«ҚОСТАНАЙ ЖОҒАРЫ ПОЛИТЕХНИКАЛЫҚ КОЛЛЕДЖІ» КМҚК
КГКП «КОСТАНАЙСКИЙ ПОЛИТЕХНИЧЕСКИЙ ВЫСШИЙ КОЛЛЕДЖ»

Учебно-методический комплекс

ПМ01 Проектирование программного обеспечения

Костанай, 2021 г

МАЗМҰНЫ

Раздел 1. Технология конструирования программного обеспечения

Тема 1.1 Базовые понятия технологии конструирования и жизненного цикла программного обеспечения (ПО)

Процесс современной разработки программного обеспечения ориентирован на жизненный цикл программного продукта. Все существующие в настоящее время технологии, методики и стандарты напрямую или косвенно касаются или регламентируют этапы жизненного цикла, как по функциональному наполнению, так и по содержанию. ... Технология разработки программного обеспечения – это система инженерных принципов для создания экономичного ПО с заданными характеристиками качества.

Технология конструирования программного обеспечения (ТКПО) – система инженерных принципов для создания экономичного ПО, которое надежно и эффективно работает в реальных компьютерах. Различают методы, средства и процедуры ТКПО. Методы обеспечивают решение следующих задач: - планирование и оценка проекта; - анализ системных и программных требований; - проектирование алгоритмов, структур данных и программных структур; - кодирование; - тестирование; - сопровождение. Средства (утилиты) ТКПО обеспечивают автоматическую поддержку методов.

Процесс современной разработки программного обеспечения ориентирован на жизненный цикл программного продукта. Все существующие в настоящее время технологии, методики и стандарты напрямую или косвенно касаются или регламентируют этапы жизненного цикла, как по функциональному наполнению, так и по содержанию. Процесс разработки программных систем тесно связан с областью управления проектами, потому что любой программный продукт является уникальным результатом. ... Существуют различные определения технологии разработки программного обеспечения. К наиболее распространенным относятся следующие.

Модели жизненного цикла и технологии Проектирования программног о обеспечения. Учебно-методическое пособие Рекомендовано методической комиссией ИИТММ для студентов ННГУ, обучающихся по направлению подготовки 09.04.03 «Прикладная информатика». Нижний Новгород 2016. ... В пособии расссмотрены как теоретические вопросы проектирования информационных систем:

Тема 1.2 Стадии и этапы разработки программ

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания. кадровый учёт программный visual. На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ: - разработка конфигурации; - разработка программной документации ... Назначение программы.

Создание программного обеспечения осуществляется последовательно в соответствии со следующими основными этапами:

постановка задачи (стадия «Техническое задание»)

анализ требований и разработка спецификаций (стадия «Эскизный проект»)

проектирование (стадия «Технический проект»)

реализация (стадия «Рабочий проект»)

Этапы разработки программного обеспечения. В зависимости от вида, масштабов и потребностей проекта определяется порядок разработки. Он будет несколько отличаться для разработки мобильных приложений, встроенного ПО, решений для автоматизации и БД, но общая последовательность действий для создания ПО универсальна: Подробно про первый и второй этапы (подготовительный и проектирование программного обеспечения) можно перечитать в прошлой статье. Перейдём к созиданию: Дизайн — вторая по важности составляющая продукта после технических характеристик, влияющая на эффективность и скорость взаимодействи ние и эксплуатация (стадия «Внедрение»)

Настоящий стандарт устанавливает стадии разработки программ и программной документации для вычислительных машин, комплексов и систем независимо от их назначения и области применения. 2. Стадии разработки, этапы и содержание работ должны соответствовать указанным в таблице. Стадии разработки. Этапы работ. Содержание работ. 1. Техническое задание. Обоснование необходимости разработки программы. Постановка задачи. Сбор исходных материалов. Выбор и обоснование критериев эффективности и качества разрабатываемой программы.

Определяются стадии, этапы и сроки разработки программы и документации. Анализ требований и определение спецификаций. Спецификации — это формализованное описание функций и ограничений создаваемого приложения. ... Реализация является последовательным процессом создания исходных кодов программы на выбранном языке

программирования (кодирование), тестирования и отладки программного обеспечения. Часто при создании программы подавляющая часть времени уходит не на ее разработку, а на отладку и тестирование. Поэтому программа должна быть наглядной, легко читаемой, сопровождаться комментариями.

Тема 1.3 Структурный подход к проектированию ПО

Структурный (функциональный) подход заключается в выделении основных функций или действий и базируется на нисходящем проектировании, основанном на структурах и алгоритмах управления. Основное понятие этого подхода - алгоритмы. В объектном или объектно-ориентированном подходе в первую очередь выделяется множество основных объектов системы и впоследствии определяется множество операций над объектами.

Структурный подход к проектированию программного обеспечения. + Прочитав эту главу, вы узнаете: + Что представляет собой структурный подход к проектированию ПО. В чем заключается метод функционального моделирования SADT. Как строятся диаграммы потоков данных и "сущность-связь". + + 2.1. + сущность структурного подхода. + 2.1.1. + проблема сложности больших систем.

Сущность структурного подхода к разработке АИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, подразделяемые на задачи, и т.д. — вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимоувязаны. При разработке системы «снизу-вверх» от отдельных задач ко всей системе целостность теряется, возникают проблемы при информационной стыковке отдельных компонентов.

Сущность структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур.

Тема 1.4 Унифицированный язык моделирования UML.

UML – унифицированный язык моделирования (Unified Modeling Language) – это система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. Его можно

использовать для визуализации, спецификации, конструирования и документирования программных систем.

UML представляет собой язык визуального моделирования для ОО мо делирования. UP обеспечивает каркас процесса производства про граммного обеспечения, указывающий, как осуществлять ОО анализ и проектирование. О UP можно говорить много. В книге представлены только аспекты, имеющие непосредственное отношение к работе ОО аналитика/проекти ровщика.

Унифицированный язык моделирования (UML) был разработан с целью обеспечить единый визуальный язык с богатой семантикой и развернутым синтаксисом для проектирования и внедрения программных систем со сложной структурой и комплексным поведением. Стоит отметить, что UML применяется не только в разработке программ, но и в других сферах, например, в схематизации потоков производственных процессов. UML напоминает стандарты, используемые в других отраслях, и поддерживает диаграммы нескольких типов. В целом, диаграммы UML описывают границы, структуру и поведение как всей системы, так и отдельных объект.

Унифицированный язык моделирования UML1 (Unified Modeling Language) представляет собой язык для определения, представления, проектирования и документирования программных систем, организационно-экономических систем, технических систем и других систем различной природы. UML содержит стандартный набор диаграмм и нотаций самых разнообразных видов. UML — это преемник того поколения методов объектно-ориентированного анализа и проектирования, которые появились в конце 1980-х и начале 1990-х годов.

В настоящее время унифицированный язык моделирования - UML, простите, the UML, - является, пожалуй, самой модной технологией в области программной инженерии. Почему это так? Дело в том, что UML позволяет системным архитекторам представлять свое видение системы в виде набора стандартных диаграмм, которые, к тому же, служат отличным средством коммуникации в команде разработчиков и прекрасным помощником в общении с заказчиком. ... Назначение языка. UML - унифицированный язык моделирования.

Тема 1.5 Виды программных документов. Техническое задание

Правильное техническое задание на разработку программного обеспечения – секрет успешного проекта. В случае с системой баннерной рекламы, мы выделим такой сценарий как создание рекламного места пользователем в роли Администратор. Название сценария: Создание рекламного места. Роль: Администратор. Пример функционального

требования: «После добавления новой площадки в системе, администратор должен создать связанные с ней рекламные места.

Техническое задание — основа как простых односложных продуктов, так и высоконагруженных систем. В каждом случае сценарии функционирования должны быть предусмотрены. Любое действие пользователя должно быть предугадано, и ответом на него должен быть полезный результат. ... Учитывая, что основные принципы программного обеспечения почти не поменялись, документ еще не утратил своей актуальности. Это можно сравнить со строительством зданий: меняются материалы и конструкции, но общие понятия — фундамент, стены, перекрытия — сохраняются.

Техническое задание – это документ, в основе которого лежат требования, сформулированные на понятном (обычном, привычном) для Заказчика языке. При этом может и должна использоваться отраслевая терминология, понятная Заказчику. Никаких привязок к особенностям технической реализации быть не должно.

Техническое задание. Назначение и область применения программы, технические, технико-экономические и специальные требования, предъявляемые к программе, необходимые стадии и сроки разработки, виды испытаний. Пояснительная записка. Схема алгоритма, общее описание алгоритма и (или) функционирования программы, а также обоснование принятых технических и технико-экономических решений. Эксплуатационные документы. Сведения для обеспечения функционирования и эксплуатации программы. ... Виды программных документов, разрабатываемых на разных стадиях разработки, приведены в таблице ниже: Код вида документа. Вид документа.

Тема 1.6 Средства тестирования.

Средства тестирования позволяют улучшить такие аспекты разработки ПО, как функциональность приложений, их надежность, эффективность, а также предоставляют возможность проектирования и тестирования в режиме реального времени. Компания Interface Ltd. предлагает различные средства тестирования приложений - от комплексных до специализированных - для решения любых связанных с тестированием задач.

