в регистре **ОстаткиМатериалов**.

- **В режиме Конфигуратор**

Скорректируем движения документа, исключив из обработки те строки табличной части, в которых находятся услуги.

Для этого откроем в конфигураторе модуль документа **ОказаниеУслуги** (контекстное меню документа - Открыть модуль объекта) и добавим в обработчик события

ОбработкаПроведения это условие.

Текст следует добавить в начало цикла обхода табличной части документа после строки **Для Каждого ТекСтрокаПереченьНоменклатурыИз ПереченьНоменклатурыЦикл**. В результате процедура **ОбработкаПроведения** должна выглядеть следующим образом:

```

Процедура ОбработкаПроведения(Отказ, Режим)
    Движения.ОстаткиМатериалов.Записывать = Истина;
    Движения.СтоимостьМатериалов.Записывать = Истина;
    Движения.Продажи.Записывать = Истина;

    Для Каждого ТекСтрокаПереченьНоменклатурыИз ПереченьНоменклатурыЦикл
        Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры = Перечисления.ВидНоменклатуры.Материал Тогда
            Движение = Движения.ОстаткиМатериалов.Добавить();
            Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
            Движение.Период = Дата;
            Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
            Движение.Склад = Склад;
            Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

            // регистр СтоимостьМатериалов.Расход
            Движение = Движения.СтоимостьМатериалов.Добавить();
            Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
            Движение.Период = Дата;
            Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
            Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Количество * ТекСтрокаПереченьНоменклатуры.Стоимость;
        КонецЕсли;

        // Регистр Продажи
        Движение = Движения.Продажи.Добавить();
        Движение.Период = Дата;
        Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
        Движение.Клиент = Клиент;
        Движение.Мастер = Мастер;
        Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
        Движение.Варучка = ТекСтрокаПереченьНоменклатуры.Сумма;
        Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Стоимость * ТекСтрокаПереченьНоменклатуры.Количество;
    КонецЦикла;
КонецПроцедуры

```

Листинг 10.1. Скорректируем движения документа

Добавленный текст исключает выполнение операторов цикла для тех строк табличной части документа, в которых номенклатура не является материалом.

- **В режиме 1С:Предприятие**

Запустим 1С:Предприятие в режиме отладки и проверим работу процедуры проведения документа **Оказание услуги**.

Откроем список документов, выполнив команду **Оказание услуг** в панели навигации раздела **Оказание услуг**.

Откроем документ **Оказание услуги № 1** и внесем в него изменения:

- удалим из табличной части строку какого-нибудь материала;
- добавим новую услугу;
- добавим материал для этой услуги.

Обратите внимание, что цены подставляются автоматически из регистра сведений **Цены**. Нажмем кнопку **Провести** в командной панели формы документа.

Затем выполним команду **Остатки материалов** в панели навигации формы, чтобы перейти к записям регистра **Остатки материалов**, связанным с данным документом.

Как мы видим, в движения по регистру **Остатки материалов** включаются только строки, содержащие материалы. Запись про новую услугу в движения не попала.

Контрольные вопросы

- Для чего предназначен объект конфигурации *Перечисление*.
- Как создать новое перечисление.

- *Как с помощью перечисления задать принадлежность элементов справочника к той или иной смысловой группе.*
- *Как обратиться к значению перечисления средствами встроенного языка.*

Лабораторная работа № 11 Проведение документа по нескольким регистрам.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Зачем нужно проведение документа по нескольким регистрам

В нашей базе учитывается только количественное движение материалов, очевидно, что необходимо также знать, какие денежные средства были затрачены на приобретение тех или иных материалов и каковы материальные запасы в денежном выражении. На ряду с этим необходимо, чтобы весь суммовой учет материалов велся по средней стоимости.

То есть при закупке материалов они должны учитываться в ценах приобретения, а при расходе - по средней стоимости, которая рассчитывается исходя из общей суммы закупок данного материала и общего количества этого материала.

Поскольку подобная информация имеет совершенно другую структуру, нежели количественный учет, для хранения данных об общей стоимости тех или иных материалов будет использоваться еще один регистр накопления **СтоимостьМатериалов**.

Таким образом, документы **ПриходнаяНакладная** и **ОказаниеУслуги** должны будут создавать движения не только в регистре **ОстаткиМатериалов**, но одновременно и в регистре **СтоимостьМатериалов**, отражая изменения суммового учета.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Добавление еще одного регистра накопления

- **В режиме Конфигуратор**

Создадим новый объект конфигурации Регистр накопления с именем

СтоимостьМатериалов.

- На закладке **Основные**: зададимРасширенное представление списка - **Движения по регистру Стоимость материалов**.

- На закладке **Подсистемы**: отметим, что этот регистр будет отображаться в подсистемах

Бухгалтерия, Учет материалов и Оказание услуг.

- На закладке **Данные**: создадим для регистра одно измерение -**Материал** с типом **СправочникСсылка.Номенклатура** и один ресурс -**Стоимость** с длиной 15 и точностью 2.

- *Отредактируем командный интерфейс:*

В дереве объектов конфигурации выделим ветвь **Подсистемы**, вызовем ее контекстное меню и выберем пункт **Все подсистемы**. В открывшемся окне слева в списке Подсистемы выделим подсистему **Бухгалтерия**.

В группе **Панель навигации.Обычное** включим видимость у команды **Стоимость материалов** и мышью перетащим ее в группу **Панель навигации.См.также**.

Аналогично, выделив подсистемы **ОказаниеУслуг** и **УчетМатериалов**, в панели навигации в группе **Обычное** включим видимость у команды **Стоимость материалов** и перенесем ее в группу **См.также**.

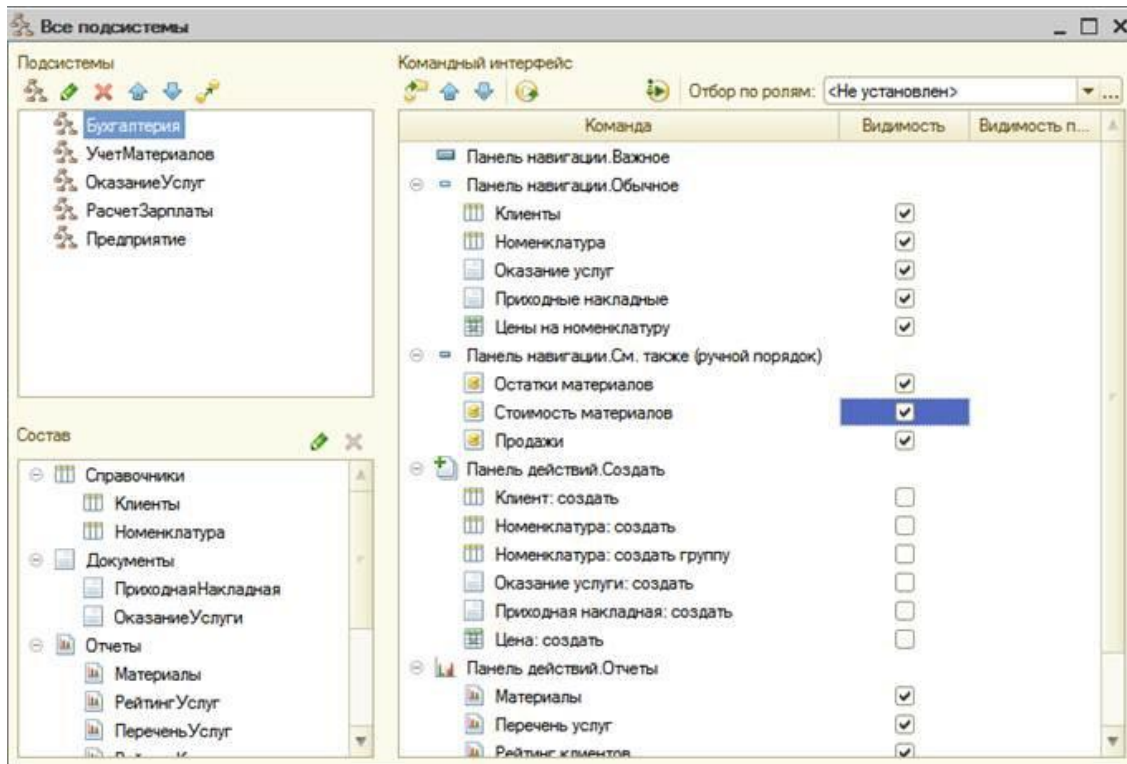


Рис. 11.1. Проведение приходной накладной по двум регистрам

- **В режиме Конфигуратор**

Откроем в конфигураторе окно редактирования объекта конфигурации Документ

ПриходнаяНакладная.

- На закладке **Движения**: В списке регистров отметим **СтоимостьМатериалов**.

- На закладке **Прочее**: нажмем кнопку **Модуль объекта**, откроем процедуру обработчика события **ОбработкаПроведения**. И изменим код следующим образом: (см.листинг):

Листинг 11.1

Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ОстаткиМатериалов.Записывать = Истина;

Движения.СтоимостьМатериалов.Записывать = Истина;

Для Каждого ТекСтрокаМатериалы Из Материалы Цикл

//регистр ОстаткиМатериалов Приход

Движение = Движения.ОстаткиМатериалов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Период = Дата;

Движение.Материал = ТекСтрокаМатериалы.Материал;

Движение.Склад=Склад;

Движение.Количество = ТекСтрокаМатериалы.Количество;

//регистр СтоимостьМатериалов Приход

Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаМатериалы.Материал;
Движение.Стоимость=ТекСтрокаМатериалы.Сумма;
КонецЦикла;
КонецПроцедуры

Команда перехода к записям регистра

В заключение отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра

СтоимостьМатериалов, связанному с документом.

- Откроем форму документа **ПриходнаяНакладная**: перейдем на закладку **Командный интерфейс**. В разделе **Панель навигации** раскроем группу **Перейти** и увидим команду для открытия регистра накопления **Стоимость материалов**. Установим свойство **Видимость** для этой команды:

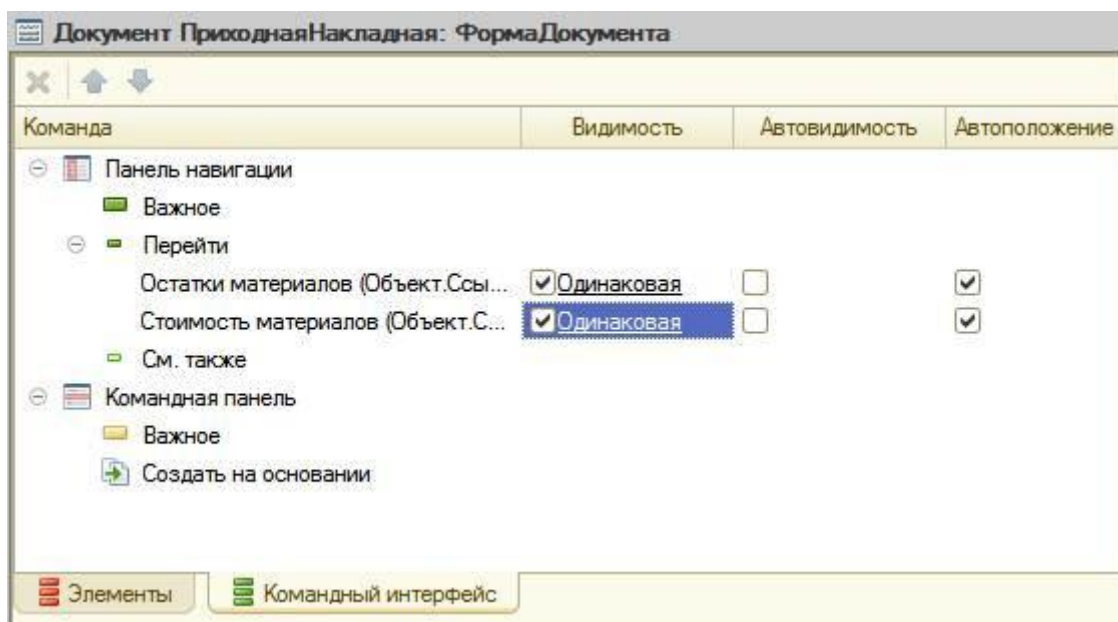


Рис. 11.2. Редактируем командный интерфейс формы документа

- **В режиме 1С:Предприятие**

В режиме 1С:Предприятие необходимо перепровести все приходные накладные. Это необходимо для того, чтобы эти документы создали новые записи в регистрах, в соответствии с алгоритмом проведения.

Запустим 1С: Предприятие в режиме отладки. Откроем список документов, выполнив команду **Приходные накладные** в панели навигации раздела **Учет материалов**.

Выделим одновременно, используя клавишу **Ctrl**, все приходные накладные и перепроведем их, выполнив команду **Все действия > Провести**.

Проведение документа «Оказание услуги» по двум регистрам

В заключение внесем изменения в процедуру обработки проведения документа

ОказаниеУслуги.

Это необходимо для того, чтобы на первом этапе, при списании материалов, израсходованных в процессе оказания услуги, должна быть возможность указывать различную стоимость для одного и того же материала, которая рассчитана руководством исходя из текущих конъюнктурных соображений.

Поскольку в документе **ОказаниеУслуги** отражена только цена номенклатуры, необходимо сделать следующее:

- Добавить в табличную часть документа еще один реквизит, в котором будет указываться стоимость номенклатуры.
 - После этого изменить процедуру проведения документа **ОказаниеУслуги**.
 - В режиме 1С: Предприятие перепровести все эти документы.
-
- **В режиме Конфигуратор**

Откроем в конфигураторе окно редактирования объекта конфигурации Документ

ОказаниеУслуги

- На закладке **Данные**: создадим новый реквизит табличной части документа с именем **Стоимость**, типом **Число**, длиной 15 и точностью 2, и отметим флажок **неотрицательное**:

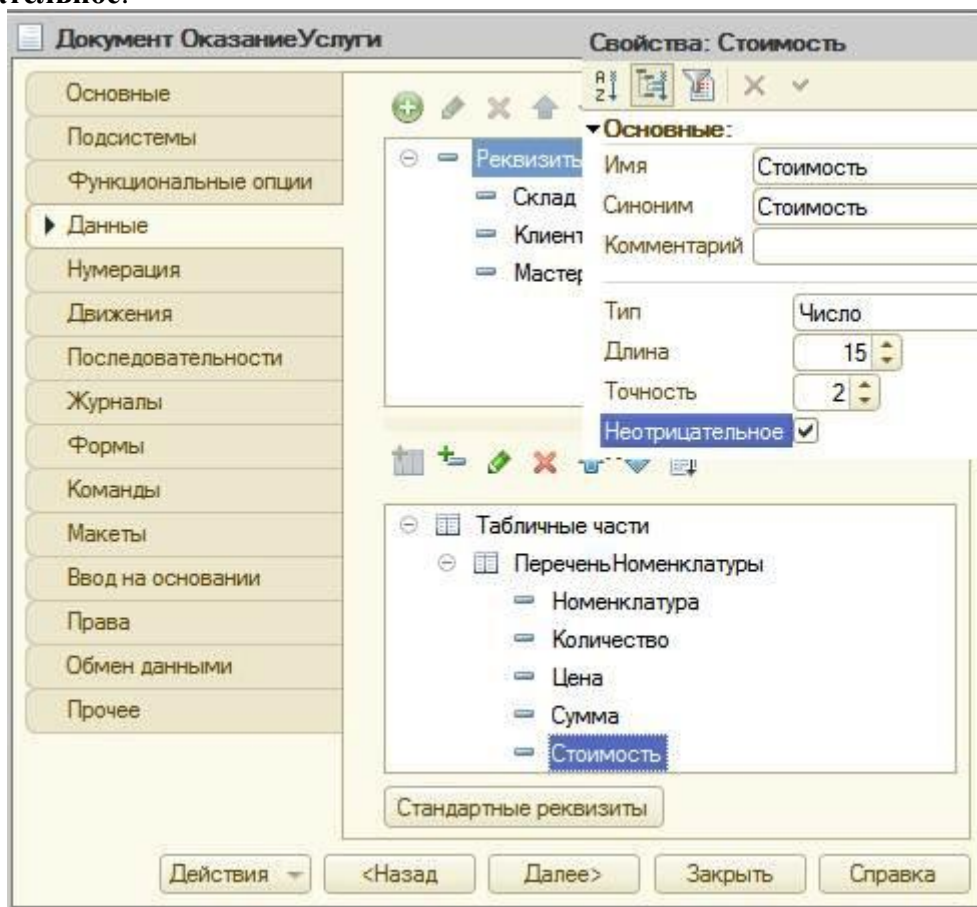


Рис. 11.3. Редактируем свойства Стоимость

Откроем форму **ФормаДокумента** документа **ОказаниеУслуг** и добавим табличную часть **ПереченьНоменклатуры** поле, отображающее новый реквизит **Стоимость**:
 - На закладке **Реквизиты**: раскроем реквизит формы **Объект**.
 Найдем в табличной части реквизит **Стоимость** и с помощью мыши перетащим его в окно элементов формы, расположенное слева в верхней части редактора форм:

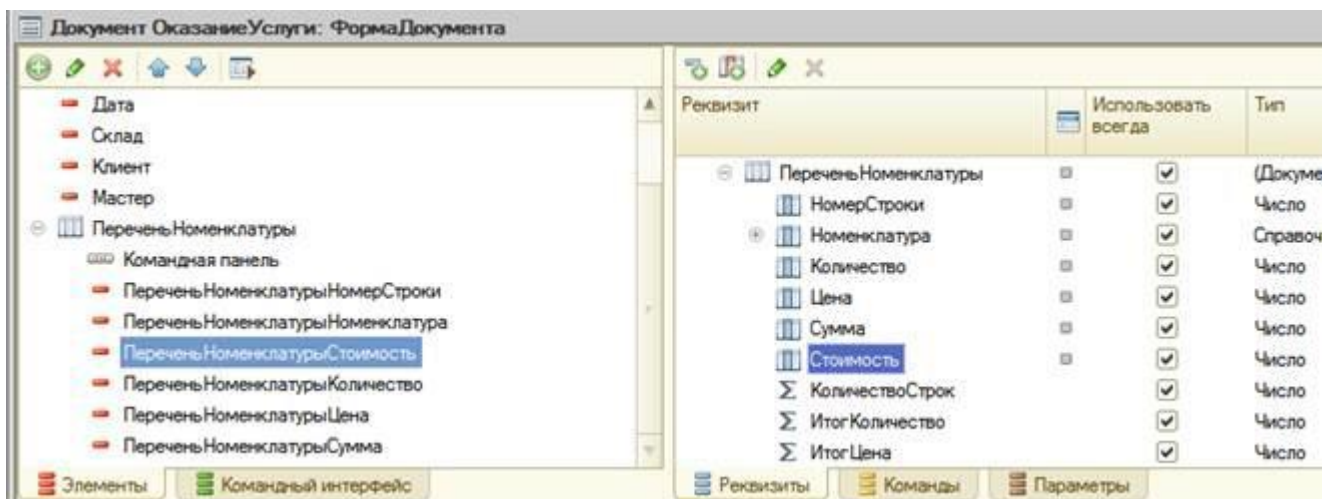


Рис. 11.4. Редактируем реквизит формы **Объект**

Новый элемент расположим в структуре элементов формы после поля **Номенклатура**. Оставим свойства элемента формы, предложенные по умолчанию.
 Новый реквизит сразу же отобразится в форме документа, расположенной в левом нижнем окне редактора форм.

Изменение процедуры проведения

Создадим движения документа **ОказаниеУслуги** таким же образом, как мы делали это для документа **Приходная Накладная**.

- На закладке **Движения**: в списке регистров отметим, что документ будет создавать теперь движения и по регистру **СтоимостьМатериалов**.

- На закладке **Прочее**: нажмем кнопку **Модуль объекта**. Откроем процедуру обработчика события **ОбработкаПроведения**. Отредактируем код следующим образом:

Листинг 11.2

Процедура ОбработкаПроведения(Отказ, Режим)

*Движения.ОстаткиМатериалов.Записывать = Истина;
 Движения.СтоимостьМатериалов.Записывать = Истина;*

*Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
 Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры
 = Перечисления.ВидыНоменклатуры.Материал Тогда*

//регистр ОстаткиМатериалов Расход

Движение = Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Склад=Склад;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

//регистр СтоимостьМатериалов Расход

Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Стоимость=ТекСтрокаПереченьНоменклатуры.Количество
**ТекСтрокаПереченьНоменклатуры.Стоимость;*
КонецЕсли;
КонецЦикла;
КонецПроцедуры

В заключение отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра **Стоимость Материалов**, связанному с документом.

- *Откроем форму документа **ОказаниеУслуги**: перейдем на закладку **Командный интерфейс**. В разделе **Панель навигации** раскроем группу **Перейти** и увидим команду для открытия регистра накопления **Стоимость материалов**. Установим свойство **Видимость** для этой команды.*

- **В режиме 1С: Предприятие**

В режиме 1С:Предприятие необходимо перепровести документ оказания услуги. Это необходимо для того, чтобы этот документ создал новые записи в регистрах. Запустим 1С: Предприятие в режиме отладки и откроем список документов, выполнив команду **Оказание услуг** в панели навигации раздела **Оказание услуг**.

Откроем документ **Оказание услуги № 1** и изменим в нем стоимость одного из материалов. Нажмем кнопку **Провести** и выполним команду перехода к регистру **Стоимость материалов**.

Создадим и проведем еще два документа **Оказание услуги**.

Для этого в форме списка документов нажмем кнопку **Создать**. Эти документы понадобятся нам в дальнейшем.

Контрольные вопросы

- *Для чего может понадобиться проведение документа по нескольким регистрам.*
- *Как создать движения документа по нескольким регистрам в обработке проведения документа.*
- *Как создать движения документа без использования конструктора движений.*
- *Как средствами встроенного языка сформировать и записать движения документа в регистр накопления.*
- *Как добавить в форму документа новый реквизит.*

Лабораторная работа № 12 Оборотные регистры накопления.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Что такое оборотный регистр накопления

Регистры накопления могут быть регистрами остатков и регистрами оборотов. Существующие в нашей конфигурации регистры **ОстаткиМатериалов** и **СтоимостьМатериалов** являются регистрами остатков.

Если вы помните, при создании отчета *Материалы* в конструкторе запроса мы видели, что для таких регистров система создает три виртуальные таблицы: таблица остатков, оборотов и совокупная таблица остатков и оборотов.

Оборотный регистр накапливает только обороты, остатки ему безразличны. Поэтому единственной виртуальной таблицей, которую будет создавать система для такого регистра, будет таблица оборотов. В остальном оборотный регистр ничем не отличается от регистра остатков.

Следует сказать об одной особенности конструирования регистров накопления, напрямую связанной с возможностью получения остатков. При создании оборотного регистра накопления нет особой сложности в определении того, какие именно данные должны являться измерениями регистра - мы можем назначить в качестве его измерений любые нужные нам данные.

Совсем иная ситуация в случае регистра накопления, поддерживающего накопление остатков. Для него выбор измерений должен выполняться исходя из того, что движения регистра могут быть осуществлены в две стороны: приход и расход. Таким образом, в качестве измерений нужно выбирать те данные, по которым движения точно будут осуществляться как в одну, так и в другую сторону.

Скорее всего, при поступлении материалов поставщик будет указан, а вот при расходовании материалов, с большой долей вероятности, поставщик указываться не будет. В большинстве случаев это совершенно лишняя информация.

Значит, поставщика следует добавить не как измерение, а как реквизит регистра накопления.

Если же при расходе материалов поставщик будет указываться наверняка, имеет смысл добавить поставщика в измерения регистра.

Иными словами, по каждому из измерений регистра накопления остатков ресурсы обязательно должны изменяться в обе стороны: приход и расход. Не должно существовать таких измерений, по которым осуществляется только приход или только расход.

Нарушение этого принципа построения регистров накопления будет вести к непроизводительному использованию ресурсов системы и как следствие к замедлению работы и падению производительности.

Для реквизитов же регистра этот принцип не важен. По реквизитам регистра ресурсы могут только приходоваться или только расходоваться.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Добавление оборотного регистра накопления

- **В режиме Конфигуратор**

Откроем конфигуратор и создадим новый объект конфигурации Регистр накопления.

- На закладке **Основные**: имя - Продажи; вид регистра – **Обороты**; расширенное представление списка – **Движения документа по регистру Продажи**.

- На закладке **Подсистемы**: отметим, что этот регистр будет отображаться в подсистемах **Бухгалтерия, Учет материалов и Оказание услуг**.

- На закладке **Данные**: создадим измерения регистра:

- **Номенклатура**, тип **СправочникСсылка.Номенклатура**;
- **Клиент**, тип **СправочникСсылка.Клиенты**;
- **Мастер**, тип **СправочникСсылка.Сотрудники**.

Создадим три ресурса

- **Количество**, тип **Число**, длина 15, точность 3;
- **Выручка**, тип **Число**, длина 15, точность 2;
- **Стоимость**, тип **Число**, длина 15, точность 2.

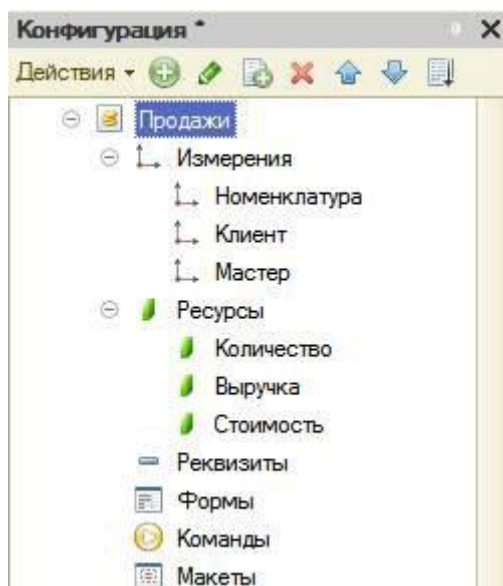


Рис. 12.1. Создание регистров

- В дереве объектов конфигурации выделим ветвь **Подсистемы**, вызовем **Все подсистемы**: В открывшемся окне выделим подсистему **Бухгалтерия**; в группе **Панель навигации.Обычное** включим видимость у команды **Продажи** и мышью перетащим ее в группу **Панель навигации.См.также**.

Аналогично, выделив подсистемы **ОказаниеУслуг** и **УчетМатериалов**, в панели навигации включим видимость у команды **Продажи** и перенесем ее в группу **См.также**.

Проведение документа «Оказание услуги» по трем регистрам

Изменим процедуру проведения документа **ОказаниеУслуги**, а затем в режиме **1С:Предприятие** перепроведем все эти документы.

- **В режиме Конфигуратор**

Откроем окно редактирования объекта конфигурации **Документ ОказаниеУслуги**.

- На закладке **Движения**: укажем, что этот документ будет создавать движения еще и по

регистру **Продажи**.

- На закладке **Прочее**: нажмем кнопку **Модуль объекта**, затем откроем процедуру обработчика события **ОбработкаПроведения** и изменим код следующим образом (см. Листинг 12.1):

Листинг 12.1

Процедура ОбработкаПроведения(Отказ, Режим)

Движения.ОстаткиМатериалов.Записывать = Истина;
Движения.СтоимостьМатериалов.Записывать = Истина;
Движения.Продажи.Записывать = Истина;

Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда

//регистр ОстаткиМатериалов Расход
Движение = Движения.ОстаткиМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Склад=Склад;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;

//регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Стоимость=ТекСтрокаПереченьНоменклатуры.Количество
**ТекСтрокаПереченьНоменклатуры.Стоимость;*
КонецЕсли;

//РегистрПродажи

Движение = Движения.Продажи.Добавить();
Движение.Период = Дата;
Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Клиент = Клиент;
Движение.Мастер – Мастер;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
Движение.Выручка = ТекСтрокаПереченьНоменклатуры.Сумма;
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Стоимость
**ТекСтрокаПереченьНоменклатуры.Количество;*

КонецЦикла;

КонецПроцедуры

Отредактируем командный интерфейс формы документа, чтобы в панели навигации формы иметь возможность переходить к списку записей регистра **Продажи**, связанному с документом:

- Откроем форму документа **ОказаниеУслуги**: перейдем на закладку **Командный**

интерфейс, раскроем группу **Перейти** и увидим команду для открытия регистра накопления **Продажи**. Установим свойство *Видимость* для этой команды.

- **В режиме 1С: Предприятие**

В режиме 1С: Предприятие перепроведем все документы для этого запустим 1С: Предприятие в режиме отладки и откроем по очереди каждый документ **Оказание услуги**.

Контрольные вопросы

- *Что такое оборотный регистр накопления.*
- *В чем отличие между регистром накопления остатков и оборотным регистром накопления.*
- *Как выбирать реквизиты и измерения при создании регистров накопления.*
- *Как создать оборотный регистр накопления.*

Лабораторная работа № 13 Отчеты. Выбор данных из одной таблицы

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Способы доступа к данным

Система 1С:Предприятие 8 поддерживает два способа доступа к данным, хранящимся в базе данных:

- **объектный** (для чтения и записи),
- **табличный** (для чтения).

Объектный способ доступа к данным реализован посредством использования объектов встроенного языка.

Важной особенностью объектного способа доступа к данным является то, что, обращаясь к какому-либо объекту встроенного языка, мы обращаемся к некоторой совокупности данных, находящихся в базе данных, как к единому целому.

Объектная техника обеспечивает сохранение целостности объектов, кеширование объектов, вызов соответствующих обработчиков событий и т. д.

Табличный доступ к данным в 1С: Предприятии 8 реализован с помощью запросов к базе данных, которые составляются на языке запросов.

В этой технике разработчик получает возможность оперировать отдельными полями таблиц базы данных, в которых хранятся те или иные данные.

Табличная техника предназначена для получения информации из базы данных по некоторым условиям (отбор, группировка, сортировка, объединение нескольких выборок, расчет итогов и т.д.). Табличная техника оптимизирована для обработки больших объемов информации, расположенной в базе данных, и получения данных, отвечающих заданным критериям.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Выбор данных из одной таблицы

Создадим отчет Реестр документов оказание услуги, используя систему компоновки данных. Этот отчет будет выводить список существующих в базе данных документов ОказаниеУслуги в порядке их дат и номеров.

- **В режиме Конфигуратор**

Добавим в конфигураторе объект конфигурации Отчет.

На закладке *Основные*: имя отчета – РеестрДокументовОказаниеУслуги; расширенное представление - Список оказанных услуг для представления отчета в интерфейсе программы. Создадим схему компоновки данных для отчета, нажмем кнопку Открыть схему компоновки данных. В открывшемся диалоговом окне конструктора макета нажмем Готово. В конструкторе схемы компоновки данных создадим Набор данных – запрос.

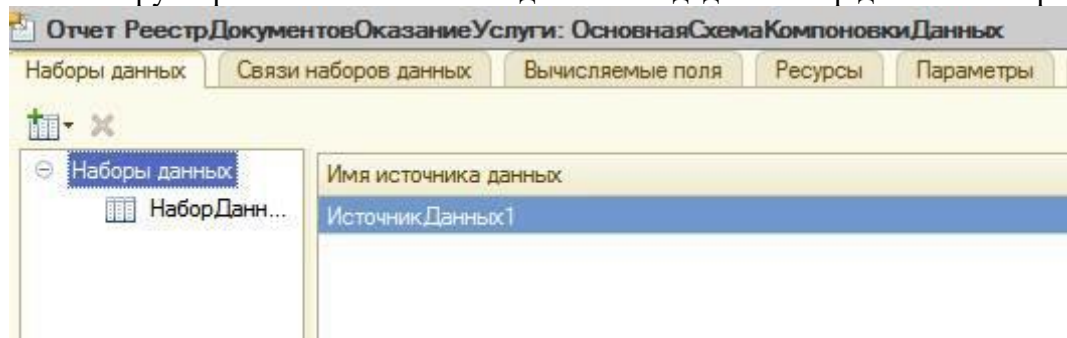


Рис. 13.1. Проведение приходной накладной по двум регистрам

После чего нажмем кнопку Конструктор запроса, в качестве источника данных для запроса выберем объектную таблицу документа ОказаниеУслуги. Из этой таблицы выберем следующие поля:

- Склад,
- Мастер,
- Клиент,
- Ссылка.

На закладке *Объединения/Псевдонимы* укажем, что поле *Ссылка* будет иметь псевдоним *Документ*. На закладке *Порядок* укажем, что результат запроса должен быть упорядочен по значению поля *Документ*.

На закладке *Настройки* создадим стандартные настройки, определяющие, как будет выводиться информация в отчет.

Иерархическая структура отчета может содержать в различных сочетаниях три основных элемента:

- Группировка - для вывода информации в виде обычного линейного отчета.
- Таблица - для вывода информации в виде таблицы.
- Диаграмма - для вывода информации в виде диаграммы.

Для добавления нового элемента, в нашем случае группировки, выделим в дереве структуры отчета корневой элемент Отчет и вызовем его контекстное меню. В окне выбора поля группировки просто нажмем ОК. В структуре отчета появится группировка Детальные записи. На закладке *Выбранные поля* перенесем мышью из списка доступных

полей те поля, которые будут выводиться в отчет:

- Документ,
- Склад,
- Мастер,
- Клиент.

В результате окно настроек отчета должно иметь вид. На этом создание отчета закончено.

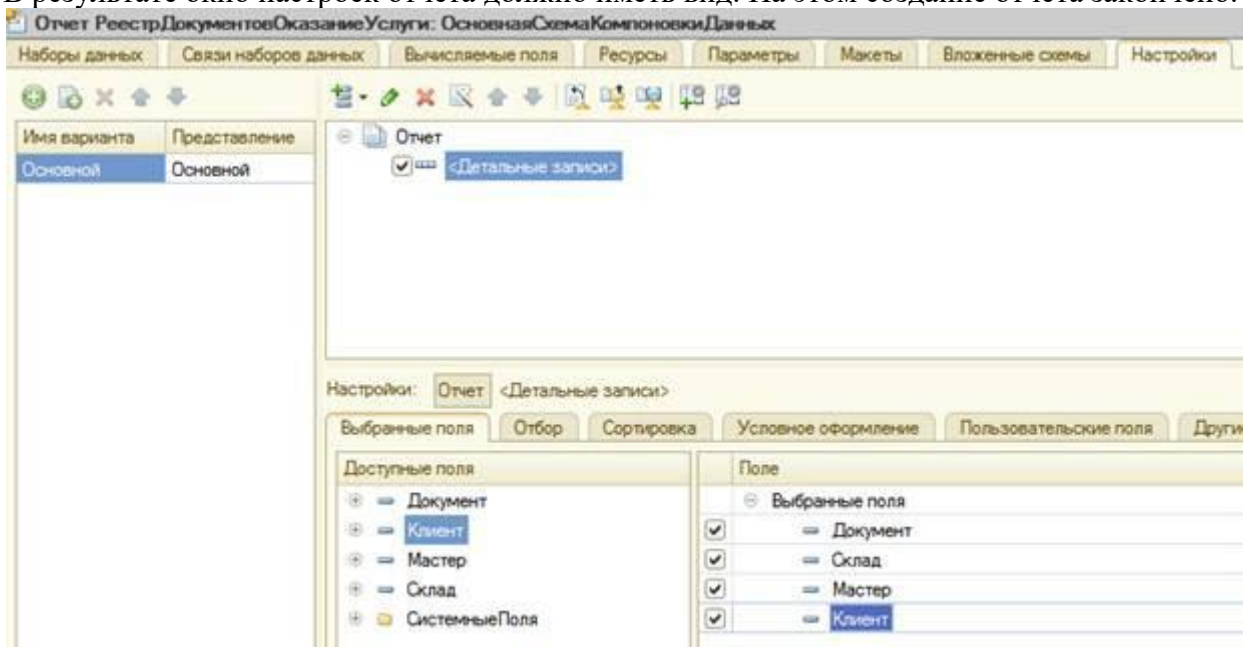


Рис. 13.2. Создание отчета

- В окне редактирования объекта конфигурации Отчет РеестрДокументовОказаниеУслуги перейдем на закладку Подсистемы.

Отметим в списке подсистему Оказание услуг.

- **В режиме 1С: Предприятие**

Запустим 1С: Предприятие в режиме отладки.

В открывшемся окне 1С: Предприятия можно наблюдать, что в панели действий раздела Оказание услуг в группе команд для выполнения отчетов появилась команда для формирования отчета Реестр документов оказание услуги.

Причем если подвести к ней мышью, то появится всплывающая подсказка Список оказанных услуг, которая определяется свойством Расширенное представление.

- Выполним команду Реестр документов оказание услуги.

Перед нами откроется форма отчета, автоматически сформированная системой.

- Нажмем кнопку Сформировать.

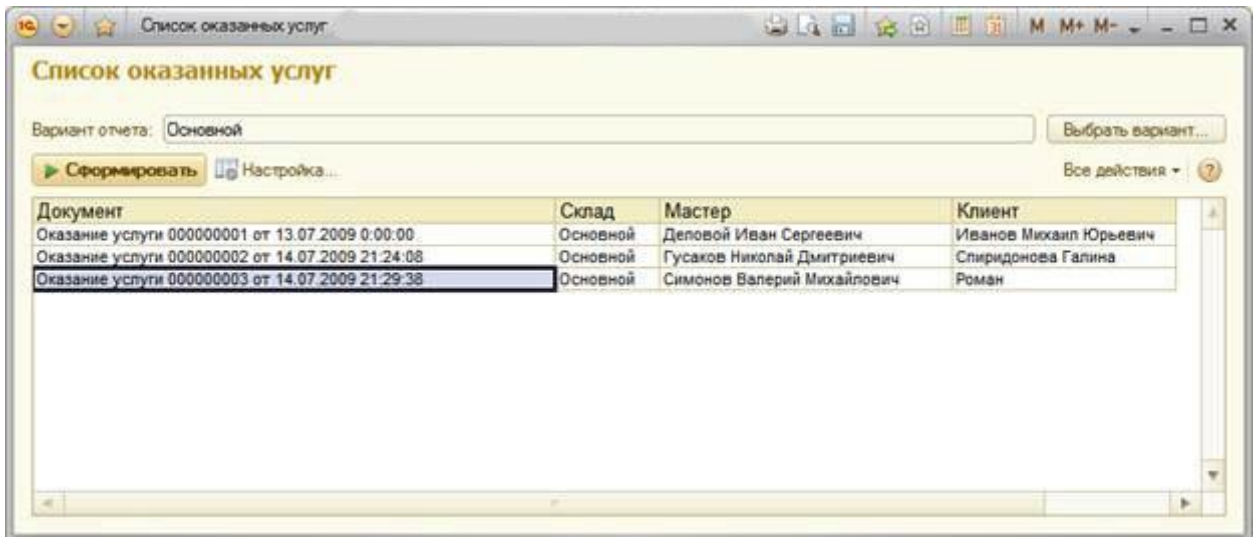


Рис. 13.3. Форма отчета, автоматически сформированная системой

Перед вами появится отчет, содержащий реестр документов Оказание услуги.

Двойным щелчком мыши на поле Документ можно открыть исходный документ, а также выполнить другие действия «расшифровки», которые предоставляет система компоновки данных.

Контрольные вопросы

- Для чего предназначен объект встроенного языка Запрос.
- Для чего предназначена система компоновки данных.
- Для чего предназначена схема компоновки данных.
- Для чего предназначены настройки компоновки данных.
- В чем отличие между реальными и виртуальными таблицами.

Лабораторная работа № 14 Отчеты. Выбор данных из двух таблиц

Работа с запросами

Для работы с запросами используется объект встроенного языка Запрос. Он позволяет получать информацию, хранящуюся в полях базы данных, в виде выборки, сформированной по заданным правилам.

Выбор данных из двух таблиц

Создадим Отчет - Рейтинг услуг, который будет содержать информацию о том, выполнение каких услуг принесло компании наибольшую прибыль в указанном промежутке времени.

- **В режиме Конфигуратор**
- Добавим в конфигураторе объект конфигурации Отчет.
- На закладке *Основные*: имя отчета – РейтингУслуг. Создадим схему компоновки данных для отчета, нажмем кнопку Открыть схему компоновки данных. В открывшемся диалоговом окне конструктора макета нажмем Готово. В конструкторе схемы компоновки данных создадим Набор данных – запрос. После чего нажмем кнопку Конструктор запроса, в качестве источника данных для запроса выберем объектную таблицу Номенклатура и виртуальную таблицу регистра накопления Продажи.Обороты.
- Чтобы исключить неоднозначность имен в запросе, необходимо переименовать таблицу *Номенклатура* в *спрНоменклатура*: выделим *Номенклатура* в списке *Таблицы*, щелкнем правой кнопкой мыши - *Переименовать таблицу*.

- На закладке *Связи*: снимим флажок Все у таблицы ПродажиОбороты и установим его у таблицы *спрНоменклатура*.

- На закладке *Условия*: раскроем таблицу *спрНоменклатура*, перетащим мышью поле *ЭтоГруппа* в список условий, установим флажок Произвольное и напишем в поле *Условие* следующий код:

Листинг 13.1. Условие запроса:

спрНоменклатураЭтоГруппа=ЛОЖЬ

- На закладке *Объединения/Псевдонимы*: укажем псевдоним представления элемента справочника как *Услуга*, а псевдоним поля регистра как *Выручка*.