Средства функционального тестирования управляют автоматизированными тестами, взаимодействуя с пользовательским интерфейсом приложения. Эти инструменты обычно имеют интерфейс записи и воспроизведения, что облегчает работу тестировщиков. Однако эти тесты могут быть довольно медленными. Tricentis Tosca. Tricentis Tosca

обеспечивает модельно-ориентированную автоматизацию тестирования, нацеливаясь на разработку тест-кейсов и решение проблем.

Исторически рынок средств тестирования возник позже рынка средств разработки, однако в последнее время он весьма активно развивается. Какие же инструменты тестирования предлагаются сегодня на этом рынке? ... Исторически рынок средств тестирования возник позже рынка средств разработки, однако в последнее время он весьма активно развивается. Какие же инструменты тестирования предлагаются сегодня на этом рынке?

На рынке доступны тонны инструментов для тестирования программного обеспечения, и из-за множества вариантов выбора становится трудно найти лучший. ... На рынке доступны тонны инструментов для тестирования программного обеспечения, и из-за множества вариантов выбора становится трудно найти лучший инструмент для вашего проекта.

Средства тестирования программ. Основными задачами тестирования является проверка соответствия функциональности разработанной программы первоначальным требованиям, а также выявление ошибок, которые в явном или неявном виде проявляются во время работы программы. Среди основных работ по тестированию можно выделить следующее: · Тестирование на отказ и восстановление. · Функциональное тестирование. · Тестирование безопасности.

Раздел 2. Содержание проектной документации этапов анализа и проектирования жизненного цикла ПО

Тема 2.1 Разработка и анализ требований к программной системе

Цель: Описать и проанализировать информационную систему, распределить роли в группе разработчиков.

Теоретические сведения

Процесс создания профессионального ПО всегда является субъектом бюджетной политики организации, где оно разрабатывается, и имеет временные ограничения. Работа *руководителя программного проекта* по большому счету заключается в том, чтобы гарантировать выполнение этих бюджетных и временных ограничений с учетом бизнес-целей организации относительно разрабатываемого ПО.

Руководители проектов призваны спланировать все этапы разработки программного продукта. Они также должны контролировать ход выполнения работ и соблюдения всех требуемых стандартов. Постоянный контроль за ходом выполнения работ необходим для того, чтобы процесс разработки не выходил за временные и бюджетные ограничения. Хорошее управление не гарантирует успешного завершения проекта, но плохое управление обязательно приведет к его провалу. Это может выразиться в задержке сроков сдачи готового ПО, в превышении сметной стоимости проекта и в несоответствии готового ПО спецификации требований.

Процесс разработки ПО существенно отличается от процессов реализации технических проектов, что порождает определенные сложности в управлении программными проектами:

1. *Программный продукт нематериален.* Программное обеспечение нематериально, его нельзя увидеть или потрогать. Руководитель программного проекта не видит процесс "роста" разрабатываемого ПО. Он может полагаться только на документацию, которая фиксирует процесс разработки программного продукта.
2. *Не существует стандартных процессов разработки ПО.* На сегодняшний день не существует четкой зависимости между процессом создания ПО и типом создаваемого программного продукта. Другие технические дисциплины имеют длительную историю, процессы разработки технических изделий многократно опробованы и проверены. Процессы создания большинства технических систем хорошо изучены. Изучением же процессов создания ПО специалисты занимаются только последнее время. Поэтому пока нельзя точно предсказать, на каком этапе процесса разработки ПО могут возникнуть проблемы, угрожающие всему программному проекту.
3. *Большие программные проекты - это часто "одноразовые" проекты.* Большие программные проекты, как правило, значительно отличаются от проектов, реализованных ранее. Поэтому, чтобы уменьшить неопределенность в планировании проекта, руководители проектов должны обладать очень большим практическим опытом. Но постоянные технологические изменения в компьютерной технике и коммуникационном оборудовании обесценивают предыдущий опыт. Знания и навыки, накопленные опытом, могут не востребоваться в новом проекте.

Перечисленные отличия могут привести к тому, что реализация проекта выйдет из временного графика или превысит бюджетные ассигнования. Программные системы зачастую оказываются новинками как в "идеологическом", так и в техническом плане. Поэтому, предвидя возможные проблемы в реализации программного проекта, следует всегда помнить, что многим из них свойственно выходить за рамки временных и бюджетных ограничений.

Процесс управления разработкой программного обеспечения

Невозможно описать и стандартизировать все работы, выполняемые в проекте по созданию ПО. Эти работы весьма существенно зависят от организации, где выполняется разработка ПО, и от типа создаваемого программного продукта. Но всегда можно выделить следующие:

- Написание предложений по созданию ПО.
- Планирование и составление графика работ по созданию ПО.
- Оценивание стоимости проекта.
- Подбор персонала.
- Контроль за ходом выполнения работ.
- Написание отчетов и представлений.

Первая стадия программного проекта может состоять из написания предложений по реализации этого проекта. Предложения должны содержать описание целей проектов и способов их достижения. Они также обычно включают в себя оценки финансовых и временных затрат на выполнение проекта. При необходимости здесь могут приводиться обоснования для передачи проекта на выполнение сторонней организации или команде разработчиков.

Написание предложений — очень ответственная работа, так как для многих организаций вопрос о том, будет ли проект выполняться самой организацией или разрабатываться по контракту сторонней компанией, является критическим. Не существует каких-либо рекомендаций по написанию предложений, многое здесь зависит от опыт.

На этапе *планирования проекта* определяются процессы, этапы и полученные на каждом из них результаты, которые должны привести к выполнению проекта. Реализация этого плана приведет к достижению целей проекта. Определение стоимости проекта напрямую связано с его планированием, поскольку здесь оцениваются ресурсы, требующиеся для выполнения плана.

Контроль за ходом выполнения работ (мониторинг проекта) — это непрерывный процесс, продолжающийся в течение всего срока реализации проекта. Руководитель должен постоянно отслеживать ход реализации проекта и сравнивать фактические и плановые показатели выполнения работ с их стоимостью. Хотя многие организации имеют механизмы формального мониторинга работ, опытный руководитель может составить ясную картину о стадии развитии проекта просто путем неформального общения с разработчиками.

Неформальный мониторинг часто помогает обнаружить потенциальные проблемы, которые в явном виде могут обнаружиться позднее. Например, ежедневное обсуждение хода выполнения работ может выявить отдельные недоработки в создаваемом программном продукте. Вместо ожидания отчетов, в которых будет отражен факт "пробуксовки" графика работ, можно обсудить со специалистами намечающиеся программистские проблемы и не допустить срыва графика работ.

В течение реализации проекта обычно происходит несколько формальных контрольных проверок хода выполнения работ по созданию ПО. Такие проверки должны дать общую картину хода реализации проекта в целом и показать, насколько уже разработанная часть ПО соответствует целям проекта.

Время выполнения больших программных проектов может занимать несколько лет. В течение этого времени цели и намерения организации, заказавшей программный проект, могут существенно измениться. Может оказаться, что разрабатываемый программный продукт стал уже ненужным либо исходные требования к создаваемому ПО просто устарели и их необходимо кардинально менять. В такой ситуации руководство организации-разработчика может принять решение о прекращении разработки ПО или об изменении проекта в целом с тем, чтобы учесть изменившиеся цели и намерения организации-заказчика.

Руководители проектов обычно обязаны сами *подбирать исполнителей* для своих проектов. В идеальном случае профессиональный уровень исполнителей должен соответствовать той работе, которую они будут выполнять в ходе реализации проекта. Однако во многих случаях руководители должны полагаться на команду разработчиков, которая далека от идеальной. Такая ситуация может быть вызвана следующими причинами:

1. Бюджет проекта не позволяет привлечь высококвалифицированный персонал. В таком случае за меньшую плату привлекаются менее квалифицированные специалисты.
2. Бывают ситуации, когда невозможно найти специалистов необходимой квалификации как в самой организации-разработчике, так и вне ее. Например, в организации "лучшие люди" могут быть уже заняты в других проектах.
3. Организация хочет повысить профессиональный уровень своих работников. В этом случае она может привлечь к участию в проекте неопытных или недостаточно квалифицированных работников, чтобы они приобрели необходимый опыт и поучились у более опытных специалистов.

Таким образом, почти всегда подбор специалистов для выполнения проекта имеет определенные ограничения и не является свободным. Вместе с тем необходимо, чтобы хотя бы несколько членов группы разработчиков имели квалификацию и опыт, достаточные для работы над данным проектом. В противном случае невозможно избежать ошибок в разработке ПО.

Руководитель проекта обычно обязан посылать *отчеты* о ходе его выполнения как заказчику, так и подрядным организациям. Это должны быть краткие документы,

основанные на информации, извлекаемой из подробных' отчетов о проекте. В этих отчетах должна быть та информация, которая позволяет четко оценить степень готовности создаваемого программного продукта.

В рамках курса «Технология разработки программного обеспечения» выделены следующие роли в группе по разработке ПО:

- Руководитель – общее руководство проектом, написание документации, общение с заказчиком ПО
- Системный аналитик – разработка требований (составление технического задания, проекта программного обеспечения)
- Тестер – составление плана тестирования и аттестации готового ПО (продукта), составление сценария тестирования, базовый пример, проведение мероприятий по плану тестирования
- Разработчик – моделирование компонент программного обеспечения, кодирование

Планирование проекта разработки программного обеспечения

Эффективное управление программным проектом напрямую зависит от правильного планирования работ, необходимых для его выполнения. План помогает руководителю предвидеть проблемы, которые могут возникнуть на каких-либо этапах создания ПО, и разработать превентивные меры для их предупреждения или решения. План, разработанный на начальном этапе проекта, рассматривается всеми его участниками как руководящий документ, выполнение которого должно привести к успешному завершению проекта. Этот первоначальный план должен максимально подробно описывать все этапы реализации проекта.

Процесс планирования начинается, исходя из описания системы, с определения проектных ограничений (временные ограничения, возможности наличного персонала, бюджетные ограничения и т.д.). Эти ограничения должны определяться параллельно с оцениванием проектных параметров, таких как структура и размер проекта, а также распределением функций среди исполнителей. Затем определяются этапы разработки и то, какие результаты документация, прототипы, подсистемы или версии программного продукта) должны быть получены по окончании этих этапов. Далее начинается циклическая часть планирования. Сначала разрабатывается график работ по выполнению проекта или дается разрешение на продолжение использования ранее созданного графика. После этого проводится контроль выполнения работ и отмечаются расхождения между реальным и плановым ходом работ.

Далее, по мере поступления новой информации о ходе выполнения проекта, возможен пересмотр первоначальных оценок параметров проекта. Это, в свою очередь, может привести к изменению графика работ. Если в результате этих изменений нарушаются сроки завершения проекта, должны быть пересмотрены (и согласованы с заказчиком ПО) проектные ограничения.

Конечно, большинство руководителей проектов не думают, что реализация их проектов пройдет гладко, без всяких проблем. Желательно описать возможные проблемы еще до того, как они проявят себя в ходе выполнения проекта. Поэтому лучше составлять "пессимистические" графики работ, чем "оптимистические". Но, конечно, невозможно построить план, учитывающий все, в том числе случайные, проблемы и задержки выполнения проекта, поэтому и возникает необходимость периодического пересмотра проектных ограничений и этапов создания программного продукта.

План проекта должен четко показать ресурсы, необходимые для реализации проекта, разделение работ на этапы и временной график выполнения этих этапов. В некоторых организациях план проекта составляется как единый документ, содержащий все виды планов, описанных выше. В других случаях план проекта описывает только технологический процесс создания ПО. В таком плане обязательно присутствуют ссылки на планы других видов, но они разрабатываются отдельно от плана проекта.

Детализация планов проектов очень разнится в зависимости от типа разрабатываемого программного продукта и организации-разработчика. Но в любом случае большинство планов содержат следующие разделы.

1. *Введение*. Краткое описание целей проекта и проектных ограничений (бюджетных, временных и т.д.), которые важны для управления проектом.
2. *Организация выполнения проекта*. Описание способа подбора команды разработчиков и распределение обязанностей между членами команды.
3. *Анализ рисков*. Описание возможных проектных рисков, вероятности их проявления и стратегий, направленных на их уменьшение.
4. *Аппаратные и программные ресурсы, необходимые для реализации проекта*. Перечень аппаратных средств и программного обеспечения, необходимого для разработки программного продукта. Если аппаратные средства требуется закупать, приводится их стоимость совместно с графиком закупки и поставки.
5. *Разбиение работ на этапы*. Процесс реализации проекта разбивается на отдельные процессы, определяются этапы выполнения проекта, приводится описание результатов ("выходов") каждого этапа и контрольные отметки.
6. *График работ*. В этом графике отображаются зависимости между отдельными процессами (этапами) разработки ПО, оценки времени их выполнения и распределение членов команды разработчиков по отдельным этапам.
7. *Механизмы мониторинга и контроля за ходом выполнения проекта*. Описываются предоставляемые руководителем отчеты о ходе выполнения работ, сроки их предоставления, а также механизмы мониторинга всего проекта.

План должен регулярно пересматриваться в процессе реализации проекта. Одни части плана, например график работ, изменяются часто, другие более стабильны. Для внесения изменений в план требуется специальная организация документопотока, позволяющая отслеживать эти изменения.

Общие сведения о требованиях к информационным системам

Проблемы, которые приходится решать специалистам в процессе создания программного обеспечения, очень сложны. Природа этих проблем не всегда ясна, особенно если разрабатываемая программная система инновационная. В частности, трудно чётко описать те действия, которые должна выполнять система. Описание функциональных возможностей и ограничений, накладываемых на систему, называется требованиями к этой системе, а сам процесс формирования, анализа, документирования и проверки этих функциональных возможностей и ограничений – разработкой требований.

Требования подразделяются на пользовательские и системные. Пользовательские требования – это описание на естественном языке (плюс поясняющие диаграммы) функций, выполняемых системой, и ограничений, накладываемых на неё. Системные требования – это описание особенностей системы (архитектура системы, требования к параметрам оборудования и т.д.), необходимых для эффективной реализации требований пользователя.

Первые шаги по разработке требований к информационным системам - анализ осуществимости

Разработка требований — это процесс, включающий мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию системных требований. Для новых программных систем процесс разработки требований должен начинаться с анализа осуществимости. Началом такого анализа является общее описание системы и ее назначения, а результатом анализа — отчет, в котором должна быть четкая рекомендация, продолжать или нет процесс разработки требований проектируемой системы. Другими словами, анализ осуществимости должен осветить следующие вопросы.

1. Отвечает ли система общим и бизнес-целям организации-заказчика и организации-разработчика?
2. Можно ли реализовать систему, используя существующие на данный момент технологии и не выходя за пределы заданной стоимости?

3. Можно ли объединить систему с другими системами, которые уже эксплуатируются?

Критическим является вопрос, будет ли система соответствовать целям организации. Если система не соответствует этим целям, она не представляет никакой ценности для организации. В то же время многие организации разрабатывают системы, не соответствующие их целям, либо не совсем ясно понимая эти цели, либо под влиянием политических или общественных факторов.

Выполнение анализа осуществимости включает сбор и анализ информации о будущей системе и написание соответствующего отчета. Сначала следует определить, какая именно информация необходима, чтобы ответить на поставленные выше вопросы. Например, эту информацию можно получить, ответив на следующее:

1. Что произойдет с организацией, если система не будет введена в эксплуатацию?
2. Какие текущие проблемы существуют в организации и как новая система поможет их решить?
3. Каким образом система будет способствовать целям бизнеса?
4. Требует ли разработка системы технологии, которая до этого не использовалась в организации?

Далее необходимо определить источники информации. Это могут быть менеджеры отделов, где система будет использоваться, разработчики программного обеспечения, знакомые с типом будущей системы, технологи, конечные пользователи и т.д.

После обработки собранной информации готовится отчет по анализу осуществимости создания системы. В нем должны быть даны рекомендации относительно продолжения разработки системы. Могут быть предложены изменения бюджета и графика работ по созданию системы или предъявлены более высокие требования к системе.

Порядок выполнения работы

1. Изучить предлагаемый теоретический материал.
2. Составить подробное описание информационной системы.
3. На основании описания системы провести анализ осуществимости. В ходе анализа ответить на вопросы:

- *Что произойдет с организацией, если система не будет введена в эксплуатацию?*
- *Какие текущие проблемы существуют в организации и как новая система поможет их решить?*
- *Каким образом система будет способствовать целям бизнеса?*
- *Требует ли разработка системы технологии, которая до этого не использовалась в организации?*

Результатом анализа должно явиться заключение о возможности реализации проекта.

4. Распределить роли в группе (руководитель проекта-разработчик, системный аналитик-разработчик, тестер-разработчик).
5. Заполнить разделы плана:

- *Введение*
- *Организация выполнения проекта*
- *Анализ рисков*

Разделы должны содержать рекомендации относительно разработки системы, базовые предложения по объёму требуемого бюджета, числу разработчиков, времени и требуемому программному обеспечению.

Содержание отчета

В отчете следует указать:

1. Цель работы

2. Введение. Краткое описание целей проекта и проектных ограничений (бюджетных, временных и т.д.), которые важны для управления проектом

3. Описание информационной системы (ПО) - наличие заключения о возможности реализации проекта, содержащего рекомендации относительно разработки системы, базовые предложения по объёму требуемого бюджета, числу разработчиков, времени и требуемому программному обеспечению

4. Анализ осуществимости, указать возможные проблемы и пути их решения.

5. Роли участников группы разработки ПО.

Практическая работа 2. Проектирование программной системы

Цель работы:познакомить студентов с методом проектирования системы путем CRC-карт.

Теоретические сведения

Важным этапом создания программного обеспечения является проектирование. На этом шаге закладывается архитектура системы.

Одним из способов проектирования является метод CRC-карточек. Этот метод проектирования является составляющей UML-проектирования.

Шаг первый. Для первоначального понимания структуры программы строится диаграмма вариантов использования: выявляются акторы (люди или системы, между которыми происходит взаимодействие), прецеденты системы (действия выполняемые системой для реализации общения акторов). Рассмотрим сказанное на примере банкомата.