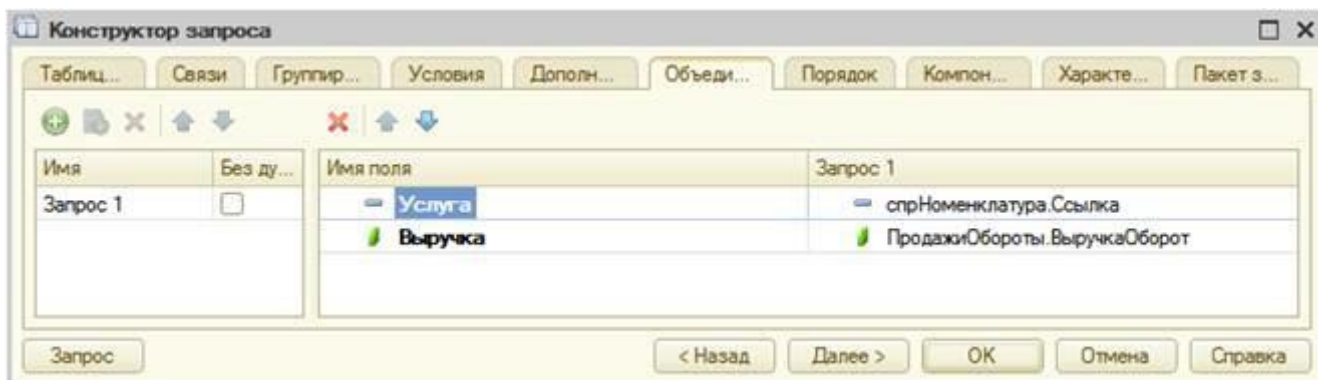


Рис. 13.4. Установка псевдонимов полей запроса

- На закладке *Порядок*: укажем сортировку значение поля Выручка по убыванию:

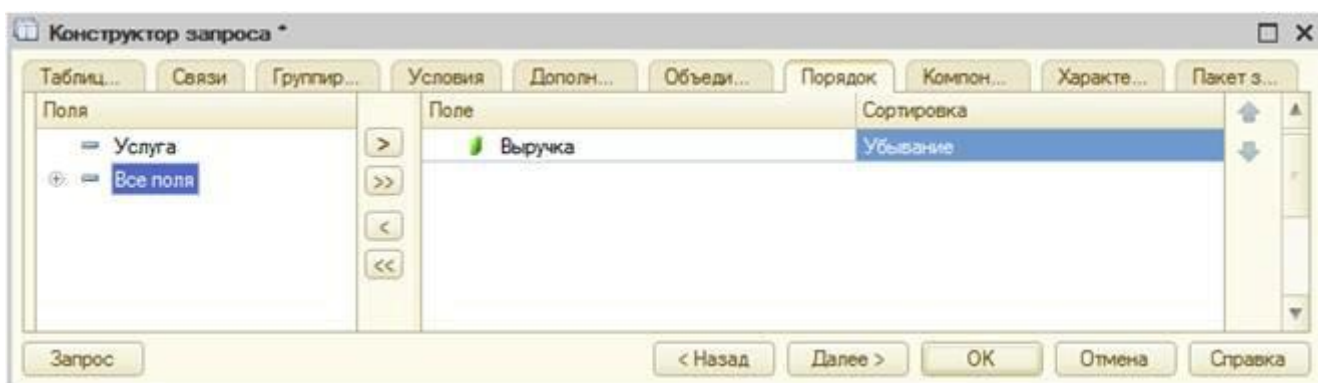


Рис. 13.5. Порядок записей запроса

После чего закончим создание запроса нажатием кнопки ОК.
Вернемся в конструктор схемы компоновки данных.

Ресурсы

Под ресурсами в системе компоновки данных подразумеваются поля, значения которых рассчитываются на основании детальных записей, входящих в группировку. По сути, ресурсы являются групповыми или общими итогами отчета.

- На закладке *Ресурсы*: нажмем кнопку », чтобы конструктор выбрал все доступные ресурсы, по которым можно вычислять итоги. В нашем случае это единственный ресурс Выручка. Платформа автоматически предложит рассчитывать сумму значений этого поля.

Параметры

Пользователя, как правило, интересуют данные о хозяйственной деятельности за определенный период. Поэтому практически в любом отчете используются параметры, задающие начало и конец отчетного периода.

- На закладке *Параметры*: на этой закладке увидим три параметра: НачалоПериода, КонецПериода и ВидНоменклатуры. Вы можете спросить: почему параметра три, хотя в запросе мы задавали всего один - ВидНоменклатуры?

- Дважды щелкнем в ячейке Тип, соответствующей параметру НачалоПериода.

Затем нажмем кнопку выбора установим Состав даты в значение Дата.

- Нажмем *OK*–

- Для параметра КонецПериода установим флажок Ограничение доступности. В ячейке Выражение напишем следующий текст:

КонецПериода(&ДатаОкончания,"День")

- Добавим новый параметр с именем ДатаОкончания. Зададим тип значения параметра - Дата, укажем состав даты - Дата.
- в ячейке Значение, соответствующей параметру ВидНоменклатуры. Воспользуемся кнопкой выбора и выберем значение Перечисление.ВидыНоменклатуры.Услуга.

Настройки

Перейдем к формированию структуры отчета.

- На закладке *Настройки*: добавим группировку и не укажем поле группировки. На закладке Выбранные поля укажем поля Услуга и Выручка.

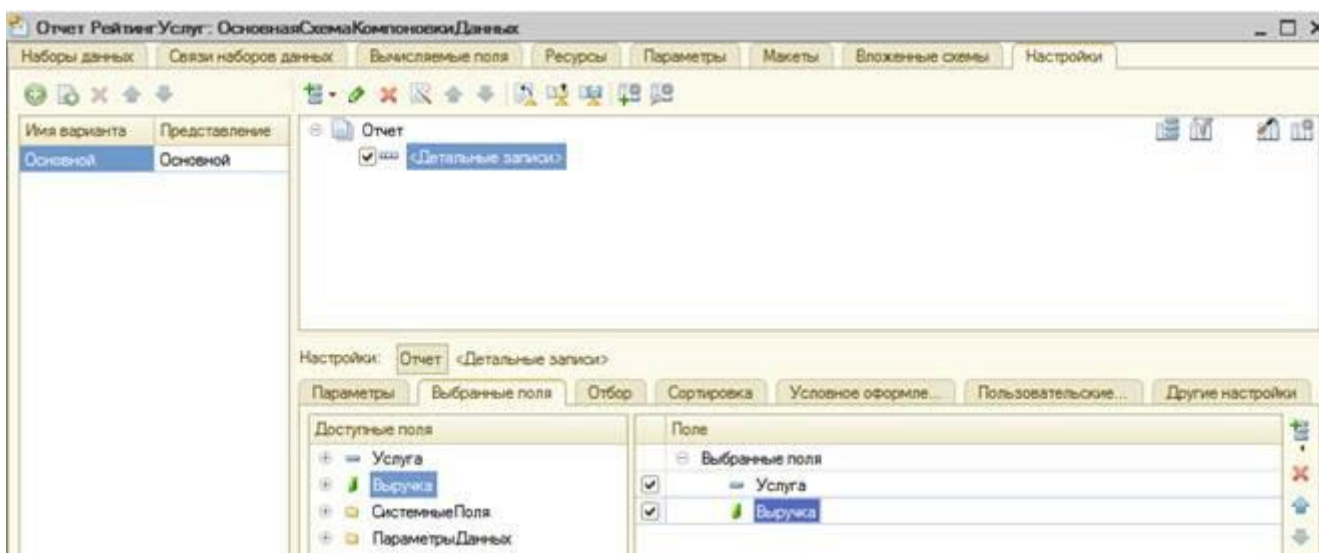


Рис. 13.6. Структура отчета "Рейтинг Услуг"

- Затем перейдем на подзакладку *Другие настройки* и зададим заголовок отчета - Рейтинг услуг:

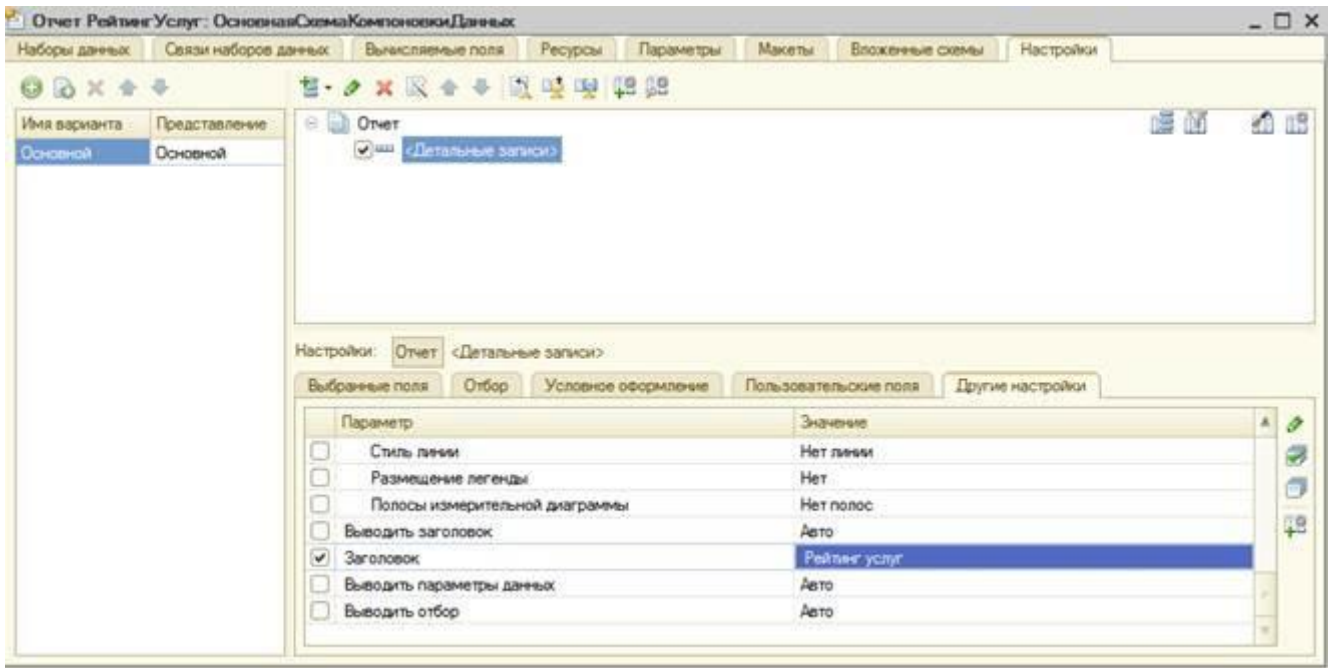


Рис. 13.7. Установка заголовка отчета

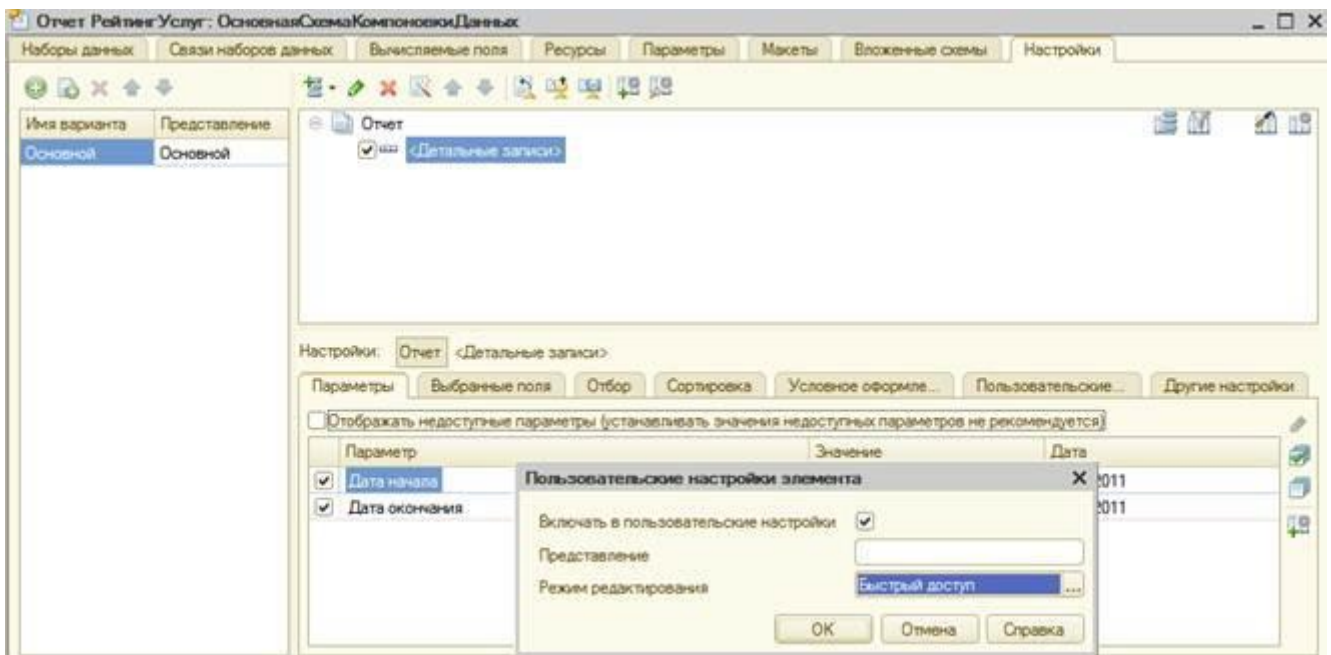


Рис. 13.8. Определение пользовательских настроек

- На подзакладке Параметры выделим по очереди каждый из параметров и нажмем кнопку Свойства элемента пользовательских настроек, расположенную в правом нижнем углу окна настроек. Установим флажок Включать в пользовательские настройки и оставим предложенное по умолчанию для свойства Режим редактирования значение Быстрый доступ.

- Закроем конструктор схемы компоновки данных и в окне редактирования объекта конфигурации Отчет Рейтинг Услуг перейдем на закладку Подсистемы. Отметим в списке подсистем: Оказание услуг и Бухгалтерия.

- **В режиме 1С:Предприятие**

Запустим 1С: Предприятие в режиме отладки и посмотрим, как работает отчет.

В открывшемся окне 1С:Предприятия видим, что в панели действий разделов Оказание услуг и Бухгалтерия в группе команд для выполнения отчетов появилась команда для формирования отчета Рейтинг услуг (рис. 13.40).

Выполним эту команду. Откроется форма отчета, автоматически сформированная системой. В окне отчета мы видим параметры, определяющие отчетный период. Он по умолчанию задан - с начала месяца по сегодняшнее число. Но можно при желании изменить его, воспользовавшись кнопкой календаря. Нажмем кнопку Сформировать:

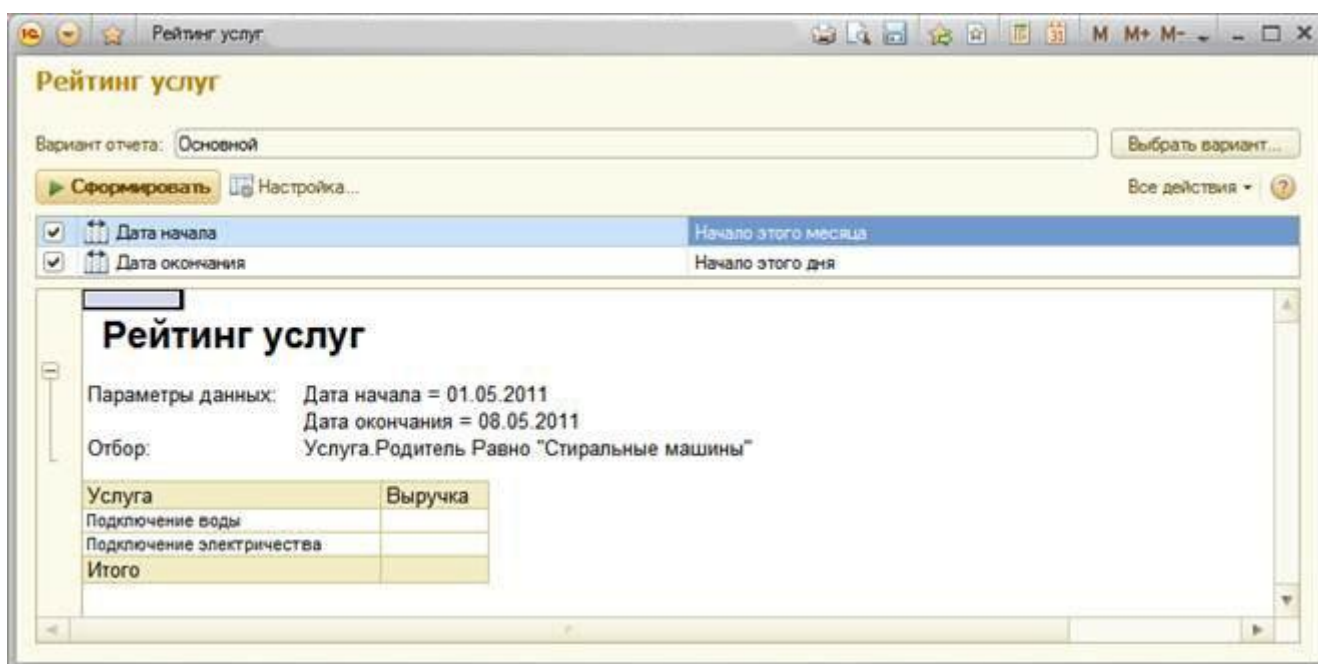


Рис. 13.9. Быстрые и обычные пользовательские настройки

Настройки в конфигураторе и в режиме 1С:Предприятие

На примере предыдущего отчета изучим другие настройки отчета - Условное оформление и Отбор.

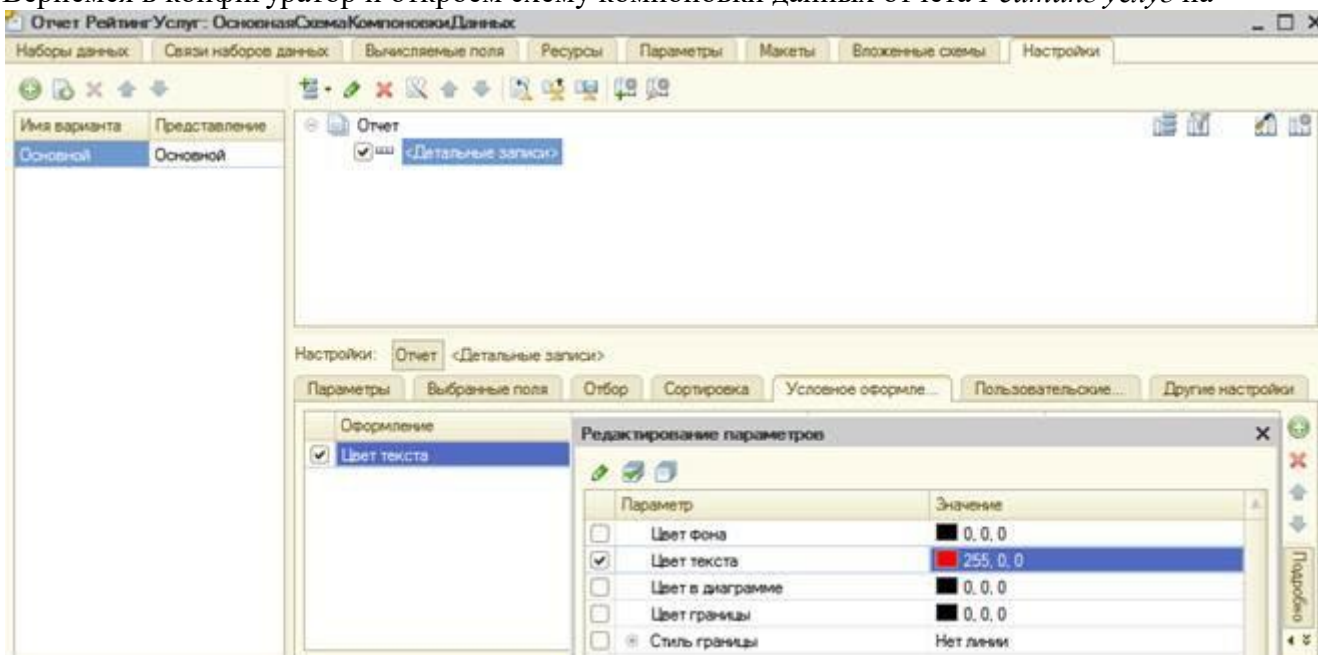
Настройки можно задать как в режиме Конфигуратор, так и в режиме 1С:Предприятие по команде Все действия > Изменить вариант. Разница в том, что те настройки, которые будут изменяться в конфигураторе, называются стандартными настройками и будут сохранены в самой схеме компоновки данных, то есть любой пользователь конфигурации будет видеть отчет именно в таком виде, как вы его настроите в конфигураторе. А в режиме 1С:Предприятие, настройка уже не будет являться частью конфигурации и будет доступна только одному конкретному пользователю конкретной информационной базы.

Условное оформление

В таком отчете, как Рейтинг услуг, было бы удобно выделять цветом записи отчета, содержащие услуги с наименьшей или с наибольшей выручкой, или еще по какому-либо условию.

- В режиме Конфигуратор

Вернемся в конфигуратор и откроем схему компоновки данных отчета *Рейтинг услуг* на



закладку *Настройки*: перейдем на закладку *Условное оформление* и нажмем кнопку *Добавить*. В пункте *Оформление*, нажмем кнопку выбора, после чего в поле установим нужный цвет текста.

Рис. 13.10. Настройка условного оформления

- Нажмем *ОК* -

Затем укажем *Условие*: нажмем кнопку выбора в поле *Условие* и в появившемся окне нажмем кнопку *Добавить* и укажем в графе *Левое значение поле* - *Выручка*, в графе *Вид сравнения* - *Меньше*, а в графе *Правое значение* - *700*.

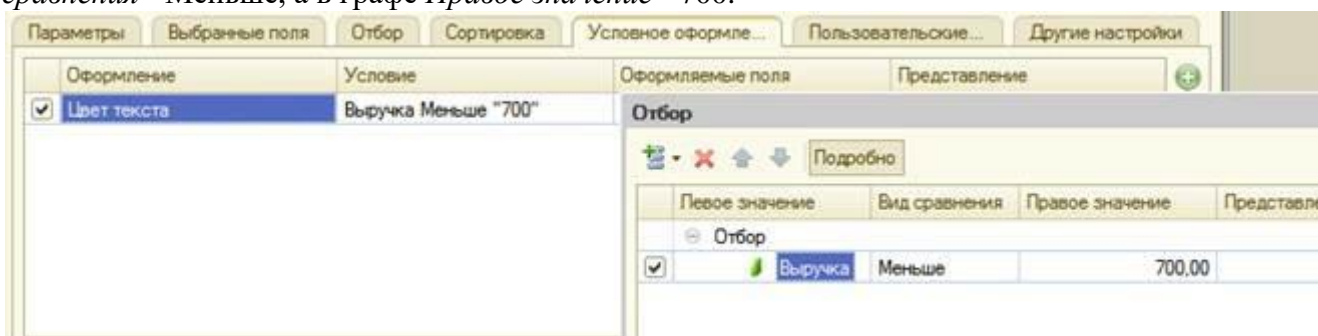


Рис. 13.11. Настройка условного оформления

- Нажмем *ОК* -

В поле *Оформляемые поля*: и в появившемся окне, нажимая кнопку, *Добавить*, можно выбрать поля *Услуга* и *Выручка*.

- Нажмем *ОК* -

В поле *Представление*: установим как *Непопулярная услуга* (это то, что увидит пользователь в своих настройках).

Нажмем кнопку *Свойства* элемента пользовательских настроек, расположенную в правом нижнем углу окна настроек (см. рис. 13.12). Установим флажок *Включать* в

пользовательские настройки и установим свойство *Режим редактирования* - Обычный.

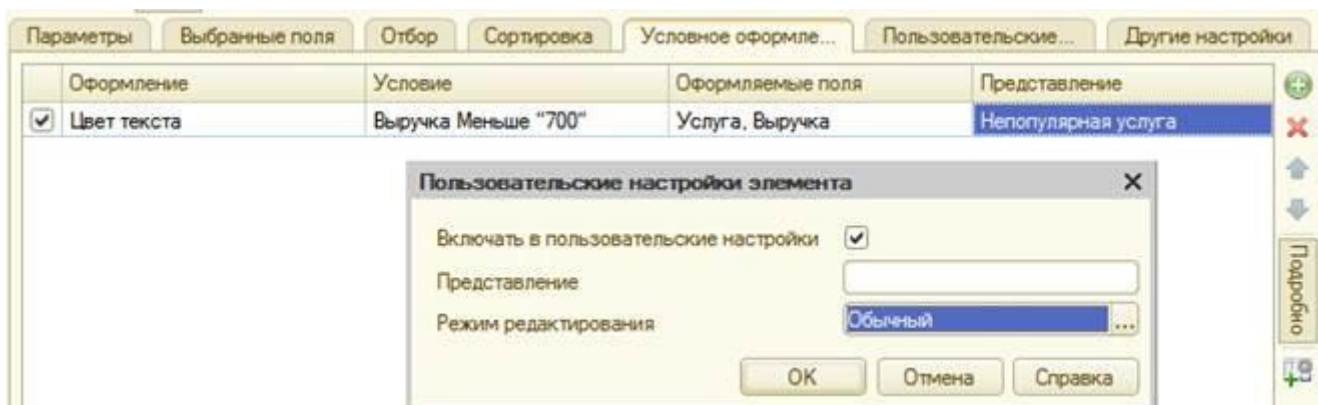


Рис. 13.12. Настройка условного оформления

- **В режиме 1С:Предприятие**
 - Вызовем отчет;
 - Зададим Дату окончания отчетного периода как Начало этого дня и нажмем кнопку Сформировать;
 - Нажмем кнопку Настройка - перед вами появится окно пользовательских настроек отчета, содержащее параметры отчетного периода и настройку условного оформления Непопулярная услуга. Можно снять флажок использования этой настройки, нажать кнопку Завершить редактирование и снова выполнить отчет.

Пользовательские настройки

- **В режиме Конфигуратор**

На закладке Настройки схемы компоновки данных, отчета Рейтинг услуг, содержатся полные настройки отчета, которые задает разработчик. Часть из них может быть представлена пользователю для создания произвольного отбора, условного оформления отчета и пр.

- Нажмем кнопку *Свойства* элемента пользовательских настроек, расположенную вверху в командной панели окна настроек. Установим признак использования для настроек Отбор и Условное оформление и установим для них свойство Режим редактирования в значение Обычный.

Тем самым включились настройки отбора и условного оформления в состав пользовательских настроек и предоставили пользователю возможность задавать их в отдельном окне, вызываемом кнопкой Настройка.

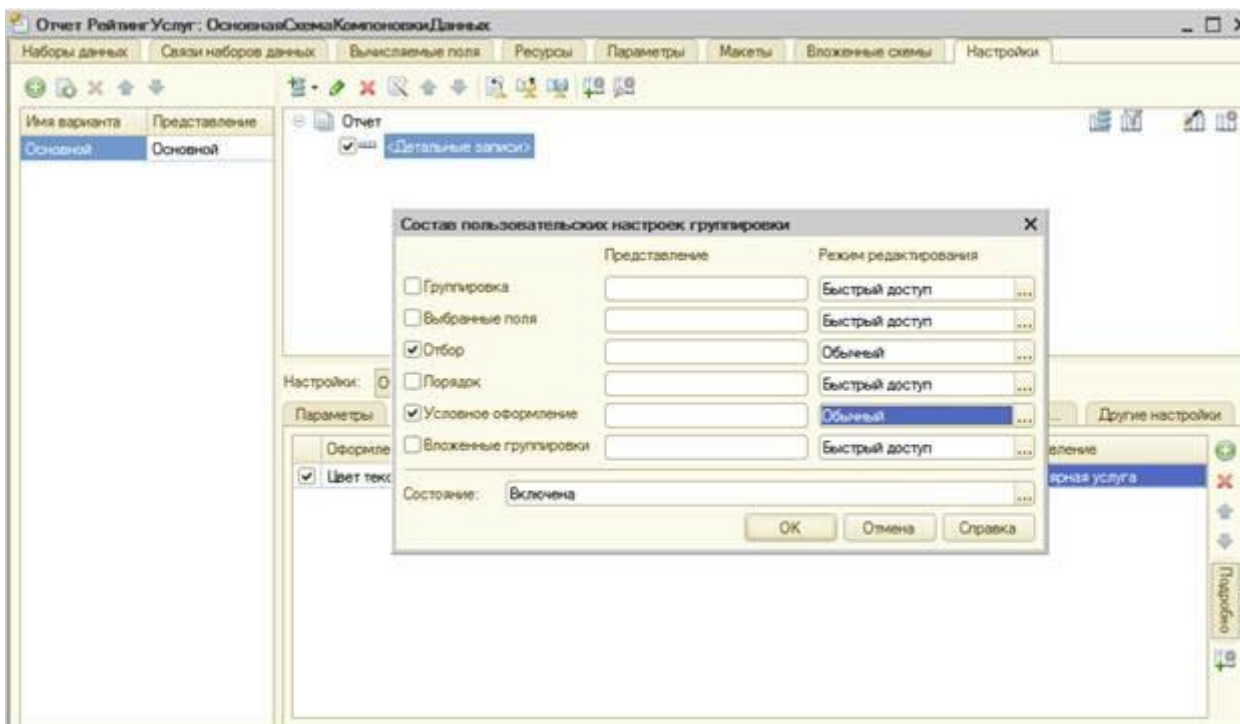


Рис. 13.13. Состав пользовательских настроек

Контрольные вопросы

- Из каких частей состоит текст запроса, какие из них являются обязательными.
- Каковы основные синтаксические конструкции языка запросов.
- Что является источником данных запроса.
- Что такое псевдонимы в языке запросов.
- Что такое параметры запроса.

Лабораторная работа № 15

Отчеты. Вывод данных по дням в выбранном периоде

Источники данных запросов

Исходную информацию запрос получает из набора таблиц. Эти таблицы представляют разработчику данные реальных таблиц базы данных в удобном для анализа виде.

Все таблицы, которыми оперирует язык запросов, можно разделить на две большие группы: реальные таблицы и виртуальные таблицы.

Отличительной особенностью реальных таблиц является то, что они содержат данные какой-либо одной реальной таблицы, хранящейся в базе данных.

Виртуальные таблицы формируются в основном из данных нескольких таблиц базы данных.

Общим для всех виртуальных таблиц является то, что им можно задать ряд параметров, определяющих, какие данные будут включены в эти виртуальные таблицы. Набор таких параметров может быть различным для разных виртуальных таблиц и определяется данными, хранящимися в исходных таблицах базы данных.

Реальные таблицы подразделяются на объектные (ссылочные) и неobjектные (нессылочные).

В объектных таблицах представлена информация ссылочных типов данных (справочники, документы, планы видов характеристик и т.д.). А в неobjектных - всех остальных типов данных (константы, регистры и т. д.).

Отличительной особенностью объектных таблиц является то, что они включают в себя поле Ссылка, содержащее ссылку на текущую запись. Кроме этого, для таких таблиц возможно получение пользовательского представления объекта. Эти таблицы могут быть иерархическими, и поля таких таблиц могут содержать вложенные таблицы.

Диаграмма

Диаграмма предназначена для размещения в таблицах и формах диаграмм и графиков различного вида. Логически диаграмма является совокупностью точек, серий и значений серий в точке.

Как правило, в качестве точек используются моменты или объекты, для которых мы получаем значения характеристик, а в качестве серий - характеристики, значения которых нас интересуют. На пересечении серии и точки находится значение диаграммы.

Например, диаграмма продаж видов номенклатуры по месяцам будет состоять из точек - месяцев, серий - видов номенклатуры и значений - оборотов продаж.

Диаграмма как объект встроенного языка имеет три области, которые позволяют управлять оформлением диаграммы: область построения, область заголовка и область легенды.

Диаграмма может быть вставлена в структуру отчета как отдельный элемент. В следующем варианте настроек отчета ВыручкаМастеров мы будем использовать диаграмму в структуре настроек схемы компоновки данных.

Вывод данных по всем дням в выбранном периоде

Следующий отчет, который мы добавим, будет называться Выручка мастеров. Он будет содержать информацию о том, какая выручка была получена ООО «Фамилия курсанта» благодаря работе каждого из мастеров, с детализацией по всем дням в выбранном периоде и разворотом по клиентам, обслуженным в каждый из дней.

- **В режиме Конфигуратор**

Добавим новый объект конфигурации Отчет. Назовем его ВыручкаМастеров и запустим конструктор схемы компоновки данных. Добавим новый **Набор данных - запрос** и вызовем конструктор запроса. В качестве источника данных для запроса выберем виртуальную таблицу регистра накопления **Продажи.Обороты**.

Запрос для набора данных. Параметры виртуальной таблицы

Зададим один из параметров этой виртуальной таблицы - Периодичность. Для этого перейдем в поле **Таблицы**, выделим таблицу и нажмем кнопку **Параметры виртуальной таблицы** (рис. 15.1). В открывшемся окне параметров зададим значение параметра **Периодичность - День** (рис. 15.2)

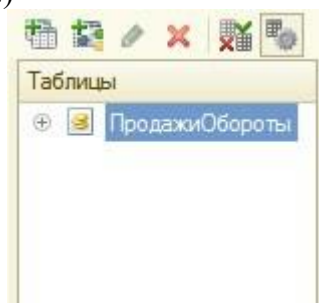


Рис. 15.1. Изменение параметров виртуальной таблицы

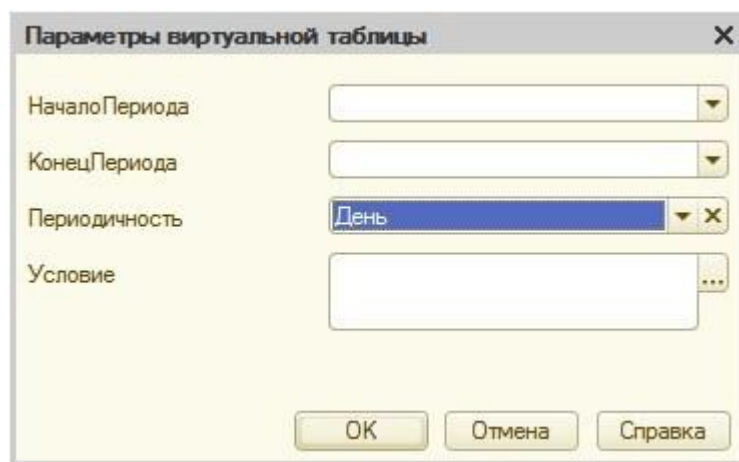


Рис. 15.2. Параметры виртуальной таблицы

Нажмем ОК.

После этого выберем из таблицы следующие поля

- ПродажиОбороты.Мастер,
- ПродажиОбороты.Период,
- ПродажиОбороты.Клиент,
- ПродажиОбороты.ВыручкаОборот.

Теперь перейдем на закладку **Объединения/Псевдонимы** и зададим псевдоним **Выручка** для поля **ПродажиОбороты.ВыручкаОборот** (рис. 15.3)

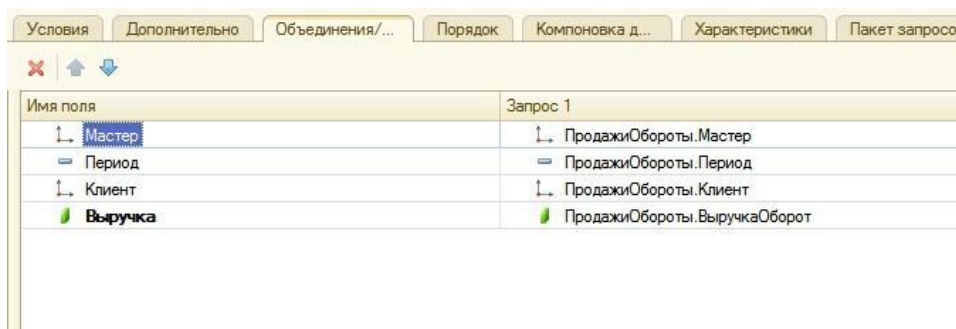


Рис. 15.3. Объединения/Псевдонимы

Ресурсы

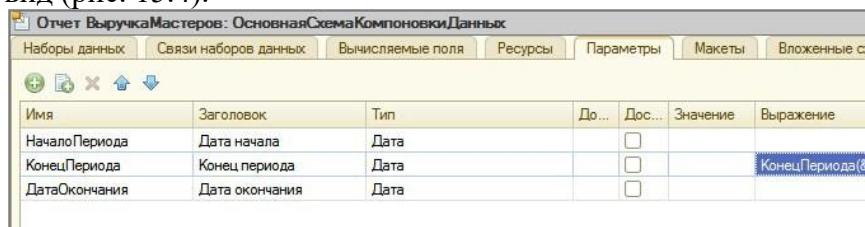
Теперь перейдем к редактированию схемы компоновки данных. На закладке **Ресурсы** нажмем кнопку >> и убедимся, что конструктор выбрал единственный имеющийся у нас ресурс - **Выручка**.

Параметры

На закладке **Параметры** выполним те же действия, что и при создании предыдущего отчета. Для параметра **НачалоПериода** зададим заголовок **Дата начала**. В поле Тип зададим состав даты - **Дата**. Затем добавим еще один параметр - **ДатаОкончания**, установим его тип как **Дата**, состав даты - **Дата**. Для параметра **КонецПериода** зададим выражение (листинг 15.1) и в поле **Ограничение доступности** установим флажок ограничения доступности.

Листинг 15.1. Выражение для расчета значения параметра «КонецПериода»
КонецПериода(&ДатаОкончания,"День")

В результате перечисленных действий параметры компоновки данных будут иметь следующий вид (рис. 15.4).



Имя	Заголовок	Тип	До...	Дос...	Значение	Выражение
НачалоПериода	Дата начала	Дата		<input type="checkbox"/>		
КонецПериода	Конец периода	Дата		<input type="checkbox"/>		КонецПериода(&ДатаОкончания(&ДатаОкончания,"День"))
ДатаОкончания	Дата окончания	Дата		<input type="checkbox"/>		

Рис. 15.4. Параметры компоновки данных

Настройки

Теперь создадим структуру отчета. На закладке **Настройки** последовательно создадим две вложенные группировки:

- верхнего уровня - по полю **Мастер**;
- вложенная в нее - по полю **Период**.

Для этого сначала выделим корневой элемент **Отчет** в структуре отчета, нажмем кнопку **Добавить** в командной панели окна настроек, добавим новую группировку и укажем поле группировки **Мастер** (рис. 15.5).

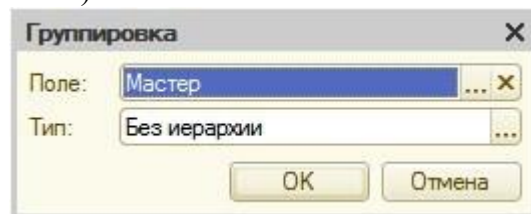


Рис. 15.5. Поле группировки

> Затем добавим в группировку **Мастер** вложенную группировку по полю **Период**. Для этого выделим группировку **Мастер**, нажмем кнопку **Добавить**, добавим новую группировку и укажем поле группировки **Период**. Затем добавим еще одну группировку, вложенную в группировку по полю **Период**, - **Детальные записи** (без указания

группировочного поля).

Для этого выделим группировку **Период**, нажмем кнопку **Добавить** и добавим новую группировку без указания группировочного поля. После этого перейдем на закладку **Выбранные поля** и добавим в список выбранных полей поля **Клиент** и **Выручка**. Поля **Мастер** и **Период** мы не задаем, так как по этим полям производится группировка данных и их значение будет выведено автоматически. В результате структура отчета будет иметь вид (рис. 15.6).

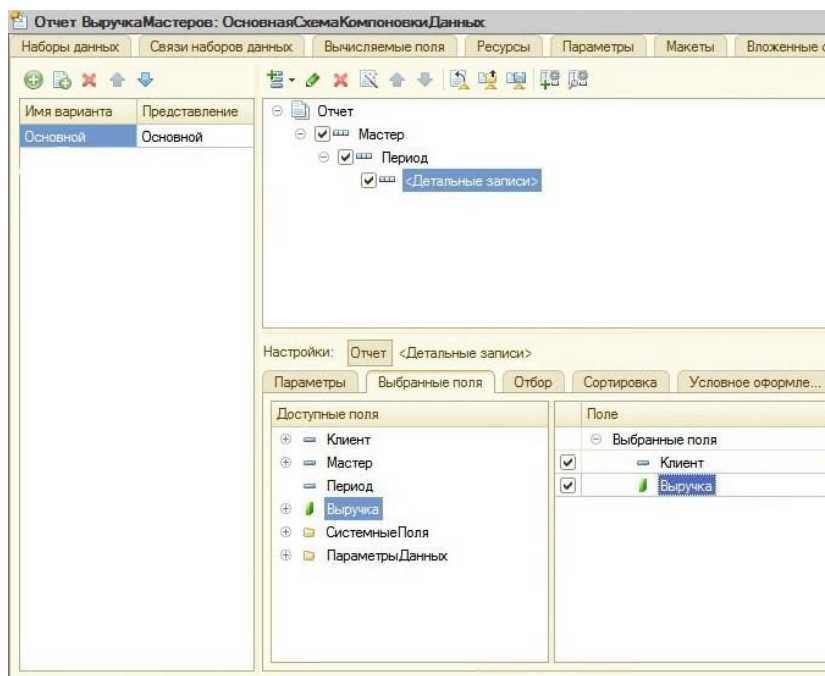


Рис. 15.6. Структура и поля отчета

В заключение перейдем на закладку **Другие настройки** и изменим следующие параметры. Для параметра **Расположение полей группировок** установим значение **Отдельно и только в итогах**.