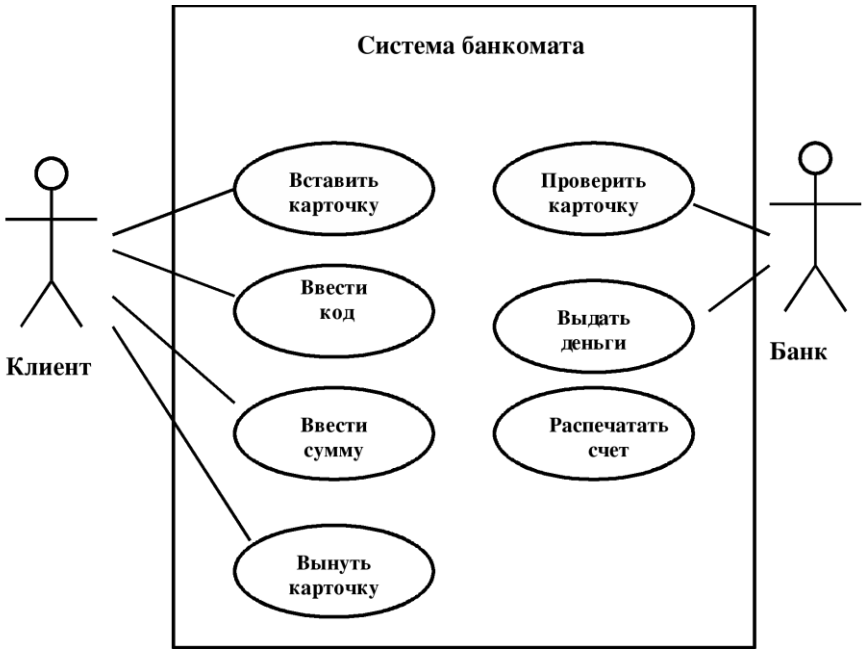


Рис. 1.1. Диаграмма вариантов использования «Банкомат»

На самом деле прецедентов может быть очень много. Допустим: проверить пароль, контролировать транзакции передачи данных, выдать информацию на экран и т.д.

Эта диаграмма дает понять, что будет делать система, как она будет функционировать. Диаграмма вариантов использования (прецедентов) бывает также очень полезна также для общения с заказчиком – она позволяет показать наиболее значимые действия системы и проверить: правильно ли вы поняли заказчика и значимость отдельных функций для него.

Шаг второй. На этом этапе выявляют классы, которые необходимо будет создать в программе для реализации системы. В случае банкомата это: клиент, банк, служба безопасности банка, сам банкомат и т.д.. Придумать можно много (таймер, счетчик купюр, карточка и т.д.).

Далее оформляются CRC-карты. Это листки бумаги 10х15. Они разделены на 3 части и выглядят следующим образом:

Название класса	
Действия, которые он выполняет (всегда начинают с глагола)	Классы, с которыми данный класс обменивается и информацией

На примере того же банкомата:

Клиент	
1. Вставить карточку в банкомат 2. Вводит пароль 3. Указывает тип операции 4. Вводит сумму 5. Получает деньги 6. Вынимает карточку	Банкомат

Банкомат	
1. Отображает информацию для клиента 2. Передает информацию в банк 3. Отсчитывает купюры 4. Распечатывает счет	Клиент Банк Служба безопасности банка

Служба безопасности банка	
1. Проверяет пароль 2. Проверяет подлинность карточки 3. Идентифицирует клиента 4. Следит за правильностью транзакций операций с деньгами	Банк Банкомат

Банк	
1. Проверяет возможность выдачи средств 2. Сообщает о наличии денег 3. Выдает информацию об остатке 4. Хранит информацию о счете клиента	Банкомат Служба безопасности банка

Шаг третий. Для проверки достаточности или избыточности придуманных классов, а также корректности их взаимодействия строится диаграмма взаимодействия:

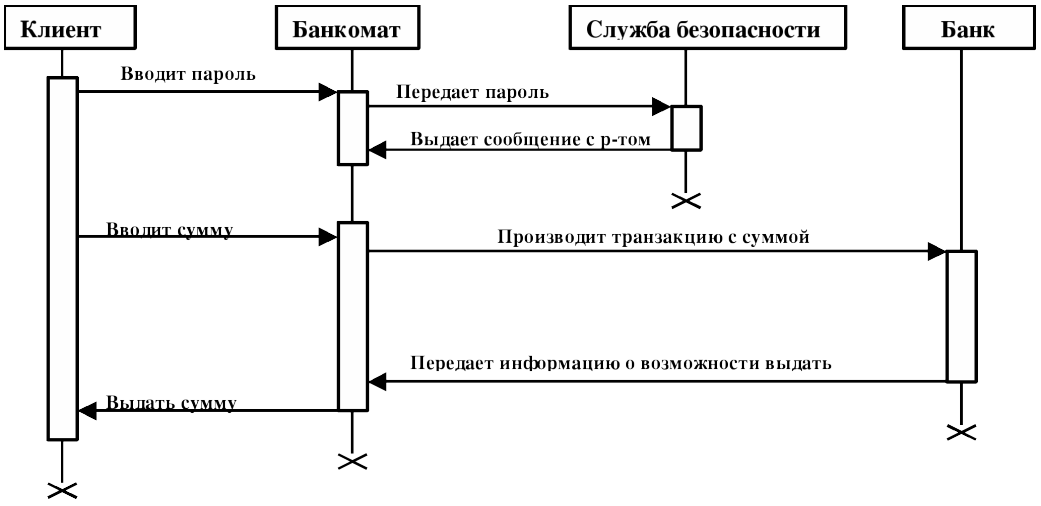


Рис. 1.2. Диаграмма взаимодействия

Метод CRC-карточек позволяет провести также инсценировку работы системы. Для этого достаточно раздать карточки с классами участникам проекта. После этого начать ролевую игру. Первый участник встает и читает действие, совершаемое его классом. Другие участники, исходя из своих карточек, сообщают об ответной реакции других классов. Если в какой-то момент реакции не последует, то это признак несовершенства проекта системы. Такая игра может подсказать и об избыточности проекта.

Задание:

Разработать проект системы по методу CRC-карт для одного из следующих вариантов.

Варианты задания:

1. Заказ билетов в аэропорту.
2. Электронный магазин.
3. Отправка sms.
4. Пропускная система.
5. Компьютерная система тестирования для оценки знаний студентов.
6. Заказ билетов в театральной кассе.
7. Телефонный коммутатор.
8. Система учета успеваемости студентов деканатом.
9. Записная книжка.
10. Пропускная система биоконтроля.
11. Автомат платежной системы.
12. Электронная почта.

Практическая работа 3. Техническое задание

Цель работы: научиться составлять техническое задание для программного обеспечения.

Теоретическая часть

ГОСТ 19.201—78. Настоящий стандарт устанавливает порядок построения и оформления технического задания на разработку программы или программного изделия для вычислительных машин, комплексов и систем независимо от их назначения и области применения.

Техническое задание должно содержать следующие разделы:

- название программы и область применения;
- основание для разработки;
- назначение разработки;
- технические требования к программе или программному изделию;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки;
- приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

Содержание разделов технического задания

1. В разделе «Наименование и область применения» указывают наименование, краткую характеристику области применения программы или программного изделия и объекта, в котором используют программу или программное изделие.

2. В разделе «Основание для разработки» должны быть указаны:

- документ (документы), на основании которых ведется разработка;
- организация, утвердившая этот документ, и дата его утверждения;

• наименование и (или) условное обозначение темы разработки.

3. В разделе «Назначение разработки» должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.

4. Раздел «Технические требования к программе или программному изделию» должен содержать следующие подразделы:

- требования к функциональным характеристикам;
- требования к надежности;
- условия эксплуатации;
- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;
- требования к маркировке и упаковке;
- требования к транспортированию и хранению;
- специальные требования.

4.1. В подразделе «Требования к функциональным характеристикам» должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т. п.

4.2. В подразделе «Требования к надежности» должны быть указаны требования к обеспечению надежного функционирования (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т. п.).

4.3. В подразделе «Условия эксплуатации» должны быть указаны условия эксплуатации (температура окружающего воздуха, относительная влажность и т. п. для выбранных типов носителей данных), при которых должны обеспечиваться заданные характеристики, а также вид обслуживания, необходимое количество и квалификация персонала.

4.4. В подразделе «Требования к составу и параметрам технических средств» указывают необходимый состав технических средств с указанием их технических характеристик.

4.5. В подразделе «Требования к информационной и программной совместимости» должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования. При необходимости должна обеспечиваться защита информации и программ.

4.6. В подразделе «Требования к маркировке и упаковке» в общем случае указывают требования к маркировке программного изделия, варианты и способы упаковки.

4.7. В подразделе «Требования к транспортированию и хранению» должны быть указаны для программного изделия условия транспортирования, места хранения, условия хранения, условия складирования, сроки хранения в различных условиях.

5. В разделе «Технико-экономические показатели» должны быть указаны: ориентировочная экономическая эффективность, предполагаемая годовая потребность, экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами.

6. В разделе «Стадии и этапы разработки» устанавливают необходимые стадии разработки, этапы и содержание работ (перечень программных документов, которые должны быть разработаны, согласованы и утверждены), а также, как правило, сроки разработки и определяют исполнителей.

7. В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.

8. В приложениях к техническому заданию при необходимости приводят:

- перечень научно-исследовательских и других работ, обосновывающих разработку;
- схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке;
- другие источники разработки.