По умолчанию поля группировок в отчете располагаются вертикально друг под другом. Установка этого свойства в значение **Отдельно и только в итогах** означает, что каждая группировка будет располагаться в отдельной области отчета слева направо и ее наименование будет выводиться только в данной группировке.

Для параметра **Расположение общих итогов по вертикали** зададим значение **Начало**. По умолчанию итоги по вертикали располагаются в конце. Установка этого свойства означает, что общие итоги будут отображаться в начале перед строками группировки. В результате другие настройки отчета примут вид (рис. 15.7).

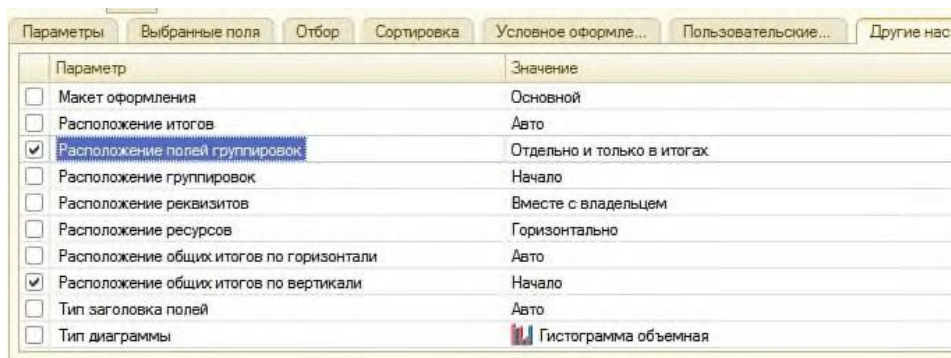


Рис. 15.7. Параметры настроек вывода отчета

Здесь же для параметра **Заголовок** зададим значение **Выручка мастеров**. Затем укажем, что параметры **Дата начала** и **Дата окончания** будут включены в состав пользовательских настроек, и эти настройки будут находиться непосредственно в отчетной форме, то есть будут «быстрыми» настройками. Таким образом, перед формированием отчета пользователь сможет задать отчетный период (рис. 15.8).

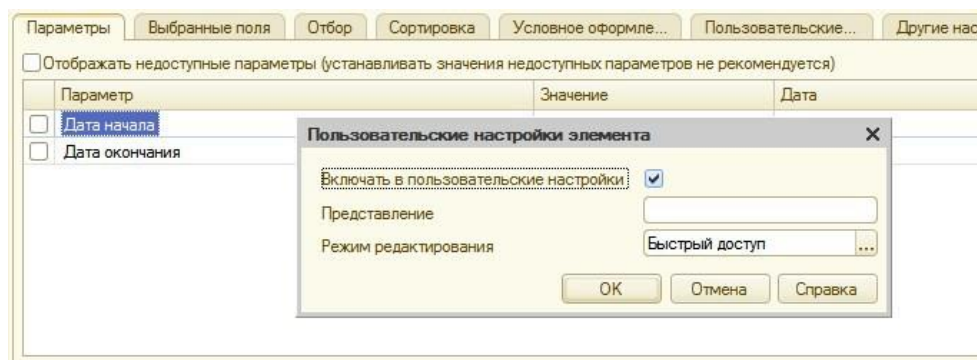


Рис. 15.8. Создание быстрых настроек отчетного периода

В заключение определим, в каких подсистемах будет отображаться наш отчет. Закроем конструктор схемы компоновки данных и в окне редактирования объекта конфигурации Отчет **ВыручкаМастеров** перейдем на закладку **Подсистемы**. Отметим в списке подсистем конфигурации подсистемы **Оказание услуг** и **Расчет зарплаты**. Таким образом, ссылка на наш отчет автоматически попадет в панель действий этих подсистем.

- **В режиме 1С Предприятие**

Запустим 1С:Предприятие в режиме отладки и посмотрим, как работает отчет. В открывшемся окне 1С:Предприятия мы видим, что в панели действий разделов **Оказание услуг** и **Расчет зарплаты** в группе команд для выполнения отчетов появилась команда для формирования отчета **Выручка мастеров**. Выполним эту команду. Зададим отчетный период с 01.07.2009 по 15.07.2009 и сформируем отчет (рис. 15.9).

Выручка мастеров

Вариант отчета: Основной

Сформировать Настройка...

Дата начала: 01.07.2009

Дата окончания: 15.07.2009

Выручка мастеров

Параметры данных: Дата начала = 01.07.2009
Дата окончания = 15.07.2009

Мастер	Период	Клиент	Выручка
Итого			4 794,00
Гусаков Николай Дмитриевич			1 700,00
	01.07.2009 0:00:00		
	02.07.2009 0:00:00		
	03.07.2009 0:00:00		
	04.07.2009 0:00:00		
	05.07.2009 0:00:00		
	06.07.2009 0:00:00		
	07.07.2009 0:00:00		
	08.07.2009 0:00:00		
	09.07.2009 0:00:00		
	10.07.2009 0:00:00		
	11.07.2009 0:00:00		
	12.07.2009 0:00:00		
	13.07.2009 0:00:00		
	14.07.2009 0:00:00		1 700,00
	15.07.2009 0:00:00	Спиридонова Галина	1 700,00
Деловой Иван Сергеевич			950,00
	01.07.2009 0:00:00		
	02.07.2009 0:00:00		
	03.07.2009 0:00:00		
	04.07.2009 0:00:00		
	05.07.2009 0:00:00		
	06.07.2009 0:00:00		
	07.07.2009 0:00:00		

Рис. 15.9. Результат выполнения отчета

Вывод всех дат в выбранном периоде

Если вы помните, в начале раздела мы говорили, что этот отчет должен показывать данные с детализацией по всем дням в выбранном периоде.

У нас же отображаются только те дни, для которых существуют ненулевые записи в таблице регистра накопления **Продажи**.

Для детализации данных в отчете система компоновки данных позволяет указывать для группировок дополнение периодов с заданной периодичностью в указанном интервале. Поэтому сейчас мы изменим настройки отчета таким образом, чтобы в отчет попадала каждая дата из периода, за который сформирован отчет.

- **В режиме Конфигуратор**

Вернемся в режим Конфигуратор и выполним более тонкую настройку структуры отчета. Откроем схему компоновки данных на закладке **Настройки**. До сих пор все настройки структуры, которые мы выполняли, относились ко всему отчету в целом. Но система компоновки данных позволяет настраивать также и каждый элемент структуры в отдельности.

При установке настроек отчета в средней части окна, под деревом структуры отчета, должна быть выделена кнопка, соответствующая режиму настроек. Кнопка **Отчет** - для настройки отчета в целом или кнопка с именем группировки, например **Детальные записи**, если настройки относятся только к ней.

В нашем случае потребуется изменить настройку группировки **Период**.

Для того чтобы перейти к настройкам именно этой группировки, в поле структуры отчета установим курсор на эту группировку, а затем нажмем кнопку **Период** в командной

панели окна. В нижней части окна будут отображены настройки, доступные для данной группировки.

Перейдем на закладку **Поля группировки**. Для поля **Период** установим **Тип дополнения** - **День** (рис. 15.10).

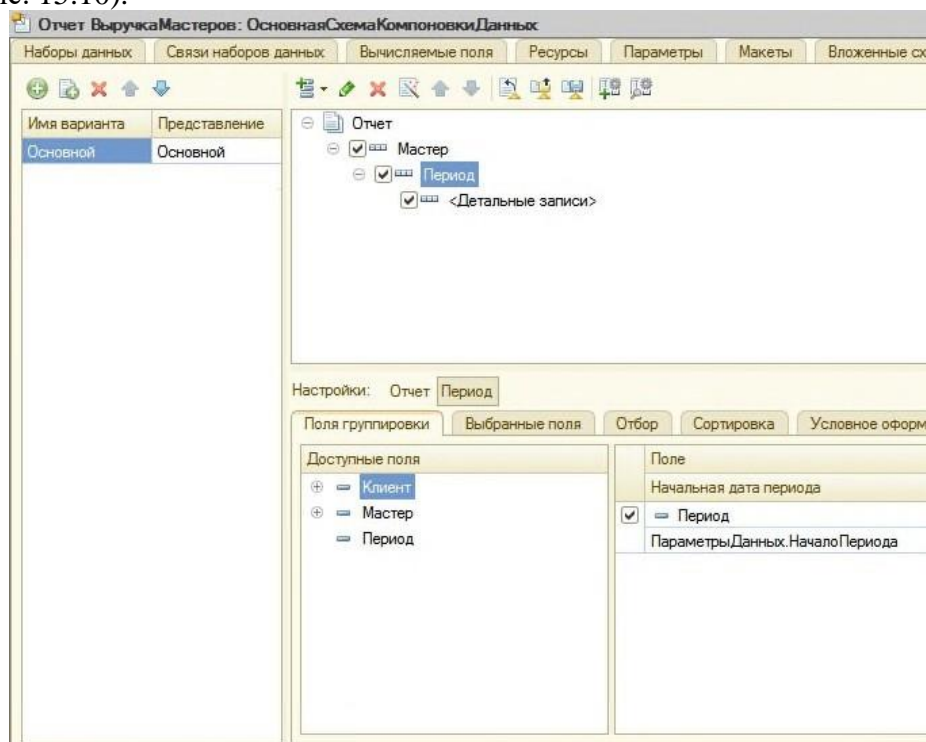


Рис. 15.10. Установка типа дополнения периода

Тем самым мы укажем, что для этой группировки существующие записи с ненулевым значением ресурса будут дополняться записями для каждого из дней.

После этого следует указать, в каком именно периоде будет выполняться такое дополнение.

В поля, расположенные строчкой ниже, можно ввести даты начала и окончания этого периода. Но указание дат в явном виде нас не устраивает, так как пользователь может сформировать отчет за произвольный период. И нам нужно, чтобы дополнение дат выполнялось не в некотором фиксированном периоде, а именно в том периоде, который выбрал пользователь для всего отчета.

Для того чтобы обеспечить именно такую работу отчета, войдем в режим редактирования поля **Начальная дата периода**, дважды кликнув на нем, и нажмем кнопку очистки *.

После этого, нажав кнопку выбора типа данных, мы сможем выбрать тип данных, отображаемых в этом поле.

Выберем **Поле компоновки данных** (рис. 15.11).

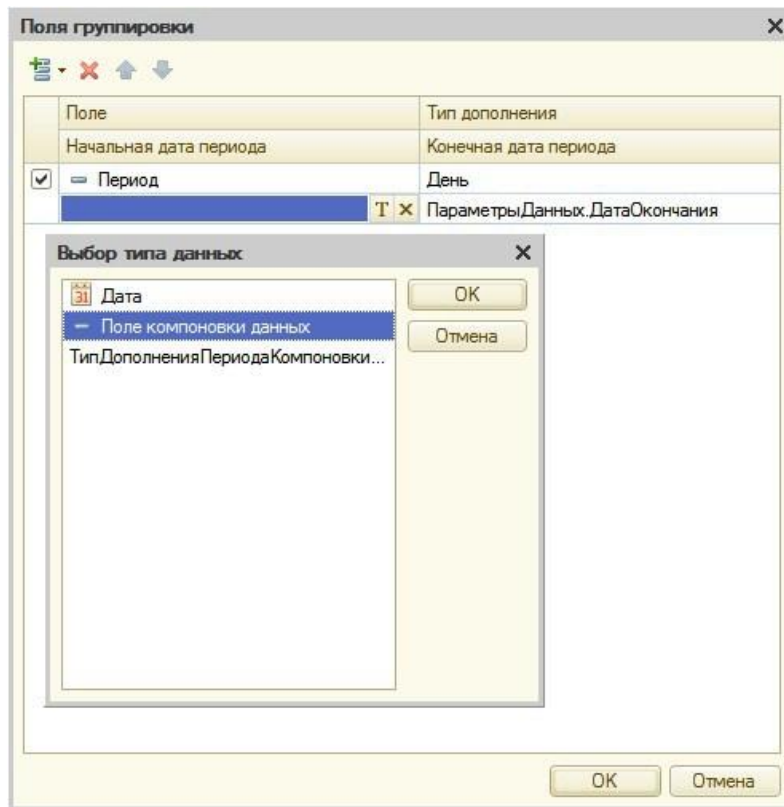


Рис. 15.11. Выбор типа данных

Нажмем ОК.

Теперь нажмем в поле ввода кнопку выбора и в открывшемся окне выбора поля отметим параметр **НачалоПериода** (рис. 15.12). Нажмем ОК.

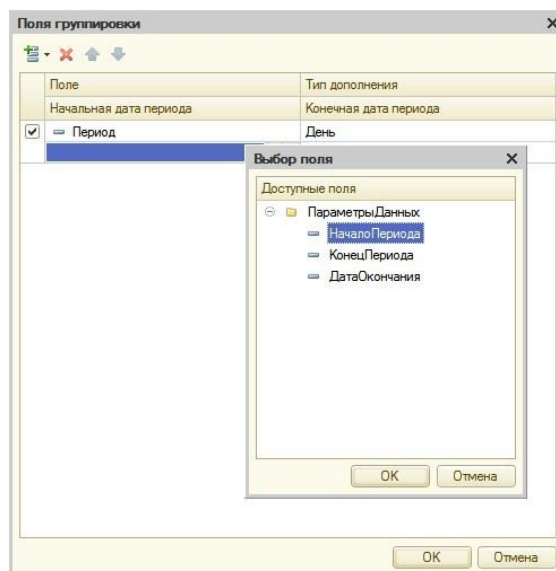


Рис. 15.12. Выбор поля

Для второго поля ввода аналогичным образом укажем, что дата окончания периода будет получена из параметра **ДатаОкончания** (рис. 15.13).

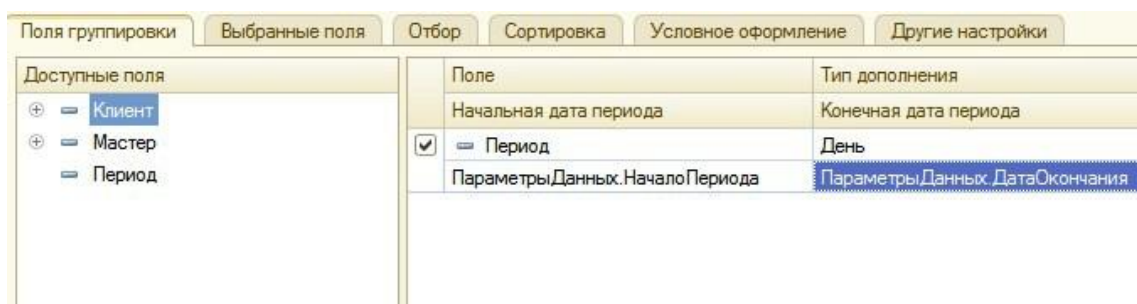


Рис. 15.13. Настройки группировки "Период"

- **В режиме 1С Предприятие**

Запустим 1С:Предприятие в режиме отладки и выполним отчет **Выручка мастеров** за период с 10.07.2009 по 15.07.2009 (рис. 15.14).

Мастер	Период	Клиент	Выручка
Итого			4 794,00
Гусаков Николай Дмитриевич	10.07.2009 0:00:00		
	11.07.2009 0:00:00		
	12.07.2009 0:00:00		
	13.07.2009 0:00:00		
	14.07.2009 0:00:00		1 700,00
	15.07.2009 0:00:00	Спиридонова Галина	1 700,00
Деловой Иван Сергеевич	10.07.2009 0:00:00		
	11.07.2009 0:00:00		
	12.07.2009 0:00:00		
	13.07.2009 0:00:00		950,00
	14.07.2009 0:00:00	Иванов Михаил Юрьевич	950,00
	15.07.2009 0:00:00		
Симонов Валерий Михайлович	10.07.2009 0:00:00		
	11.07.2009 0:00:00		
	12.07.2009 0:00:00		
	13.07.2009 0:00:00		
	14.07.2009 0:00:00		2 144,00
	15.07.2009 0:00:00	Роман	2 144,00

Рис. 15.14. Результат выполнения отчета

Новый вариант отчета

Для анализа работы мастеров за определенный период может понадобиться представить ту же информацию в другом, более наглядном виде.

Например, директору при начислении зарплаты, чтобы понять, какой из мастеров лучше работает, вполне может понадобиться увидеть диаграмму, отражающую вклад каждого мастера в общую выручку предприятия за период.

Поэтому мы создадим другой вариант отчета **ВыручкаМастеров**, представляющий данные в виде диаграммы.

- **В режиме Конфигуратор**

Вернемся в конфигуратор и откроем схему компоновки данных на закладке **Настройки**. В левой части окна находится список вариантов отчета.

При создании настроек отчета в первый раз система компоновки данных по умолчанию создает **Основной** вариант настроек. И мы видим его в списке вариантов нашего отчета. Чтобы добавить новый вариант, нажмем кнопку **Добавить** над этим списком. Зададим имя варианта - **ОбъемВыручки** (рис. 15.15).

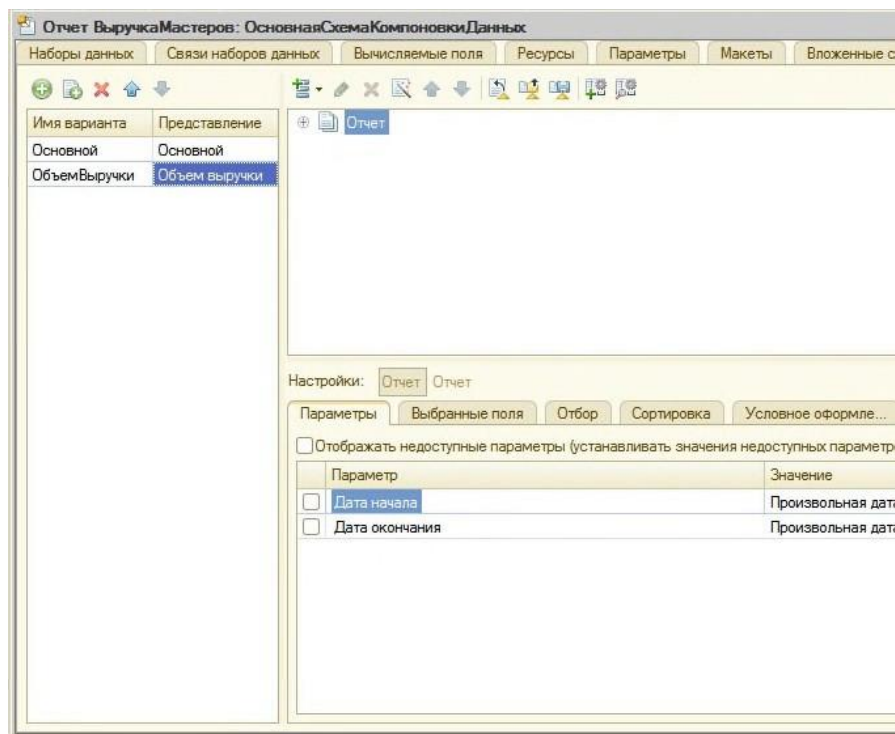


Рис. 15.15. Добавление нового варианта настроек

Мы видим, что структура отчета и все его настройки очистились.

Но они не пропали, а стали невидимы, так как относятся к **Основному** варианту настроек. Если у отчета есть несколько вариантов, то мы видим и можем изменять настройки того варианта, который выделен в данный момент. Причем вся остальная информация в схеме компоновке данных (ресурсы, параметры, наборы данных) осталась без изменений.

Данные для отчета будут получены с помощью того же запроса к базе данных. Изменятся лишь настройки, которые определяют, как будет представлен отчет.

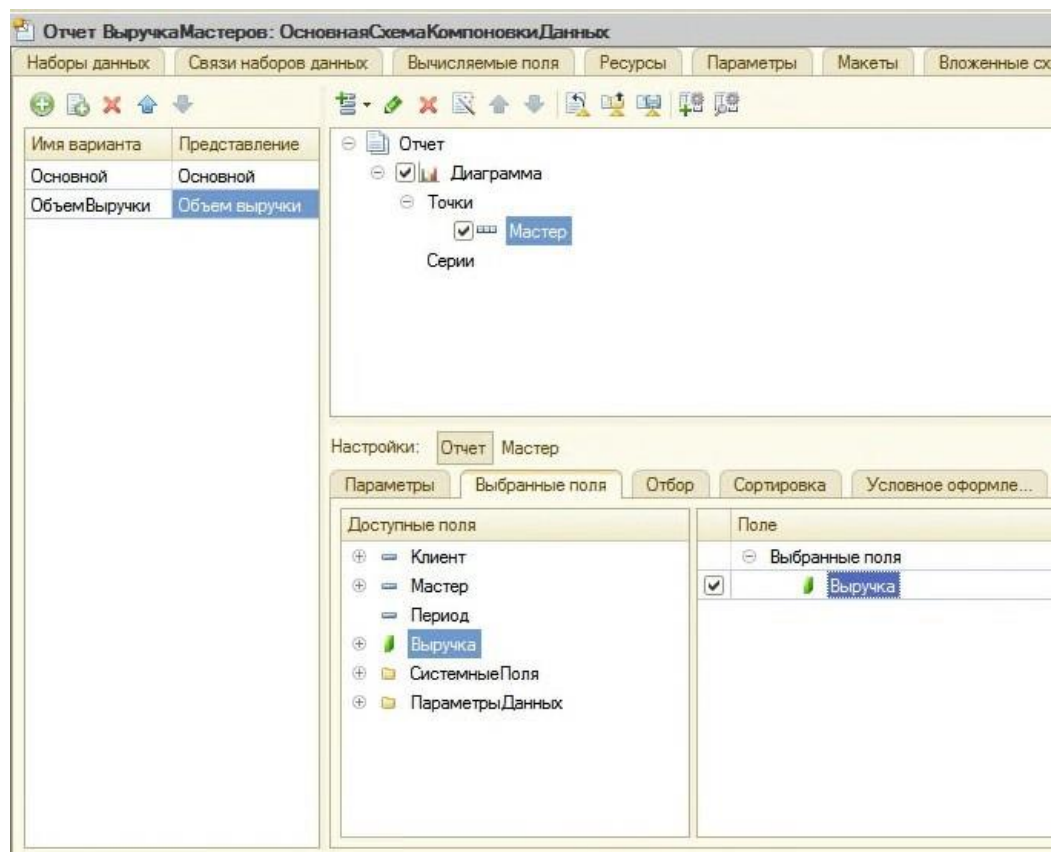
Добавим в структуру отчета диаграмму. Для этого выделим корневой элемент **Отчет**, вызовем его контекстное меню и добавим диаграмму. Затем выделим ветку **Точки** и добавим в нее группировку по полю **Мастер**.

Серии диаграммы оставим без изменений.

Для демонстрации вклада мастеров в общий объем выручки хорошо подойдет измерительная диаграмма, которую мы хотим показать. Для этого вида диаграммы достаточно задать только точки, поэтому серии мы не задаем.

В значения диаграммы всегда выводится один из ресурсов отчета. У нас всего один ресурс - **Выручка** (поле ресурса помечено соответствующей пиктограммой и отличается от обычных полей).

Поэтому перейдем на закладку **Выбранные поля**, перейдем на уровень настроек отчета в целом (нажав кнопку **Отчет**) и выберем поле **Выручка** для вывода в отчет.



Структура отчета должна принять следующий вид (рис. 15.16).

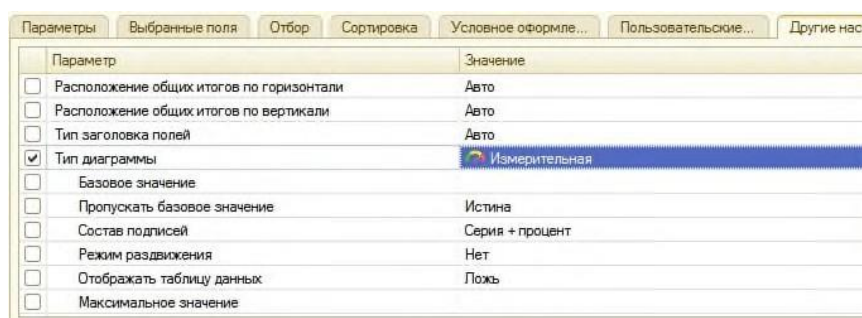


Рис. 15.16. Структура отчета и настройки диаграммы

На закладке **Другие настройки** выберем тип диаграммы - **Измерительная** (рис. 15.17).

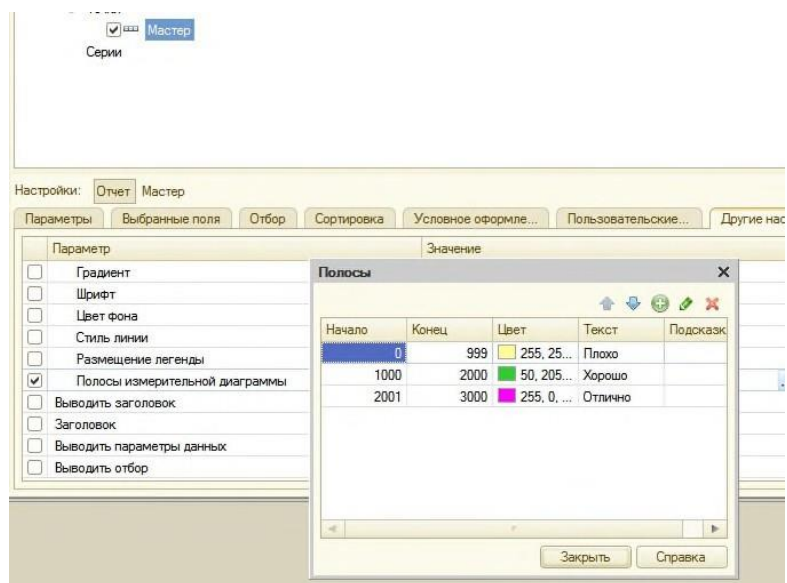


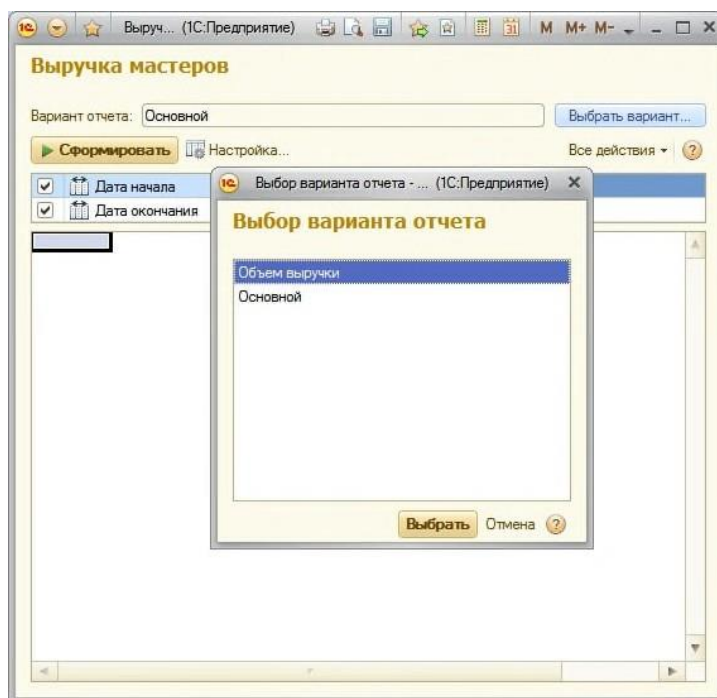
Рис. 15.17. Настройка типа диаграммы

Прокрутив вниз список свойств измерительной диаграммы, зададим ее полосы - **Плохо**, **Хорошо** и **Отлично** (рис. 15.18).

В заключение включим параметры **Дата начала** и **Дата окончания** в состав пользовательских настроек и установим для них **Режим редактирования - Быстрый доступ**.

- **В режиме 1С Предприятие** апустим 1С:Предприятие в режиме отладки и выполним команду **Выручка мастеров** в панели действий раздела **Расчет зарплаты**. В открывшемся окне отчета нажмем кнопку **Выбрать вариант** (рис. 15.19).

Рис. 15.19. Выбор варианта отчета



В окне вариантов отчета мы видим теперь два варианта - **Основной** и только что созданный нами вариант **Объем выручки**. Выделим его и нажмем кнопку **Выбрать**. Зададим отчетный период с 01.07.2009 по 15.07.2009 и сформируем отчет (рис. 15.20).

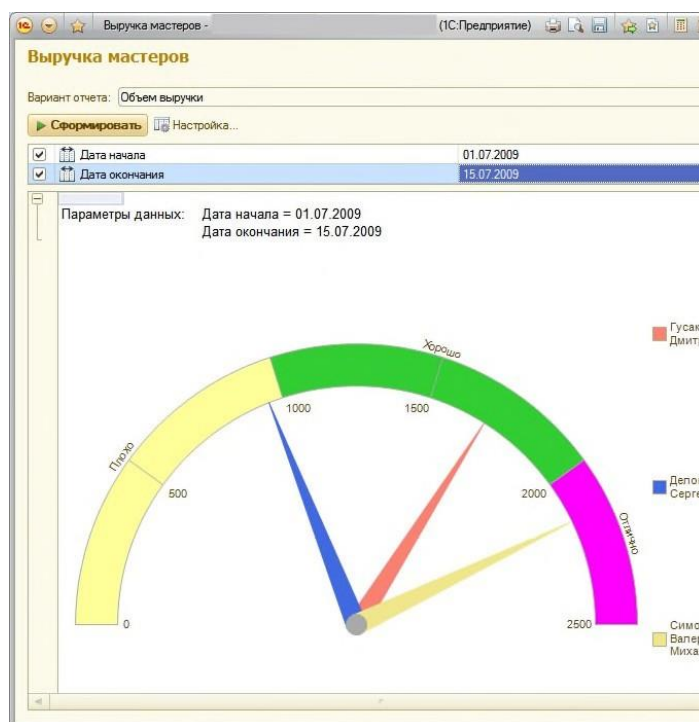


Рис. 15.20. Результат выполнения отчета

В результате мы видим те же данные, что и в основном варианте отчета, представленные в виде измерительной диаграммы. На диаграмме хорошо видна доля каждого мастера в общем объеме выручки. Обратите внимание, что при наведении курсора на сектор диаграммы появляется подсказка.

Если же понадобится посмотреть данные о работе какого-либо мастера с разбивкой по дням и клиентам, достаточно выбрать Основной вариант отчета и пересформировать отчет.

Таким образом, на примере отчета Выручка мастеров мы показали создание и использование различных вариантов отчета в целях наилучшего представления информации о работе мастеров.

Контрольные вопросы

- Что такое параметры виртуальной таблицы.
- Что такое левое соединение.
- Как использовать конструктор запроса.
- Как выбрать данные в некотором периоде для отчета.
- Как упорядочить данные в отчете.
-

Лабораторная работа № 16 **Отчеты. Получение актуальных сведений из периодического регистра сведений**

Язык запросов

Алгоритм, по которому данные будут выбраны из исходных таблиц запроса, описывается на специальном языке - языке запросов.

Текст запроса может состоять из нескольких частей:

- описание запроса,
- объединение запросов,
- упорядочивание результатов,
- автоупорядочивание,
- описание итогов.

Обязательной частью запроса является параметр «описание запроса». Описание запроса определяет источники данных, поля выборки, группировки и т. д. Объединение запросов определяет, как будут объединены результаты выполнения нескольких запросов. Упорядочивание результатов определяет условия упорядочивания строк результата запроса. Автоупорядочивание позволяет включить режим автоматического упорядочивания строк результата запроса. Описание итогов определяет, какие итоги необходимо рассчитывать в запросе и каким образом группировать результат. Следует заметить, что в случае, когда язык запросов используется для описания источников данных в системе компоновки данных, секция описания итогов языка запросов не используется. Это связано с тем, что система компоновки данных самостоятельно рассчитывает итоги на основании тех настроек, которые сделаны разработчиком или пользователем.

Отбор

- **В режиме Конфигуратор**

Создадим настройку отбора в отчете. Для этого в нижней части окна настроек перейдем на закладку Отбор.

Слева мы видим список доступных полей отчета.

Раскроем поле Услуга и двойным щелчком мыши на поле Родитель перенесем его в список условий отбора в правой части окна.

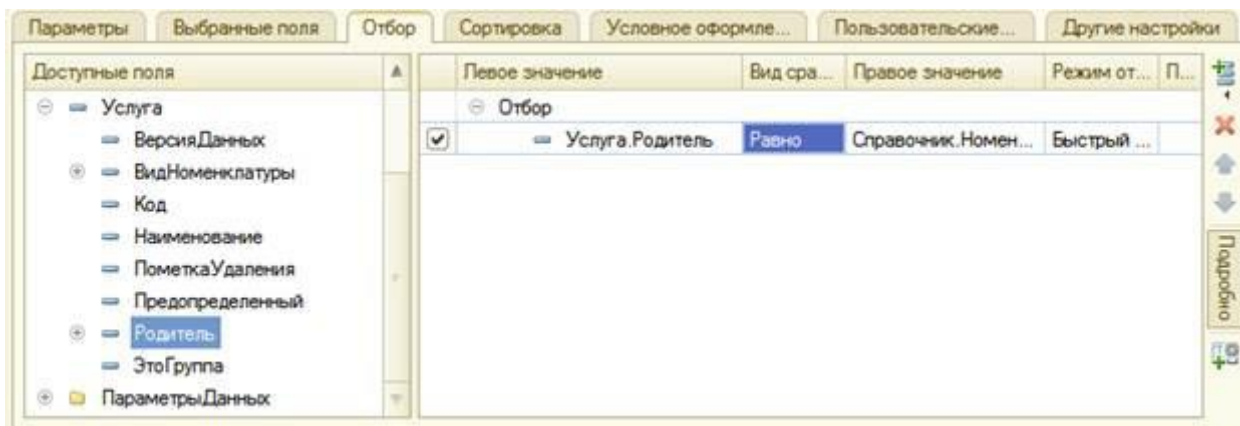


Рис. 13.14. Настройка отбора

Тем самым была создана возможность отбора по группам услуг, которые пользователь может задать в режиме 1С:Предприятие.

- **В режиме 1С:Предприятие**

Откроем отчет в режиме 1С: Предприятие и нажмем кнопку Настройка.

В окне пользовательских настроек отчета появились настройки Отбор и Условное оформление, которые мы только что отметили.

Настройку Непопулярная услуга мы заранее создали в конфигураторе. Добавив настройку условного оформления, мы предоставили пользователю возможность создавать любое количество собственных условий для условного оформления.

Пользователь сможет, при наличии определенной квалификации, задавать многие настройки по своему желанию.

Если же такого желания или соответствующих знаний у него нет, лучше задавать эти настройки жестко, а пользователю останется только включать или выключать их использование. Да собственно часто достаточно только отчетного периода или еще какой-то жизненно важной настройки, и такие настройки, конечно, нужно размещать непосредственно в отчетной форме. Получение актуальных значений из периодического регистра сведений.

Изучим возможность получения последних значений из периодического регистра сведений и с возможность вывода иерархических справочников.

- **В режиме Конфигуратор**

Добавим новый объект конфигурации Отчет.

Назовем его ПереченьУслуг и запустим конструктор схемы компоновки данных. Добавим новый Набор данных - запрос и вызовем конструктор запроса.

Запрос для набора данных

В качестве источника данных для запроса выберем объектную таблицу справочника Номенклатура и виртуальную таблицу регистра сведений ЦеныСрезПоследних.

Для того чтобы исключить неоднозначность имен в запросе, переименуем таблицу Номенклатура в СпрНоменклатура. Для этого выделим ее в списке Таблицы, вызовем ее контекстное меню и выберем пункт Переименовать таблицу.

Параметры виртуальной таблицы

Вызовем диалог ввода параметров виртуальной таблицы ЦеныСрезПоследних и укажем, что период будет передан в параметре ДатаОтчета. Для этого выделим эту таблицу в списке Таблицы и нажмем кнопку Параметры виртуальной таблицы.

Затем выберем из таблиц следующие поля:

- СпрНоменклатура.Родитель,
- СпрНоменклатура.Ссылка,
- ЦеныСрезПоследних.Цена.

Левое соединение таблиц

- На закладке *Связи*: в поле Условие связи, что значение измерения Номенклатура регистра сведений должно быть равно ссылке на элемент справочника Номенклатура. А также снимим флажок Все у таблицы регистра и установим его у таблицы справочника, тем самым установив вид связи как левое соединение для таблицы справочника:

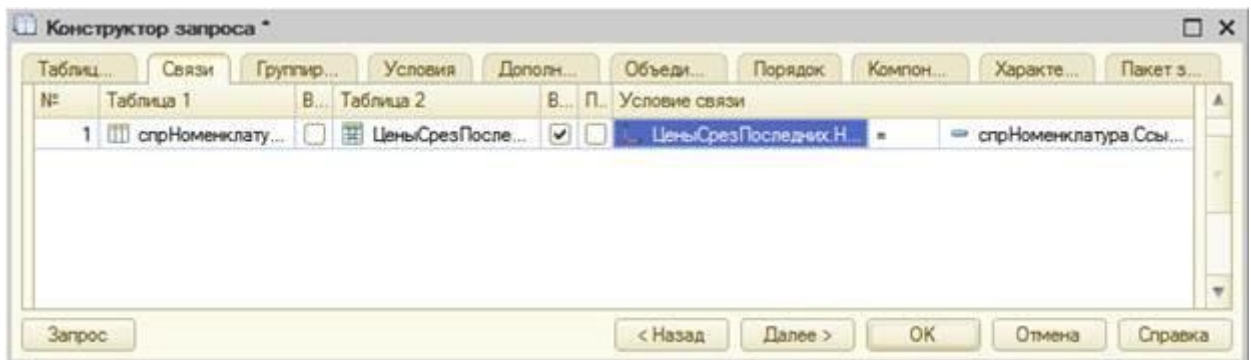


Рис. 13.15. Связь таблиц в запросе

- На закладке *Условия* зададим условие выбора элементов справочника Номенклатура - выбираемые элементы должны соответствовать виду номенклатуры, переданному в параметре запроса Вид Номенклатуры:

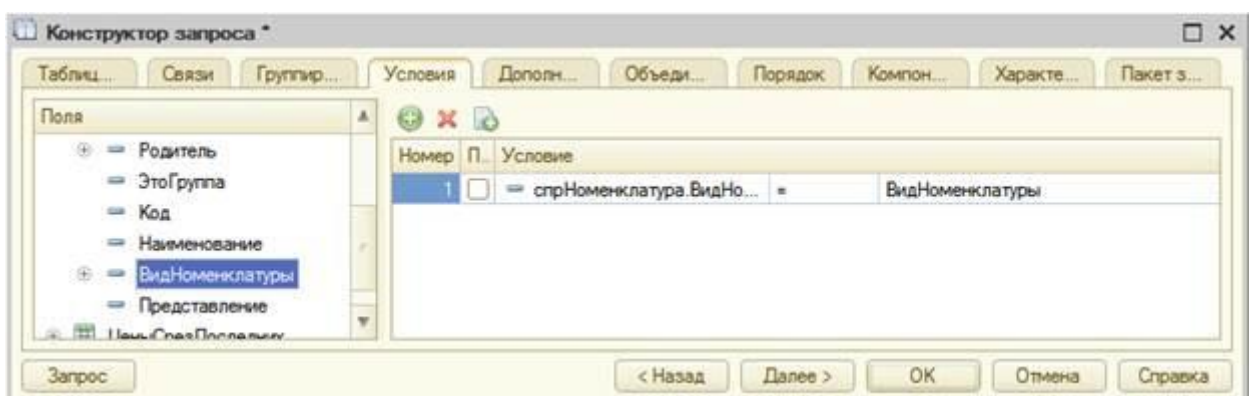


Рис. 13.16. Условия выбора элементов

- На закладке *Объединения/Псевдонимы*: указать псевдоним поля Родитель = ГруппаУслуг, а поля Ссылка = Услуга.

- Нажмем **OK**–

После этого, необходимо отредактировать схему компоновки данных, для этого на закладке **Ресурсы**, нажмем на кнопку **добавить** и выберем ресурс - **Цена**

- На закладке *Параметры* зададим значение параметра ВидНоменклатуры - Перечисление.ВидыНоменклатуры.Услуга.

Кроме этого, снимем ограничение доступности для параметра ДатаОтчета. В поле Тип этого параметра зададим состав даты - Дата. Для параметра Период, наоборот, установим ограничение доступности:

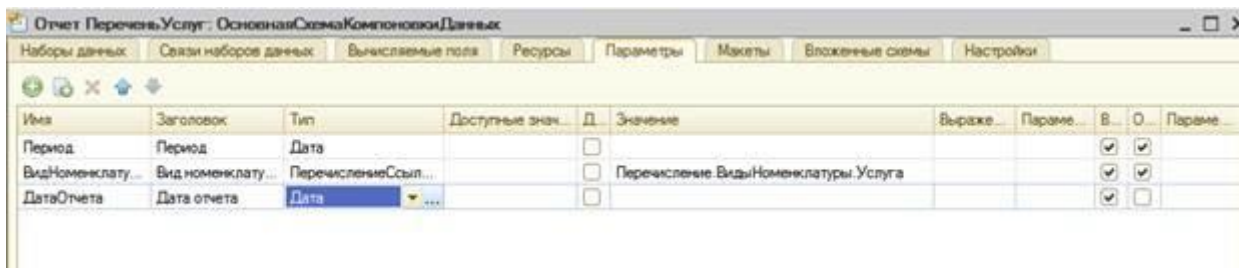


Рис. 13.17. Параметры схемы компоновки

Настройки

- Перейдем на закладку *Настройки*: создадим группировку по полю ГруппаУслуг, указав тип группировки Иерархия.