Задание

Составить техническое задание в соответствии с ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.

Требования к отчёту

Отчёт должен содержать титульный лист, аннотацию, содержание и основную часть, оформленную в соответствии с ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению

Варианты заданий

1. Разработать программный модуль «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.
2. Разработать программный модуль «Личные дела студентов». Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.
3. Разработать программный модуль «Решение комбинаторно-оптимизационных задач». Модуль должен содержать алгоритмы поиска цикла минимальной длины (задача коммивояжера), поиска кратчайшего пути и поиска минимального связывающего дерева.
4. Разработать программный модуль «Обработка матрицы». Модуль должен содержать алгоритмы поиска сумм и произведения элементов матрицы по строкам и столбцам, а также вычисление средних, минимальных и максимальных величин в матрице.
5. Разработать приложение Windows «Органайзер». Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц и организаций, а также расписания, встреч и др. Приложение предназначено для любых пользователей компьютера.
6. Разработать приложение Windows «Калькулятор». Приложение предназначено для любых пользователей и должно содержать все арифметические операции (с соблюдением приоритетов) и желательно (но не обязательно) несколько математических функций.
7. Разработать программный модуль «Кафедра», содержащий сведения о сотрудниках кафедры (ФИО, должность, ученая степень, дисциплины, нагрузка, общественная работа, совместительство и др.). Модуль предназначен для использования сотрудниками отдела кадров и деканата.
8. Разработать программный модуль «Лаборатория», содержащий сведения о сотрудниках лаборатории (ФИО, пол, возраст, семейное положение, наличие детей, должность, ученая степень). Модуль предназначен для использования сотрудниками профкома и отдела кадров.
9. Разработать программный модуль «Химчистка». При записи на обслуживание заполняется заявка, в которой указываются ФИО владельца, описание изделия, вид услуги, дата приема заказа и стоимость услуги. После выполнения работ распечатывается квитанция.
10. Разработать программный модуль «Учет нарушений правил дорожного движения». Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.
11. Разработать программный модуль «Картотека автомагазина», предназначенный для использования работниками агентства. В базе содержатся сведения об автомобилях (марка, объем двигателя, дата выпуска и др.). При поступлении заявки на покупку производится поиск подходящего варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.

12. Разработать программный модуль «Картотека абонентов АТС». Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и повременной). Считается, что повременная оплата местных телефонных разговоров уже введена.

13. Разработать программный модуль «Автокасса», содержащий сведения о наличии свободных мест на автобусные маршруты. В базе должны содержаться сведения о номере рейса, маршруте, водителе, типе автобуса, дате и времени отправления, а также стоимости билетов. При поступлении заявки на билеты программа производит поиск подходящего рейса.

14. Разработать программный модуль «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена). Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.

15. Разработать программный модуль «Автостоянка». В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.

16. Разработать программный модуль «Кадровое агентство», содержащий сведения о вакансиях и резюме. Программный модуль предназначен как для поиска сотрудника, отвечающего требованиям руководителей фирмы, так и для поиска подходящей работы.

Практическая работа 4. Анализ выбранного стиля программирования

Задание на лабораторную работу: разработать ПС, представленное по правилам хорошего стиля программирования. Сделайте вывод о проделанной работе.

Вариант №1

Сгенерировать две последовательности по 50 случайных чисел с равномерным распределением в диапазоне [1..6]. Полученные последовательности расположить в одном массиве по возрастанию. Вычислить среднее значение и дисперсию для полученной последовательности и вывести на печать в виде гистограммы, разделив диапазон на десять интервалов.

Вариант №2

Сгенерировать последовательность 100 случайных чисел с равномерным законом распределения в диапазоне от 0 до 100. Упорядочить полученную последовательность по возрастанию. Образовать новую последовательность, состоящую из разности соседних элементов последовательности X_i-X_{i-1}. Для полученной последовательности определить среднее значение, дисперсию и вывести на печать гистограмму распределения, разделив диапазоны на 10 интервалов.

Вариант №3

Сгенерировать последовательность 100 случайных чисел с нормальным законом распределения (*mx*=1, *dx*=1). Упорядочить полученную последовательность, расположив элементы по возрастанию. Образовать новую последовательность, состоящую из разности соседних элементов X_i-X_{i-1}. Для полученной последовательности вычислить среднее значение, дисперсию и вывести ее на печать в виде гистограммы, разбив диапазон на 10 интервалов.

Вариант № 4

Сгенерировать последовательность 100 случайных чисел X_i =0,1. Сформировать новую последовательность, состоящую из элементовλс экспоненциальным законом распределения с параметром $y_i=y_{i-1}+x_i$ ($y_1=x_1$). Для полученной последовательности вычислить среднее значение, дисперсию и вывести ее на печать в виде гистограммы. Разделив диапазон на 10 интервалов.

Вариант №5

Сгенерировать последовательность 100 случайных чисел X_i с равномерным законом распределения в диапазоне от 1 до 10. Сформировать новую последовательность, состоящую из элементов $y_i=y_{i-1}+x_i$ ($y_1=x_1$). Для полученной последовательности вычислить среднее значение, дисперсию, вывести ее на печать в виде гистограммы, разделив диапазон на 10 интервалов.

Вариант №6

Сгенерировать последовательность 100 случайных чисел X_i с нормальным законом распределения (*mx=5, dx=2*). Сформировать новую последовательность, состоящую из элементов $y_i=y_{i-1}+x_i$ ($y_1=x_1$). Для полученной последовательности вычислить среднее значение, дисперсию и вывести ее на печать в виде гистограммы, разбив диапазон на десять интервалов.

Вариант №7

Сгенерировать последовательность 100 случайных чисел с экспоненциальным законом =0,5. Упорядочить полученнуюλраспределения с параметром последовательность, расположив элементы по возрастанию. Образовать новую последовательность, состоящую из разности соседних элементов X_i-X_{i-1}. Для полученной последовательности вычислить среднее значение, дисперсию и вывести ее на печать в виде гистограммы, разбив диапазон на десять интервалов.

Вариант №8

Сгенерировать три последовательности по 30 случайных чисел каждая. Числа в каждой последовательности равномерно распределены в диапазонах от 0 до 5, от 2 до 8, от 4 до 10. Свести их в один массив, расположив по возрастанию. Для сформированного массива вычислить среднее значение, дисперсию и вывести результаты на печать в виде гистограммы, разбив диапазон на 10 интервалов.

Вариант 9

Сгенерировать три последовательности по 30 случайных чисел. Числа в каждой последовательности распределены по нормальному закону с параметрами *mx=2, dx=4; mx=3, dx=3; mx=4, dx=4*. Свести все числа в один массив, упорядочив по возрастанию. Для сформированного массива вычислить среднее значение, дисперсию и вывести на печать результаты в виде гистограммы, разбив диапазон на 10 интервалов.

Вариант №10

Сгенерировать три последовательности по 30 случайных чисел. В каждой последовательности числа распределены по =4. Свести числа вλ=3, λ=2, λэкспоненциальному закону с параметрами один массив, упорядочив их по возрастанию. Для сформированного массива вычислить среднее значение, дисперсию и вывести результаты на печать в виде гистограммы, разбив диапазон на 10 интервалов.

Вариант №11

Сгенерировать последовательность из 50 случайных чисел с равномерным законом распределения в диапазоне от 0 до 10 и 50 случайных чисел с нормальным законом распределения *mx=5, dx=4*. Все числа свести в массив, расположив их по возрастанию. Вычислить среднее значение, дисперсию и вывести результаты на печать в виде гистограммы, разбив последовательность чисел на 10 интервалов.

Вариант №12

Сгенерировать последовательность из 50 случайных чисел с нормальным законом распределения *mx=5, dx=4* =5.λи последовательность из 50 случайных чисел с экспоненциальным законом распределения с параметром Все числа свести в массив, расположив их по возрастанию. Вычислить среднее значение, дисперсию и вывести результаты на печать в виде гистограммы, разбив последовательность чисел на 10 интервалов.

Вариант № 13

Сгенерировать последовательность 100 случайных чисел с экспоненциальным законом =2. Вычислить среднее значение и дисперсию.λраспределения с параметром Построить гистограмму для полученного распределения, разбив числа на 10 интервалов.

Вариант №14

Сгенерировать последовательность 80 случайных чисел X_i с нормальным законом распределения (*mx=3, dx=4*). Сформировать новую последовательность, состоящую из элементов $y_i=y_{i-1}+x_i$ ($y_1=x_1$). Для полученной последовательности вычислить среднее значение, дисперсию и вывести ее на печать в виде гистограммы, разбив диапазон на десять интервалов.

Вариант № 15

Сгенерировать последовательность 60 случайных =0,8.λчисел с экспоненциальным законом распределения с параметром Упорядочить полученную последовательность, расположив элементы по возрастанию. Образовать новую последовательность, состоящую из разности соседних элементов X_i-X_{i-1}. Для полученной последовательности вычислить среднее значение, дисперсию и вывести ее на печать в виде гистограммы, разбив диапазон на десять интервалов.

Вариант №16

Сгенерировать 4 последовательности по 30 случайных чисел каждая. Числа в каждой последовательности равномерно распределены в диапазонах от 0 до 5, от 2 до 8, от 4 до 10, от 6 до 12. Свести их в один массив, расположив по возрастанию. Для

сформированного массива вычислить среднее значение, дисперсию и вывести результаты на печать в виде гистограммы, разбив диапазон на 10 интервалов.

Вариант №17

Сгенерировать две последовательности по 50 случайных чисел. В каждой последовательности числа распределены по экспоненциальному закону с =3. Свести числа в один массив, упорядочив их поλ=2, λпараметрами возрастанию. Для сформированного массива вычислить среднее значение, дисперсию и вывести результаты на печать в виде гистограммы, разбив диапазон на 10 интервалов.

Вариант №18

Сгенерировать последовательность из 60 случайных чисел с равномерным законом распределения в диапазоне от 0 до 8 и 50 случайных чисел с нормальным законом распределения *mx=3, dx=4*. Все числа свести в массив, расположив их по возрастанию. Вычислить среднее значение, дисперсию и вывести результаты на печать в виде гистограммы, разбив последовательность чисел на 10 интервалов.

Вариант №19

Сгенерировать последовательность из 30 случайных чисел с нормальным законом распределения *mx=2, dx=4* и последовательность из 70 случайных чисел с экспоненциальным законом =4. Все числа свести в массив, расположив ихλраспределения с параметром по возрастанию. Вычислить среднее значение, дисперсию и вывести результаты на печать в виде гистограммы, разбив последовательность чисел на 10 интервалов.

Вариант № 20

Сгенерировать последовательность 100 случайных чисел с экспоненциальным законом =3. Вычислить среднее значение и дисперсию.λраспределения с параметром Построить гистограмму для полученного распределения, разбив числа на 10 интервалов.Вариант №21Сгенерировать две последовательности .]1,5[по 40 случайных чисел с равномерным распределением в диапазоне Полученные последовательности расположить в одном массиве по возрастанию. Вычислить среднее значение и дисперсию для полученной последовательности и вывести на печать в виде гистограммы, разделив диапазон на десять интервалов.

Практическая работа 5. Разработка проекта программного обеспечения

***Цель работы*:** изучить вопросы проектирования программного обеспечения

Теоретическая часть

ПРОЕКТ ТЕХНИЧЕСКИЙ - образ намеченного к созданию объекта, представленный в виде его описания, схем, чертежей, расчетов, обоснований, числовых показателей.

Технический проект

Цель технического проекта — определение основных методов, используемых при создании информационной системы, и окончательное определение ее сметной стоимости.

Техническое проектирование подсистем осуществляется в соответствии с утвержденным техническим заданием.

Технический проект программной системы подробно описывает:

•выполняемые функции и варианты их использования;

•соответствующие им документы;

•структуры обрабатываемых баз данных;

•взаимосвязи данных;

•алгоритмы их обработки.

Технический проект должен включать данные об объемах и интенсивности потоков обрабатываемой информации, количестве пользователей программной системы, характеристиках оборудования и программного обеспечения, взаимодействующего с проектируемым программным продуктом.

При разработке технического проекта оформляются:

•ведомость технического проекта. Общая информация по проекту;

•пояснительная записка к техническому проекту. Вводная информация, позволяющая ее потребителю быстро освоить данные по конкретному проекту;

•описание систем классификации и кодирования;

•перечень входных данных (документов). Перечень информации, которая используется как входящий поток и служит источником накопления;

•перечень выходных данных (документов). Перечень информации, которая используется для анализа накопленных данных;

•описание используемого программного обеспечения. Перечень программного обеспечения и СУБД, которые планируется использовать для создания информационной системы;

•описание используемых технических средств. Перечень аппаратных средств, на которых планируется работа проектируемого программного продукта;

•проектная оценка надежности системы. Экспертная оценка надежности с выявлением наиболее благополучных участков программной системы и ее узких мест;

•ведомость оборудования и материалов. Перечень оборудования и материалов, которые потребуются в ходе реализации проекта.

Структурная схема

Структурной называют схему, отражающую состав и взаимодействие по управлению частями разрабатываемого программного обеспечения. Структурная схема определяется архитектурой разрабатываемого ПО.

Функциональная схема

Функциональная схема — это схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств.

Разработка алгоритмов

Метод пошаговой детализации реализует нисходящий подход к программированию и предполагает пошаговую разработку алгоритма.

Структурные карты

Методика структурных карт используется на этапе проектирования ПО для того, чтобы продемонстрировать, каким образом программный продукт выполняет системные требования. Структурные карты Константайна предназначены для описания отношений между модулями.

Техника структурных карт Джексона основана на методе структурного программирования Джексона, который выявляет соответствие между структурой потоков данных и структурой программы. Основное внимание в методе сконцентрировано на соответствии входных и выходных потоков данных.

Задание

1.На основе технического задания из лабораторной работы № 5 и спецификаций из лабораторной работы № 1 - 2 разработать

- Структурной схемы программного продукта.
- Функциональной схемы.
- Алгоритма программы.
- Дерева диалога.

2.в виде псевдокода уточненные алгоритмы программ, составляющих заданный программный модуль. Использовать метод пошаговой детализации.

3.На основе уточненных и доработанных алгоритмов разработать структурную схему программного продукта.

4.Разработать функциональную схему программного продукта.

5. Разработать интерфейс системы в виде дерева диалога.

5.Оформить результаты, используя MS Office в виде технического проекта.

6.Сдать и защитить работу.

Защита отчета по лабораторной работе

Отчет по лабораторной работе должен быть оформлен на основании СТП и состоять из следующих структурных элементов:

1. титульный лист;
2. текстовая часть;
3. приложение: разработанный рабочий проект.

Текстовая часть отчета должна включать пункты:

- условие задачи;
- порядок выполнения.

Отчет по лабораторной работе должен состоять из:

1.Законченного технического проекта программного модуля.

Защита отчета по лабораторной работе заключается в предъявлении преподавателю полученных результатов (на экране монитора), демонстрации полученных навыков и ответах на вопросы преподавателя.

Контрольные вопросы

1.Назовите этапы разработки программного обеспечения.

2.В чем заключается проектирование программного обеспечения?

3.Перечислите составляющие технического проекта.

4.Охарактеризуйте структурный подход к программированию.

5.Из чего состоят структурная и функциональная схемы?

6.Охарактеризуйте метод пошаговой детализации при составлении алгоритмов программ.

7.Приведите понятие псевдокода.

8.В чем заключается методика Константайна

9.В чем заключается методика Джексона?

Практическая работа 6. Применение методов структурирования программ. Использование структурированных типов данных строки и множества при решении задач.

Цель: Научиться использовать структурированные типы данных строки и множества при решении задач на языке Turbo Pascal.

Теоретические сведения:

В языках программирования вспомогательные алгоритмы называются *подпрограммами.* В Паскале различаются две разновидности подпрограмм: *процедуры* и *функции.*

При вызове подпрограммы (процедуры или функции), работа главной программы на некоторое время приостанавливается и начинает выполняться вызванная подпрограмма. Она обрабатывает данные, преданные ей из главной программы. По завершении выполнения программа – функция возвращает главной программе результат (программа – процедура не возвращает явно результирующего значения).

Передача данных из главной программы в подпрограмму и возврат результата выполнения функции осуществляются с помощью параметров.

Параметром называется переменная, которой присваивается некоторое значение в рамках указанного применения. Различают и **формальные параметры**- параметры, определенные в заголовке подпрограммы, и **фактические параметры**- выражения, задающие конкретные значения при обращении к подпрограмме. При обращении к подпрограмме ее формальные параметры замещаются фактическими, переданными из главной программы.

Между формальными и фактическими параметрами вспомогательного алгоритма должны выполняться следующие **правила соответствия:**

- *По количеству* (сколько формальных, столько и фактических параметров).
- *По последовательности* (1-му формальному соответствует 1-й фактический \ т.д.).
- *По типам* (типы соответствующих формальных и фактических параметров должны совпадать).

В качестве фактических параметров можно использовать выражения.

Областью действия описания любого программного объекта (переменной, типа, константы и т.д.) является тот блок, в котором расположено это описание. Если данный блок вложен в другой (подпрограмма), то присутствующие в нем описания являются *локальными.* Они действуют только в пределах внутреннего блока. Описания же, стоящие во внешнем блоке, называются глобальными по отношению к внутреннему блоку. Если глобально описанный объект используется во внутреннем блоке, то на него распространяется внешнее (глобальное) описание.

Процедуры:

Структура описания процедуры и функции имеет следующий вид.

```
Procedure   Имя (Список формальных Параметров);
label  ─────┐
const       │                Описание локальных меток,
type        │                констант, типов и переменных
var    ─────┘

procedure ──┐
function  ──┘──────────  Описание внутренних
                         процедур и функций

begin ──────┐
            │            Операторы
end;  ──────┘
```

Задача: Даны натуральные числа a и b. Найти их НОД (наибольший общий делитель) трех величин: $a + b$, $|a - b|$, $a \cdot b$. Запишем это так: НОД ($a + b$, $|a - b|$, $a \cdot b$).