Существуют следующие типы иерархии для группировок отчета:

- Без иерархии - в группировке выводятся только неиерархические записи.
- Иерархия - в группировке выводятся как неиерархические, так и иерархические записи.
- Только иерархия - в группировке выводятся только иерархические (родительские) записи.

Внутри этой группировки создадим еще одну, без указания группового поля. На подзакладке *Выбранные поля*: укажем поля для вывода Услуга и Цена:

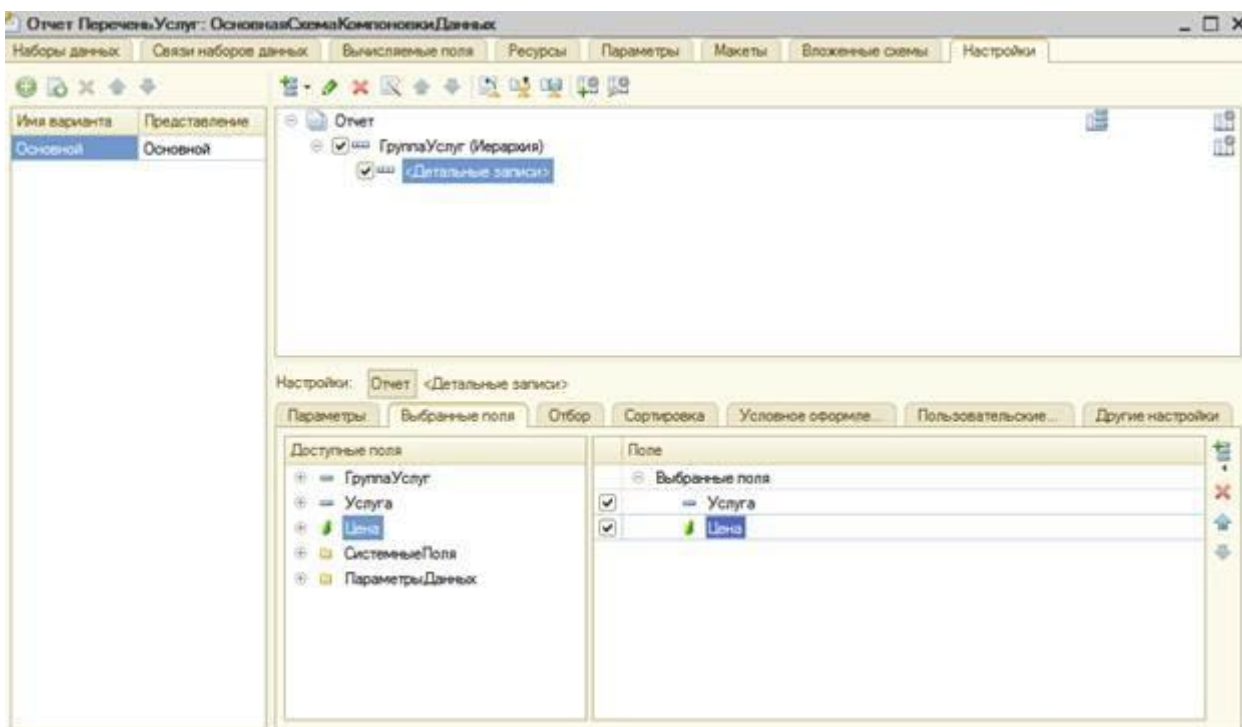


Рис. 13.18. Структура и поля отчета

- На подзакладке *Другие настройки* осуществим следующие действия:

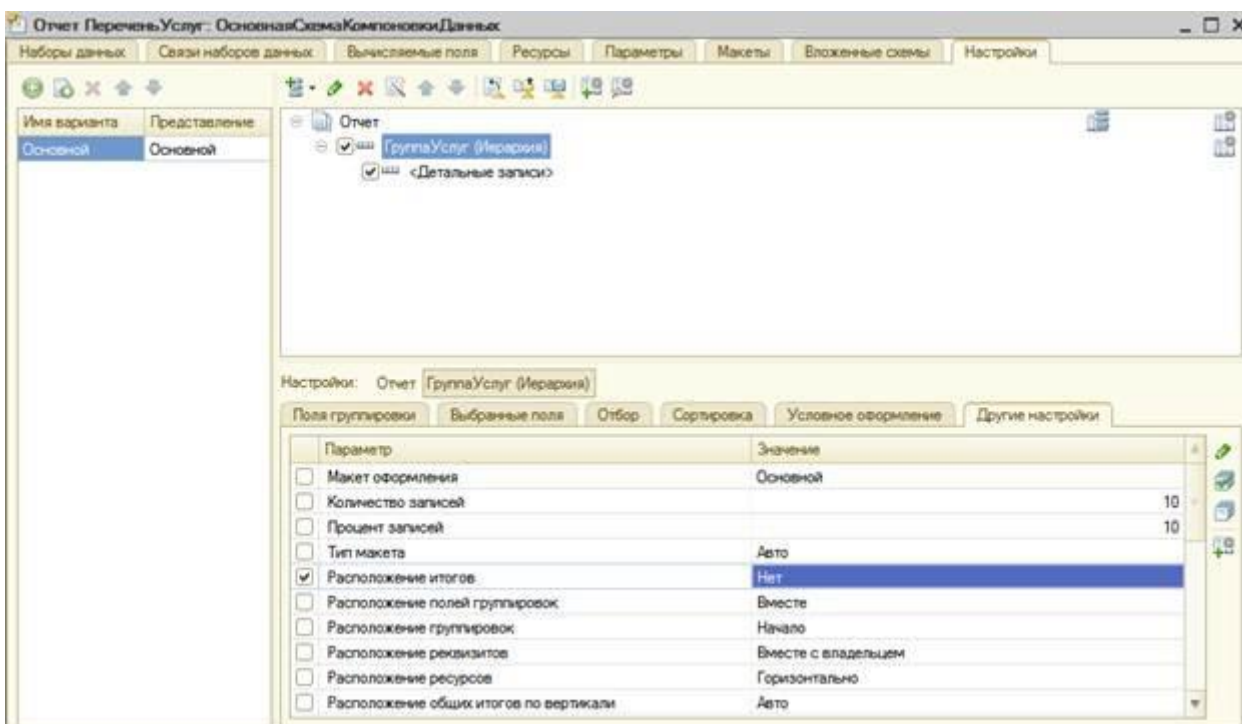


Рис. 13.19. Настройки вывода общих итогов для группировки "Группа Услуг"

- затем:

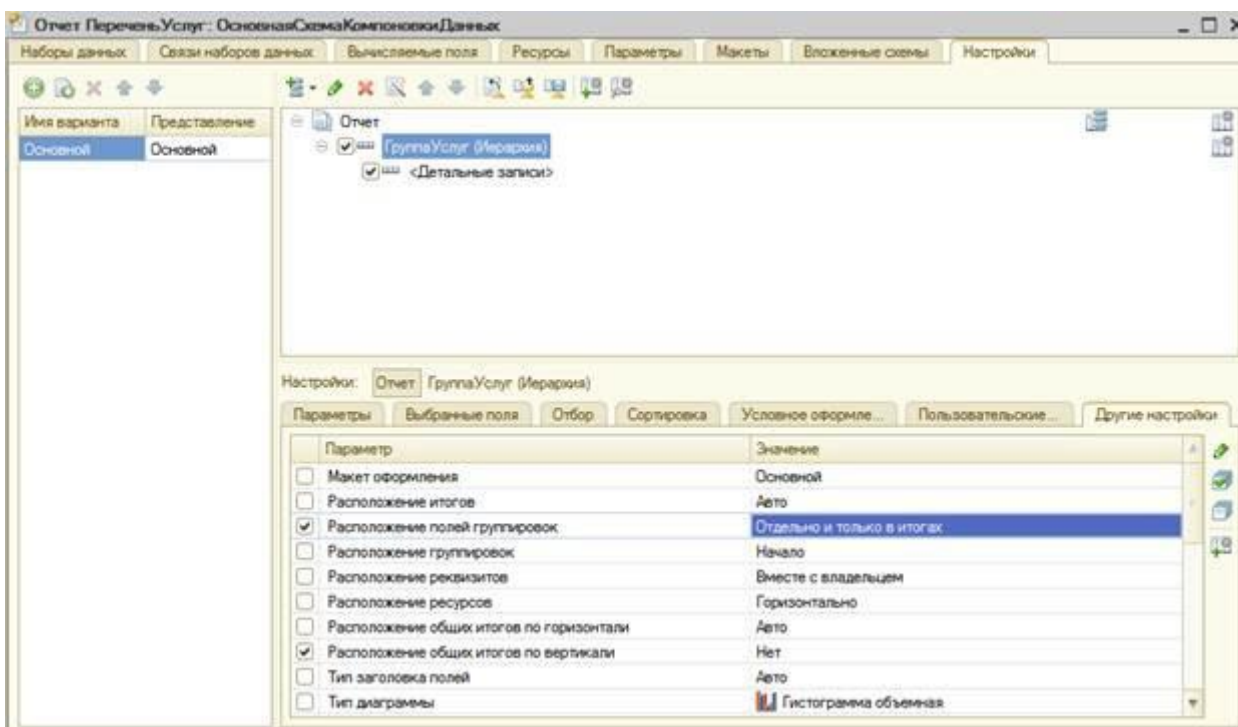


Рис. 13.20. Настройк и вывода итогов для глобального отчета

В заключение включим параметр Дата отчета в состав пользовательских настроек и установим для него Режим редактирования -Быстрый доступ.

Закроем конструктор схемы компоновки данных и в окне редактирования объекта ПереченьУслуг перейдем на закладку Подсистемы.

Отметим в списке подсистем конфигурации подсистемы Оказание услуг и Бухгалтерия.

- **В режиме 1С: Предприятие**

Запустим 1С:Предприятие в режиме отладки и прежде всего откроем периодический регистр Цены. *После чего протестируем отчет.*

На примере этого отчета было изучено, как система компоновки данных получает последние значения из периодического регистра сведений и как выводятся группировки по иерархии справочника.

Контрольные вопросы

- Как использовать в отчете данные нескольких таблиц.
- Как использовать группировки в структуре отчета.
- Как получить последние значения регистра сведений.
- Как вывести в отчет иерархические данные.
- Как управлять выводом итогов по группировкам и общим итогов.

Лабораторная работа № 17

Отчеты. Использование вычисляемого поля в отчете

Система компоновки данных

Система компоновки данных предназначена для создания произвольных отчетов в системе 1С: Предприятие и состоит из нескольких основных частей.

Исходные данные для компоновки отчета содержит в себе схема компоновки данных. Разработчик создает схему компоновки данных, в которой описывает текст запроса, наборы данных, связи между ними, доступные поля, параметры получения данных, и задает первоначальные настройки компоновки - структуру отчета, макет оформления данных и др.

Отчет системы компоновки имеет сложную иерархическую структуру и может состоять из различных элементов, таких как группировки, таблицы и диаграммы.

При этом пользователь может изменить существующую структуру отчета или вообще создать совершенно новую структуру отчета. Может настроить необходимый ему отбор, оформление элементов структуры отчета, получить расшифровку по каждому элементу и т. д.

Например, может быть задана такая структура отчета, состоящая из одной таблицы и одной диаграммы.

В этом случае сформированный отчет будет иметь следующий вид.

В представленном отчете таблица будет состоять из записей регистра накопления ПродажиОборота о клиентах и оказанных им услугах. Эти записи сгруппированы по мастерам, которые выполняли заказы. А в группировке будет выведен список услуг, оказанных данным мастером, и затраченных на это материалов.

Как мы уже говорили в начале раздела, система компоновки данных представляет собой совокупность нескольких объектов. При формировании и исполнении отчета происходит последовательная передача данных от одного объекта системы компоновки данных к другому, до получения конечного результата - документа, показанного пользователю.

Алгоритм взаимодействия этих объектов выглядит следующим образом:

Разработчик создает схему компоновки данных и настройки по умолчанию. В общем случае на основе одной схемы компоновки данных может быть создано большое количество различных отчетов. Настройки компоновки данных, создаваемые разработчиком или изменяемые пользователем, определяют, какой именно отчет будет получен в конкретном случае.

- На основе схемы компоновки и имеющихся настроек компоновщик макета создает макет. Это этап подготовки к исполнению отчета. Макет компоновки данных является уже готовым заданием для выполнения процессором компоновки. Он содержит необходимые запросы, макеты областей отчета и др.
- Процессор компоновки данных выбирает данные из информационной базы согласно макету компоновки, агрегирует и оформляет эти данные.

Результат компоновки обрабатывается процессором вывода, и в итоге пользователь получает результирующий табличный документ.

Использование вычисляемого поля в отчете

Следующий отчет - Рейтинг клиентов - будет показывать в графическом виде, каков доход

от оказания услуг каждому из клиентов за все время работы предприятия.

- **В режиме Конфигуратор**

Добавим новый объект конфигурации Отчет. Назовем его РейтингКлиентов и запустим конструктор схемы компоновки данных.

Создадим новый Набор данных - запрос и вызовем конструктор запроса.

Запрос для набора данных

В качестве источника данных для запроса выберем виртуальную таблицу регистра накопления Продажи.Обороты.

Затем выберем из нее следующие поля:

- ПродажиОбороты. Клиент,
- ПродажиОбороты.ВыручкаОборот,
- ПродажиОбороты.СтоимостьОборот.

- На закладке *Объединения/Псевдонимы* укажем, что поле ВыручкаОборот будет иметь псевдоним Выручка, а поле СтоимостьОборот -Стоимость.

- *Нажмем ОК*–

Перейдем к редактированию схемы компоновки данных.

Вычисляемые поля

На этом этапе мы столкнулись с необходимостью отразить в отчете поле, которого нет в наборе данных. Раньше мы использовали в отчете те поля, которые описывались в наборе данных. Теперь, чтобы отобразить доход от оказания услуг в разрезе клиентов, нам необходимо дополнительное поле, рассчитанное как разница между выручкой и стоимостью оказания услуг.

Для этого в системе компоновки данных есть возможность определения вычисляемого поля.

Вычисляемые поля: дополнительные поля схемы компоновки данных, значения которых будут вычисляться по некоторой формуле.

- На закладке *Вычисляемые поля* добавим вычисляемое поле, нажав кнопку *Добавить*. Дадим ему имя - Доход, в колонку *Выражение* введем следующий код:

Листинг 13.2. Выражение для расчета вычисляемого поля «Доход»

Выручка – Стоимость

Ресурсы

- На закладке *Ресурсы* нажатием кнопки » выберем все доступные ресурсы отчета:

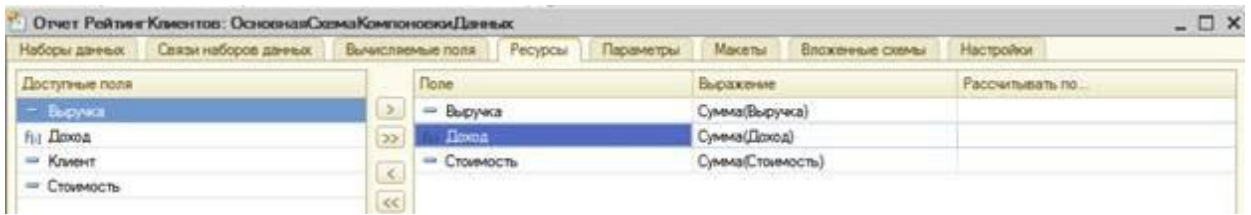


Рис. 13.21. Ресурсы схемы компоновки данных

Настройки

- На закладке *Настройки* добавим в структуру отчета диаграмму. Для этого нажмем кнопку *Добавить* в командной панели окна настроек и добавим диаграмму:

Затем выделим ветку *Точки* и добавим в нее группировку по полю *Клиент*. Серии диаграммы оставим без изменений.

Дело в том, что для демонстрации рейтинга клиентов хорошо подойдет круговая диаграмма, которую мы хотим показать. Для этого вида диаграммы достаточно задать только точки, поэтому серии мы не задаем.

- На закладке *Выбранные поля* выберем поле *Доход* для вывода в отчет.

Структура отчета должна принять следующий вид:

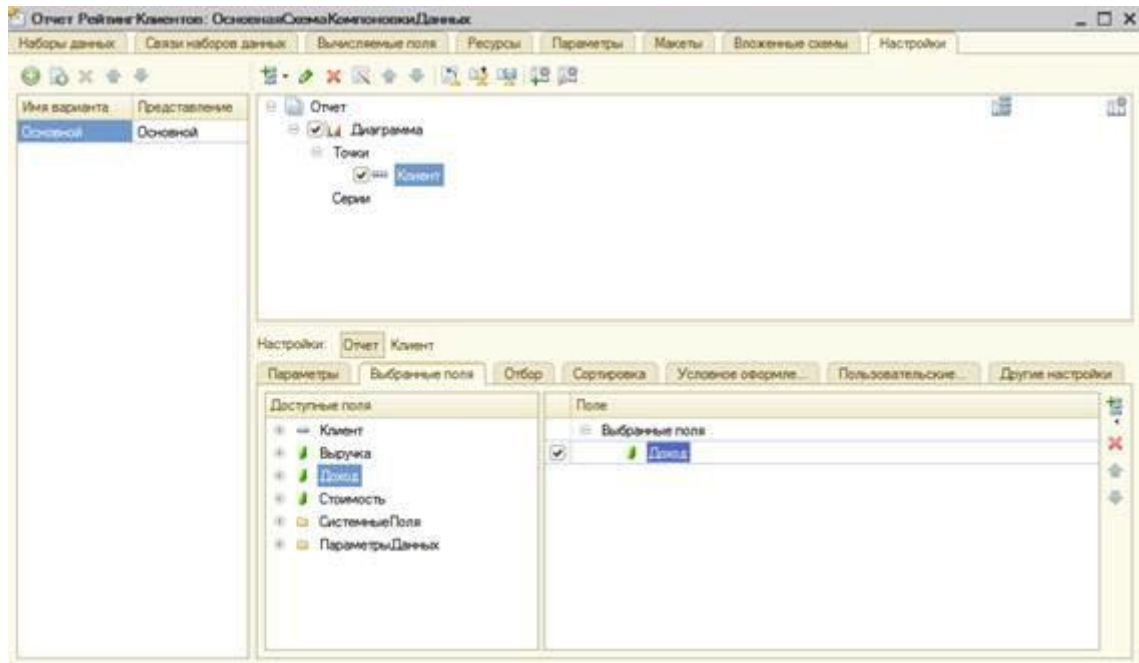


Рис. 13.22. Структура отчета и настройки диаграммы

- На закладке *Другие настройки* выберем тип диаграммы *Круговая объемная* и зададим заголовок отчета - *Рейтинг клиентов*.

После этого в окне редактирования объекта *РейтингКлиентов* перейдем на закладку *Подсистемы*. Отметим в списке подсистем конфигурации подсистемы *Оказание услуг* и *Бухгалтерия*.

- **В режиме 1С: Предприятие**

Запустим 1С:Предприятие в режиме отладки и выполним команду Рейтинг клиентов в панели действий раздела Бухгалтерия. Нажмем Сформировать.

Мы видим данные о доходе от оказания услуг по каждому из клиентов, представленные в виде круговой диаграммы. Если вернуться в конфигуратор и изменить тип диаграммы на Гистограмма объемная, то наша диаграмма поменяет вид.

Контрольные вопросы

- Как создать отчет, содержащий диаграмму.
- Как использовать параметры в системе компоновки данных.
- Что такое ресурсы в системе компоновки данных.
- Что такое вычисляемые поля в системе компоновки данных.
- Как дополнить данные отчета всеми датами в группировке по периоду.

Лабораторная работа № 18 Отчеты. Вывод данных в таблицу

Виртуальные таблицы запросов

При создании запроса платформа предоставляет нам в качестве источников данных некоторое количество виртуальных таблиц. Название «виртуальные» полностью соответствует их сути, поскольку эти таблицы, в свою очередь, также являются результатом запроса, который система формирует в момент выполнения соответствующего участка кода.

По большому счету разработчик может самостоятельно получить те же самые данные, которые система предоставляет ему в качестве виртуальных таблиц, однако алгоритм получения этих данных не будет оптимизирован в силу следующих двух причин. Во-первых, все виртуальные таблицы параметризованы, то есть разработчику предоставляется возможность задать некоторые параметры, которые система будет использовать при формировании запроса создания виртуальной таблицы.