Идея решения состоит в следующем математическом факте: если x, y, z – три натуральных числа, то НОД (x, y, z) = НОД (НОД (x, y), z)..

```
Program NOD;
Var A, B, C: Integer;
```

```
Procedure       Evklid (M, N: Integer; Var K: Integer);
Begin
        While M<>N Do
        If M>N
        Then M: = M – N
        Else N: = N – M
        K: = M
End;
Begin
        Write ( ' a = ' );
        Readln (A);
        Write ( ' b = ' );
        Readln (B);
        Evklid (A + B, Abs (A – B), C);
        Evklid (C, A * B, c);
        Writeln (' НОД = ', C)
End.
```

Функция. Структура функции.

Структура описания функции имеет следующий вид:

function Имя (Список формальных Параметров): Тип результата;

label
const
type
var

Описание локальных меток, констант, типов и переменных

procedure
function

Описание внутренних процедур и функций

begin

end;

Операторы, среди которых должен быть хотя бы один, который присваивает имени функции значение результата.

Отличия в описании процедур и функций касаются только заголовка и раздела операторов

Программа решения рассмотренной выше задачи нахождения НОД с использованием функции будет выглядеть следующим образом:

```
Program NOD3;
Var A, B, Rez: Integer;
Function Evklid (M, N: Integer): Integer;
  Begin
        While M<>N Do
        If M>N
        Then M: = M – N
        Else N: = N – M;
        Evklid: = M
  End;
        Begin
        Write ( ' a = ' );
        Readln (A);
        Write ( ' b = ' );
        Readln (B);
        Rez: = Evklid (Evklid (A + B, Abs (A – B)), A * B);
        Writeln ( ' NOD равен ', Rez)
End.
```

Практическая часть

Задание: Решить задачу вашего варианта с использование подпрограммы (процедуры или функции), оформить отчет, который должен содержать код вашей программы и блок-схему к ней.

1) Даны координаты вершин многоугольника (x1,y1,x2,y2,x3,y3,…x10,y10). Определить его периметр (вычисление расстояния между вершинами оформить подпрограммой).
2) Составить программу для вычисления суммы факториалов всех нечетных чисел от1 до 9.
3) Составить программу для нахождения наименьшего общего кратного двух натуральных чисел

$$НОК(A,B) = \frac{A * B}{НОД(A,B)}$$

4) Составить программу для нахождения наибольшего общего делителя четырех натуральных чисел.
5) Задан массив D. Определить следующие суммы: D[1]+D[2]+D[3]; D[3]+D[4]+D[5]; D[5]+D[6]+D[7].
6) На плоскости заданы своими координатами n точек. Составить программу, определяющую, между какими из пар точек самое большое расстояние. Координаты точек занести в массив.
7) Составить программу для вычисления суммы факториалов всех четных чисел от m до n.
8) Заменить отрицательные элементы линейного массива их модулями, не пользуясь стандартной функцией вычисления модуля. Подсчитать количество произведенных замен.
9) Дан массив A(N) (N-четное). Сформировать массив B(N), элементами которого являются большие из двух рядом стоящих в массиве A чисел. (Например, A=(1,3,5,-2,0,4,0). Элементами массива B будут 3,5,5,0,4,4)
10) Дано натуральное число N. Составить программу для формирования массива, элементами которого являются цифры числа N.
11) Составить программу, определяющую, в каком из данных двух чисел больше цифр.
12) Дан массив *A(N) (N*— четное). Сформировать массив *B(M),* элементами которого являются средние арифметические соседних пар рядом стоящих в массиве *A* чисел. (Например, массив *A* состоит из элементов 1; 3; 5; -2; 0; 4; 0; 3. Элементами массива *B* будут 2; 1,5; 2; 1,5).
13) Даны числа *a, b, c, d* (длины сторон прямоугольника) и число e (диагональ прямоугольника). Вычислить его площадь, разделив данный прямоугольник на 2 треугольника и используя формулу Герона для нахождения их площади.
14) Даны отрезки a, b, c, d. Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь данного треугольника. Определить функцию Plo(x, y, z), печатающую площадь треугольника со сторонами x, y, z, если такой треугольник существует.
15). Даны действительные числа s, *t.* Получить g(1.2, s)+g(t, s)-g(2s-l, st), где

$$g(a,b) = (a^2+b^2) / (a^2+2ab+3b^2+4)$$

Практическая работа 7. Применение методов структурирования программ. Использование одномерных и двумерных массивов

Цель: Научиться использовать структурированные типы данных строки и множества при решении задач на языке Turbo Pascal

Теоретические сведения:

СТРОКИ

Строковая переменная описывается в разделе описания переменных следующим образом:

Var <идентификатор>: String[<максимальная длина строки>]

Например:

Var Name: String [20]

Параметр длины может и не указывается в описании. В таком случае подразумевается, что он равен максимальной величине 255. Например:

Var slovo: String

Для обработки строковых величин в Турбо Паскале существуют специальные процедуры и функции:

Length(st) – значением функции является длинна строковой переменной **st**.

Copy(st,m,n) – значением функции является подстрока из **n** символов, вырезанных из строки **st**, начиная с позиции указанной параметром **m**.

Delete(st,m,n) – данная процедура удаляет **n** символов из строки **st**, начиная с позиции указанной параметром **m**.

Concat(st1,st2,…stn) – соединение строк. Можно использовать конструкцию **st1+ st2+…+ stn**.

Insert(st1, st2,m) – втавка в строку **st2** строки **st1**, начиная с позиции **m**. Общая длина строки не превышает длину строки **st2**.

Pos(st1, st2) - значением функции будет номер позиции в которой в строке **st2** первый раз встречается строка **st1**.

Str(x,st) – заданное числовое значение преобразуется в строку символов. Значение присваивается переменной **st**.

Val(st,x,c) – строка символов **st**, состоящая из цифр, преобразуется в число. Значение передаётся переменной **x**. Параметр определяется средствами Турбо Паскаля.

Функция Copy (S, Poz, N) выделяет из строки S подстроку длиной в N символов, начиная с позиции Poz . N и Poz – целочисленные выражения.

Пример:

Значение S	Выражение	Результат
' ABCDEFG '	Copu (S, 2, 3)	' BCD '
' ABCDEFG '	Copu (S, 4, 4)	' DEFG '

Функция Concat (S1, S2, … , SN) выполняет сцепление (конкатенацию) строк S1, … , SN в одну строку.

Пример:

Выражение	Результат
Concat (' AA ', ' XX ', ' Y ')	' AAXXY '

Функция Length (S) определяет текущую длину строки S. Результат – значение целого типа.

Пример:

Значение S	Выражение	Результат
' test – 5 '	Length (S)	6
' (A + B) * C '	Length (S)	7

Функция Pos (S1, S2) обнаруживает первое появление в строке S2 подстроки S1. Результат – целое число, равное номеру позиции, где находится первый символ подстроки S1.

Если в строке S2 подстроки S1 не обнаружено, то результат равен 0.

Пример:

Значение S2	Выражение	Результат
' abcdef '	Pos (' cd ', S2)	3
' abcdcdef '	Pos (' cd ', S2)	3
' abcdef '	Pos (' k ', S2)	0

Процедура Delete (S, Poz, N) выполняет удаление N символов из строки S, начиная с позиции Poz.

Пример:

Исходное значение S	Оператор	Конечное значение S
' abcdefg '	Delete (S, 3, 2)	' abefg '
' abcdefg '	Delete (S, 2, 6)	' a '

В результате выполнения процедуры уменьшается текущая длина строки в переменной S.

Процедура Insert (S1, S2, Poz) выполняет вставку строки S1 в строку S2, начиная с позиции Poz.

Пример:

Начальное S2	Оператор	Конечное S2
' ЭВМ PC '	Insert (' IBM- ', S2, 5)	' ЭВМ IBM-PC '
' Рис. 2 '	Insert (' N ', S2, 6)	' Рис. N2 '

Пример:

Составить алгоритм, подсчитывающий количество тех слов в строке из N букв, в которых третьей является заданная буква b. Слова разделены пробелами. Других знаков препинания нет.

Алгоритм может быть следующим: строка просматривается и всюду, где нужная буква стоит третьей после пробела или начала фразы, а перед ней находится не пробел, счётчик увеличивается на 1. Перед этим проверяется, есть ли в фразе хотя бы две буквы.

Задача может иметь и другое решение.

```
Program  bukvy (input,output):
Uses crt;
Var strk:string;
    bukva:string[1];
    I, k:integer;
Begin
       Clrscr; {ввод строки};
       Writeln(‘введите строку символов’);
       Readln (bukva);
       clrscr;
       k:=0; {проверка строки на наличие хотя бы трёх первых символов}
       if length(strk)=3
          then
              if (copy(strk, I, 1)<>'') and (copy(strk,3,3) =bukva)
                then k:=k+1;
       for I:=1 to length(strk)-3 do {поиск буквы в остальных словах строки}
          if (copy(strk, I, 1)=’’) and (copy(strk, I+1)<>’’) and (copy(strk, I+2, 1)<>’’) and (copy(strk,
I+3,1)=bukva)
            then k:=k+1;
       {печать исходной строки и количества слов}
       writeln(‘В заданной строке:’);
```

```
    writeln (strk);
    writeln ('слов с искомой буквой', bukva:3,k);
    repeat until keypressed;
end.
```

МНОЖЕСТВА

Под множеством понимают ограниченный, неупорядоченный набор различных элементов одного типа. В отличие от массивов к элементам множества нет прямого доступа (по индексам этих элементов, как в массивах). Поэтому ввод-вывод множеств производится с использованием операций объединения (при вводе) и проверки принадлежности (при выводе). Под мощностью множества понимают количество элементов, содержащихся в данном множестве.