Примечательным здесь является то, что задание параметров виртуальной таблицы далеко не всегда приводит к простой подстановке указанных разработчиком значений в текст запроса. В зависимости от того, какие параметры виртуальной таблицы указаны разработчиком, система может формировать РАЗЛИЧНЫЕ запросы для получения одной и той же виртуальной таблицы, причем они будут оптимизированы с точки зрения переданных параметров. Во-вторых, не всегда разработчик имеет возможность получить доступ к тем данным, к которым имеет доступ система. Например, при использовании виртуальных таблиц регистров сведений разработчику доступна по большому счету вся та же информация о данных регистров, которую использует система при формировании запроса виртуальной таблицы.

Совсем иная картина с виртуальными таблицами регистров накопления. Здесь система динамически формирует запрос в зависимости не только от переданных параметров, но и от периода рассчитанных итогов регистра, причем в запросе она использует

данные рассчитанных итогов, которые просто недоступны для разработчика при создании запроса.

Конечно, разработчик может самостоятельно перебрать все записи регистра накопления и в итоге получить те же самые данные, которые система предоставляет в виде виртуальной таблицы. Однако очевидно, что такой запрос будет менее эффективным и потребует от разработчика гораздо больше трудозатрат.

Вывод данных в таблицу

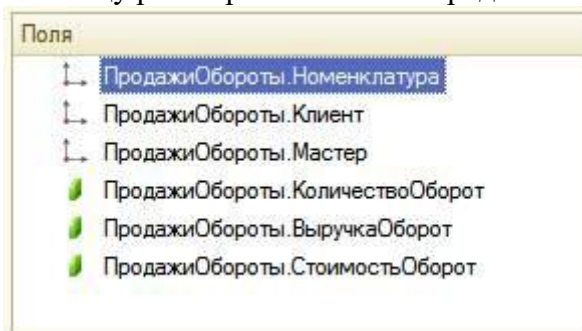
Это наиболее гибкий и универсальный отчет, который позволяет пользователю изменять его структуру и внешний вид самостоятельно.

- **В режиме Конфигуратор**

Добавим новый объект конфигурации *Отчет*. Назовем его *Универсальный* и запустим конструктор схемы компоновки данных. Создадим новый Набор данных - запрос и вызовем конструктор запроса.

Запрос для набора данных

В качестве источника данных для запроса выберем виртуальную таблицу регистра накопления *Продажи.Обороты*.



Затем выберем из нее все поля:

Рис. 13.23. Выбранные поля

- Нажмем *OK*–
 - На закладке *Ресурсы* нажатием кнопки » выберем все доступные ресурсы отчета.
 - На закладке *Настройки*: нажмем кнопку *Добавить* в командной панели окна настроек и добавим таблицу.
- Мы не будем здесь задавать строки и колонки этой таблицы, а также список выбранных полей, так как хотим предоставить полную свободу пользователю в этих действиях.

Для этого выделим в структуре элементов отчета элемент *Таблица* и нажмем кнопку *Свойства* элемента пользовательских настроек, расположенную вверху в командной панели окна настроек.

В появившемся окне сделаем следующие изменения:

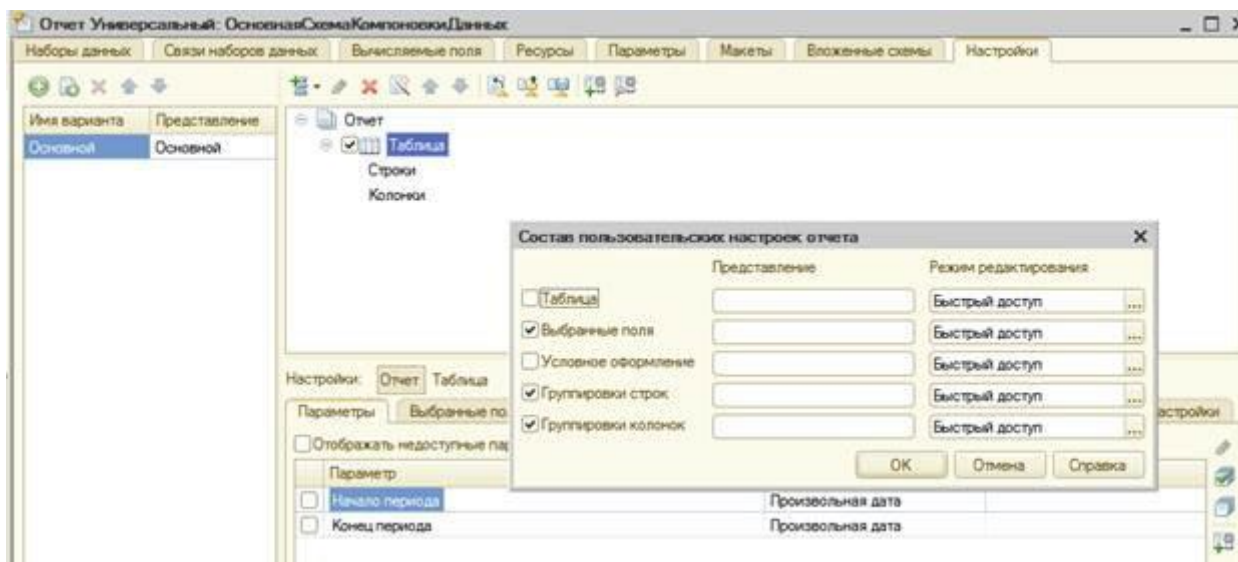


Рис. 13.24. Состав пользовательских настроек

В заключение определим, в каких подсистемах будет отображаться наш отчет. Закроем конструктор схемы компоновки данных и в окне редактирования объекта Универсальный перейдем на закладку Подсистемы. Отметим в списке подсистем конфигурации подсистему Оказание услуг.

- **В режиме 1С: Предприятие**

Запустим 1С: Предприятие в режиме отладки и выполним команду Универсальный в панели действий раздела ОказаниеУслуг.

Если мы сейчас нажмем Сформировать, то ничего не увидим в результате, так список выбранных полей, группировок строк и колонок таблицы пуст. Пользователь имеет возможность самостоятельно формировать требуемый отчет по регистру Продажи.

Контрольные вопросы

- Как создать пользовательские настройки отчета.
- В чем отличие «быстрых» настроек от остальных пользовательских настроек.
- Как определить состав пользовательских настроек отчета.
- Как вывести данные в виде таблицы.
- Как сделать отчет универсальным.

Контрольно-измерительный материал

1. С помощью чего осуществляется разработка бизнес-приложений в системе 1С:Предприятие 8?

- Технологическая платформа***
- Конфигурация
- Информационная база
- СУБД

2. Где определяется структура создаваемого бизнес-приложения в системе 1С:Предприятия 8?
- Технологическая платформа
 - Конфигурация***
 - Информационная база
 - СУБД
3. Где хранятся учетные данные бизнес-приложения в системе 1С:Предприятие 8?
- Технологическая платформа
 - Конфигурация
 - Информационная база***
4. С помощью чего система 1С:Предприятие 8 работает с данными?
- Технологическая платформа
 - Конфигурация
 - Информационная база
 - СУБД***
5. С какими СУБД работает система 1С:Предприятие 8?
- Microsoft SQL Server
 - Microsoft SQL Server, PostgreSQL
 - Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database, FileDBMS***
 - Microsoft SQL Server, PostgreSQL, IBM DB2
6. Какого вида клиентского приложения не существует в системе 1С:Предприятие 8?
- Отладочный клиент***
 - Толстый клиент
 - Тонкий клиент
 - Веб - клиент
7. Для каких целей может использоваться "Толстый клиент"?
- Использование прикладного решения
 - Отладка прикладного решения
 - Разработка прикладного решения
 - Верны все варианты***
8. Для каких целей может использоваться "Тонкий клиент"?
- Использование прикладного решения***
 - Отладка прикладного решения
 - Разработка прикладного решения
 - Верны все варианты
9. На компьютерах с какой операционной системой могут работать пользователи системы 1С:Предприятие 8.2?
- На всех версиях Windows
 - На всех версиях Windows и Linux
 - На Windows - с использованием толстого клиента и тонкого, на Linux - через интернет браузер - Веб-клиент***
 - На Windows - Толстый клиент, на Linux - Тонкий и Веб-клиент
10. Если рабочие серверы кластера 1С:Предприятие работают с использованием разных операционных систем то:
- никаких ограничений на используемые СУБД нет
 - с MS SQL сервер можно работать с рабочего сервера под управлением операционных систем Windows, с другими СУБД только из под Linux

- С. с MS SQL сервер можно работать с рабочего сервера под управлением операционных систем Windows, с другими СУБД как из под Windows, так и из под Linux***
- 8? 11. Что разрешено разработчикам прикладных решений в системе 1С:Предприятие 8?
- А. Изменять функциональность типовых тиражных решений
 - В. Создавать собственные прикладные решения
 - С. Изменять функциональность технологической платформы
 - Д. Правильны ответы 1 и 2***
12. Какие компоненты типовой поставки в системе 1С:Предприятие 8 защищены аппаратным ключом?
- А. Клиентское приложение
 - В. Типовое тиражное решение
 - С. Сервер 1С:Предприятия
 - Д. Правильны ответы 1 и 3***
13. Для каких компонент существуют сетевые ключи защиты?
- А. Клиентское приложение***
 - В. Типовое тиражное решение
 - С. Сервер 1С:Предприятия
 - Д. Правильны ответы 1 и 3
14. Сколько конфигураций обязательно должно существовать в прикладном решении?
- А. Одна
 - В. Две***
 - С. Три
 - Д. Сколько угодно
15. Сколько информационных баз может быть с одной и той же конфигурацией?
- А. Только одна
 - В. Только две (рабочая и демонстрационная)
 - С. Неограниченно***
 - Д. Определяется комплектом поставки прикладного решения
16. Какую конфигурацию можно изменять интерактивно в конфигураторе?
- А. Основная конфигурация***
 - В. Конфигурация базы данных
 - С. Конфигурация поставщика
 - Д. Внешние конфигурации
17. Из чего состоит конфигурация?
- А. Объекты конфигурации***
 - В. Объекты встроенного языка
 - С. Объекты информационной базы
 - Д. Верны ответы 1 и 3
18. Какие объекты используются в алгоритмах?
- А. Объекты конфигурации
 - В. Объекты встроенного языка
 - С. Объекты информационной базы
 - Д. Верны все варианты***
19. В каком объекте содержится редактируемая пользователем информация?
- А. Объект конфигурации
 - В. Объект встроенного языка***
 - С. Объект информационной базы
 - Д. Верны все варианты

20. Что содержится в объекте информационной базы при редактировании нового элемента справочника с табличными частями?
- A. Данные реквизитов
 - B. Ссылка на элемент справочника
 - C. Строки табличных частей
 - D. Верны ответы 1 и 3
21. Что содержится в объекте информационной базы при редактировании существующего в базе элемента справочника с табличными частями?
- A. Данные реквизитов
 - B. Ссылка на элемент справочника
 - C. Строки табличных частей
 - D. Верны все варианты***
22. Для включения отборов в списках по содержимому табличных частей объектов и их свойств, неотображаемых в форме:
- A. ничего дополнительного настраивать не надо, все реализовано на уровне платформы
 - B. реализуется только запросами
 - C. осуществляется только с помощью дополнительной настройки форм
 - D. необходимо создать объект критерии отбора***
23. Параметры сеанса иницируются:
- A. при запуске системы
 - B. при программном обращении
 - C. во время работы интерактивно
 - D. Верны варианты 1 и 2
24. Роль, как объект конфигурации, отражает:
- A. статус и сферу ответственности ее носителя как сотрудника компании
 - B. совокупность доступных определенному пользователю объектов в среде "1С:Предприятие"
 - C. совокупность прав действий в среде "1С:Предприятие" с возможностью присвоения конкретному пользователю***
 - D. набор параметров пользователя
25. Доступ к данным может задаваться...
- A. для видов объектов информационной базы
 - B. на объекты информационной базы
 - C. на свойства объектов информационной базы
 - D. на стандартные реквизиты объектов информационной базы
 - E. Верны все варианты
26. При настройке ограничения прав доступа на уровне записей и полей базы данных значения параметров для условий определяются...
- A. как переменные, определенные в модуле приложения
 - B. как переменные, определенные в модуле приложения с ключевым словом Экспорт
 - C. как параметры сеанса
 - D. как константы
27. Ограничения, накладываемые на работу с объектами (полученные при настройке ролей) действуют...
- A. только на интерактивные действия пользователя
 - B. только на программные действия, выполняемые в контексте пользователя с данной ролью
 - C. одновременно на интерактивные действия и программные механизмы

- D. в зависимости от выбранного права, либо на интерактивные действия, либо на программные механизмы, выполняемые в контексте пользователя с данной ролью***
28. Для запрета режима непосредственного удаления объектов пользователем необходимо...
- A. в свойствах конфигурации снять флаг "Разрешить непосредственное удаление объектов"
 - B. при определении роли для соответствующих объектов снять отметку у флага "Удаление"
 - C. при определении роли для соответствующих объектов снять отметку у флага "Интерактивное удаление"***
 - D. в свойствах роли снять отметку у флага "Разрешить удаление объектов"
29. При настройке ограничений прав доступа (в ролях), при определении шаблонов:
- A. в них можно определить один параметр, значение которого будет передаваться при указании шаблона в конкретном ограничении доступа к данным
 - B. в них можно определить любое количество параметров, значение которых будет передаваться при указании шаблона в конкретном ограничении доступа к данным***
 - C. параметры в шаблонах использовать нельзя
 - D. при описании ограничений прав доступа нет возможности использовать шаблоны
30. Картинки для прикладного решения могут браться из ...
- A. библиотеки картинок технологической платформы
 - B. библиотеки картинок конфигурации
 - C. из файловой системы
 - D. верны все варианты***
31. Для возможности переключения пользователем интерфейсного языка прикладного решения необходимо:
- A. в настройках пользователя выбрать возможный язык, из списка дополнительных интерфейсов
 - B. в настройках пользователя выбрать возможный язык, из списка языков интерфейсов, заданных в ветви метаданных языки***
 - C. в настройках пользователя выбрать возможный язык, из списка predeterminedных в платформе языков
 - D. верны все варианты
32. Данные какого-либо созданного объекта (элемента справочника для которого включен механизм полнотекстового поиска) попадают в полнотекстовый индекс
- A. при записи объекта
 - B. при выполнении специального метода встроенного языка системы
 - C. при выполнении команды из пункта меню "Операции/..."
 - D. верны варианты 2,3***
33. При необходимости внести в механизм полнотекстового поиска свой дополнительный словарь (синонимы, специфичные для какого-либо рода деятельности) его можно включить в конфигурацию:
- A. как константу
 - B. как общий макет
 - C. как параметр сеанса
 - D. верны ответы 1,2***

34. В механизме полнотекстового поиска индексированию не подлежат реквизиты следующих типов:
- Дата
 - Хранилище значений
 - Булево***
 - Все ответы верны
35. Произвольная классификация объектов конфигурации осуществляется с помощью...
- свойства Комментарий у объекта конфигурации
 - создания реквизита "Классификация"
 - с помощью объекта конфигурации Подсистемы***
36. Какой из вариантов лучше всего предложить пользователю, чтобы привести учет, контролируемый последовательностью, в актуальное состояние?
- Воспользоваться режимом восстановления последовательности из диалога "Операции" / "Проведение документов..." / "Восстановление последовательностей"***
 - Перепровести в хронологической последовательности все документы
 - Перепровести в хронологической последовательности все документы, принадлежащие последовательности
 - Перепровести в хронологической последовательности все документы, принадлежащие последовательности, начиная с границы последовательности. Границу последовательности определить через "Табло" по формуле:
Последовательности.ИмяПоследовательности.ПолучитьГраницу().Ссылка
37. Объект конфигурации Константа...
- предназначен для хранения предположительно не изменяющийся информации
 - для хранения значений во времени необходимо установить признак периодичности
 - в платформе 8 не поддерживает периодичности, необходимо использовать периодический регистр сведений
 - верны ответы 1 и 3***
38. Какие виды иерархии существуют в системе 1С:Предприятие 8?
- Иерархия групп
 - Иерархия элементов
 - Иерархия групп и элементов
 - Верны ответы 2 и 3***
39. Какие варианты подчинения существуют в системе 1С:Предприятие?
- группам
 - элементам
 - группам и элементам
 - Верны все указанные ответы***
40. В какое значение можно установить свойство "Серии кодов" объекта конфигурации "Справочник"?
- Во всем справочнике
 - В пределах подчинения
 - В пределах подчинения владельцу
 - Верны все указанные ответы***
41. Объект конфигурации Перечисление...
- прикладной объект, значения которого задаются в Конфигураторе
 - может хранить различные типы значений

- С. для него может быть заданы форма списка и выбора
 D. верны ответы 1 и 3***
42. Назовите основное назначение объектов типа "Документ"?
- A. Предназначены для хронологического отражения в системе событий предметной области, например, хозяйственных операций предприятия, контактов с покупателями***
- B. Предназначены для отражения в системе условно-постоянной информации, например, карточек контрагентов
- C. Предназначены только для отражения хозяйственных операций в регистрах учета, например, в регистрах бухгалтерии
43. Какое максимальное количество видов документов позволяет создать система 1С:Предприятие 8 в процессе конфигурирования?
- A. Неограниченное***
- B. Ограничивается свойствами конфигурации "Количество видов документов"
- C. 256, т.к. система не позволяет обрабатывать более 256 таблиц объектов одного типа
- D. 50, т.к. максимальная длина номера документа равна 50 символов
44. Назовите основное назначение объектов типа "ЖурналДокументов"
- A. Регистрация в единой таблице документов различных видов, собранных по принципу общности отражаемых событий предметной области и предоставляемых пользователю в едином, упорядоченном по хронологии списке***
- B. Журнал представляет собой список документов определенного вида, упорядоченный по хронологии. Каждый вид документа должен быть зарегистрирован в одном из журналов документов, иначе пользователь не сможет работать со списком документов этого вида
- C. Журнал документов представляет собой таблицу для хранения всех документов в едином списке, упорядоченном по хронологии. Отличаются же журналы друг от друга только составом дополнительных граф
- D. В системе уже есть предопределенный объект типа "ЖурналДокументов", называемый "Общий журнал документов", назначение которого - регистрация в единой таблице документов всех видов в упорядоченном по хронологии виде. Разработчик не может создавать свои журналы документов
45. Объект конфигурации Отчет...
- A. может быть сохранен в составе конфигурации и как внешняя обработка
- B. может сравниваться с внешней обработкой, формируя отчет в печатной форме
- C. может быть добавлен в конфигурацию из внешней обработки
- D. верны все указанные ответы***
46. Внешние обработки используются:
- A. На стадии отладки, для более быстрого внесения изменений в обработку***
- B. Для разграничения прав доступа к обработке
- C. Для возможности редактирования обработки пользователем в режиме 1С:Предприятие
47. Назовите основную функцию нумератора документов?
- A. Сквозная нумерация документов разного вида***
- B. Чтобы при конфигурировании легче было определять свойства нумерации документов
- C. Только для обеспечения уникальности номеров документов в пределах года
- D. Он используется только для нумерации тех документов, у которых снят флажок свойства "Автономмерация"
48. При создании нового документа и отказа от его записи номер документа:
- A. освобождается (он может автоматически назначаться другому экземпляру документа в этом периоде)

- В. не освобождается (он не будет автоматически назначаться другому экземпляру документа в этом периоде)
 - С. освобождается или не освобождается зависит от значения соответствующего свойства конфигурации***
49. С помощью какого свойства можно определить, какому документу принадлежит запись регистра?
- А. Регистратор***
 - В. Документ
 - С. Текущий Документ
 - Д. Специального свойства у записи регистра нет. Для этих целей можно определить у регистратора реквизит типа "ДокументСсылка"
50. Для каких объектов конфигурации возможен ввод predetermined значений?
- А. Справочники
 - В. Документы
 - С. Планы видов характеристик, планы счетов, планы видов расчета
 - Д. Справочники, планы видов характеристик, планы счетов, планы видов расчета***

Литература

1. Гартвич А. Бухгалтерский учет в 1С:Бухгалтерии 8.3. Самоучитель. – СПб: БХВ-Петербург, 2015
2. Кашаев С. 1С:Предприятие 8.3. Программирование и визуальная разработка в примерах.– СПб: БХВ-Петербург, 2015
3. Гартвич А. 1С:Бухгалтерии 8.3. с нуля. 101 урок для начинающих – 2-е изд., перераб. И доп. – СПб: БХВ-Петербург, 2016
4. Ощенко И. Азбука программирования в 1С:Предприятие 8.3. – СПб: БХВ-Петербург, 2015

Дополнительные источники

1. Официальный сайт фирмы «1С» <http://v8.3c.ru/>