Перед выполнением работы необходимо ознакомится с правилами описания и использования переменных типа множество, типизированных констант типа множество, переменных, заданных перечислением, изучить допустимые операции над переменными этих типов.

Представим формат записи множественных типов:

type

<имя типа> = set of <элемент1, …, элементn> ; var

<идентификатор, . . . > : <имя типа> ; Можно задать множественный тип и без предварительного описания:

var

<идентификатор,...> : set o£ <элемент1,...>;

Примеры описания множеств:

```
L=set of 'A'..'Z';
C=(Red, Green);
H=set of 1..31;
N=set of '0'..'9';
M=set of ['.',',','!','?','-','+','='];
D=set of 0..9;
U=set of char;
```

Процедуры работы с множествами:

Include(S,i) – включает новый элемент i в множество S;

Exclude(S,i) – исключает элемент i из множества S.

Пример программы

```
program r;
var m1: set of 1..50;
     m2: set of 1..50;
     m3: set of 1..50;
   i:integer;
begin
  m1:=[1,4]; m2:=[4,3];
  m3:=m1+m2;
  for i:=1 to 4 do                {вывод элементов множества}
   if i in m3 then write (i:3);
end.
```

ОПЕРАЦИИ НАД МНОЖЕСТВАМИ

При работе с множествами допускается использование операций отношения "=", "<>", ">=", "<=", объединения, пересечения, разности множеств и операции in. Результатом выражений с применением этих операций является значение True или False.

Операция "равно" (=). Два множества A и B считаются равными, если они состоят из одних и тех же элементов. Порядок следования элементов в сравниваемых множествах значения не имеет.

Например:

Значение A	Значение B	Выражение	Результат
[1,2,3,4] [‘a’,’b’,’c’] ['a'..'z']	[1,2,3,4] [‘c’..’a’] ['z'..’a’]	A = B A = B A = B	True False True

Операция "не равно" (<>). Два множества A и B считаются не равными, если они отличаются по мощности или по значению хотя бы одного элемента.

Например:

Значение A	Значение B	Выражение	Результат
[1,2,3] [‘a’..’z’] [‘c’..’t’]	[3,1,2,4] ['b'..'z'] ['t’..'c']	A<>B A<>B A<>B	True True False

Операция "больше или равно" (>=). Операция "больше или равно" (>=) используется для определения принадлежности множеств. Результат операции A >- B равен True, если все элементы множества B содержатся в множестве A. В противном случае результат равен False.

Например:

Значение A	Значение B	Выражение	Результат
[1,2,3,4] [‘a’..’z’] ['z’,’x’,’c’]	[2,3,4] [‘b’..’t'] [‘c’,’x’]	A>=B A>=B A>=B	True True True

Операция "меньше или равно" (<=). Эта операция используется аналогично предыдущей операции, но результат выражения A <= B равен True, если все элементы множества A содержатся в множестве B. В противном случае результат равен False.

Например:

Значение A	Значение B	Выражение	Результа
[1,2,3] ['d'..'h'] ['a',’v’]	[1,2,3,4] ['z'..’a'] ['a’,’n’,’v’]	A<=B A<=B A<=B	True True True

Операция in. Операция in используется для проверки принадлежности какого-либо значения указанному множеству. Обычно применяется в условных операторах.

Например:

Значение A	Выражение	Результат
[2] ‘v’ X1	if A in [1,2,3] then... if A in ['a'..'n'] then ... if A in [X0, X1, X2, X3] then...	True False True

При использовании операции in проверяемое на принадлежность значение и множество в квадратных скобках не обязательно предварительно описывать в разделе описаний. Операция in позволяет эффективно и наглядно производить сложные проверки условий, заменяя иногда десятки других операций. Например, выражение if (a=l) or (a=2) or (a=3) or (a=4) or (a=5) or (a=6) then... можно заменить более коротким выражением if a in [1.6] then....

Часто операцию in пытаются записать с отрицанием: **x NOT in м.**

Такая запись является ошибочной, так как две операции следуют подряд; правильная инструкция имеет вид NOT (x in M) .

Объединение множеств (+). Объединением двух множеств является треть* множество, содержащее элементы обоих множеств. Например:

Значение A	Значение B	Выражение	Результат
[1,2,3] [‘a’..’d’] []	[1,4,5] [‘e’..’z’] []	A+ B A+ B A+B	[1,2,3,4,5] [‘a’..’z’] []

Пересечение множеств (*). Пересечением двух множеств является третье множество, которое содержит элементы, входящие одновременно в оба множества. Например:

Значение A	Значение B	Выражение	Результат
[1,2,3] ['A'..'Z'] []	[1,4,2,5] ['B’..'R'] []	A*B A*B A*B	[1,2] ['B'.’R'] []

Разность множеств (—). Разностью двух множеств является третье множество, которое содержит элементы первого множества, не входящие во второе множество.

Например:

Значение A	Значение B	Выражение	Результат
[1,2,3,4] ['A'..'Z'] [X1,X2,X3,X4]	[3,4,1] ['D'..'Z'] [X4,X1]	A —B A —B A —B	[2] ['A’..'C] [X2,X3]

Практическая часть

Задания: Решить задачу своего варианта, оформить отчет, который должен содержать код вашей программы и блок-схему к ней.

1. Подсчитать сколько раз в заданной строке встречается заданная буква.

Для заполнения карточек «Спортлото» необходимо получить набор из к случайных чисел:

- числа должны находиться в диапазоне 1..36;
- числа не должны повторяться.

2. Дана строка, заканчивающаяся точкой. Подсчитать, сколько слов в строке.

3. Имеются три множества символьного типа, которые заданы своими конструкторами:

Y1=['A','B','D','R','H']
Y2=['R','A','H','D']
Y3=['A','R'].

Сформировать новое множество Y=(Y1+Y2)*Y3. Распечатать элементы множества Y.

4. В заданной фразе после каждой буквы «о» вставить сочетание «ok».

5. Опишите множества R и L, содержащие русские и латинские буквы. В цикле вводите русские и латинские буквы и выводите соответствующее сообщение. Выход из цикла — введенная буква Z.

6. Написать программу, которая перевернёт введённое с клавиатуры слово или фразу.

7. Известны сорта роз, выращиваемых тремя цветоводами: «Анжелика», «Виктория», «Гагарин», «Ave Maria», «Катарина», «Юбилейная». Определить те сорта, которые имеются у каждого из цветоводов, которые есть хотя бы у одного из цветоводов, которых нет ни у одного из цветоводов.
8. Дана строка. Преобразовать ее, удалив каждый символ *.
9. В озере водится несколько видов рыб. Три рыбака поймали рыб, представляющих некоторые из имеющихся видов. Определить:
- какие виды рыб есть у каждого рыбака;
- какие рыбы есть в озере, но нет ни у одного из рыбаков.

10. Заменить в строке символов каждую группу букв child группой букв children.
11. Описать множество гласных и согласных букв русского языка. Определить количество гласных и согласных букв в предложении, введенном с клавиатуры.
12. Дана строка символов. Подсчитать число вхождений символов +, -, * в строку.
13. Опишите множества Ml (1,2,3,4) и М2 (3,4,1). Получите результирующее множество М3= Ml -— М2. Определите, имеется ли в М3 элемент 2.
14. В озере водится несколько видов рыб. Три рыбака поймали рыб, представляющих некоторые из имеющихся видов. Определить:
- какие виды рыб есть у каждого рыбака;
- какие рыбы есть в озере, но нет ни у одного из рыбаков.

15. В строке между словами вставить вместо пробела запятую и пробел.
16. На трех участках возделывают сельскохозяйственные культуры. Известны виды культур, выращиваемых на каждом из участков. Определить виды тех культур, которые возделывают на каждом из участков; возделывают хотя бы на одном из участков; не возделывают ни на одном участке. (Культуры: картофель, укроп, морковь, горох, капуста, редис.)

Практическая работа 8. Применение методов объектно-ориентированного программирования

Цель:
- научиться работать с элементами управления CheckBox, LinkLabel и DateTimePicker
- настроить свойства для каждого объекта пользовательского интерфейса;
- создать код программы;
- сохранить и запустить программу;

Элементы управления для ввода информации

Ход работы

В Visual Basic имеется несколько механизмов для приема в программе входных данных. Текстовые поля предназначены для приема информации, набираемой на клавиатуре, меню представляют собой команды, которые можно выбирать с помощью мыши или клавиатуры, а диалоговые окна содержат совокупность элементов, которые можно выбирать как по отдельности, так и группой. В этом упражнении мы познакомимся с четырьмя элементами управления, предназначенными для приема информации в различных ситуациях. Вы узнаете об элементах управления RadioButton (переключатель), CheckBox (флажок), ListBox (список) и ComboBox (комбинированный список). Мы познакомимся с ними на примере программы на Visual Basic, которая называется Input Controls и представляет собой пользовательский интерфейс для системы заказов. Давайте на примере элемента управления CheckBox посмотрим, как введенная информация обрабатывается в форме и коде программы. Выполните следующие действия.

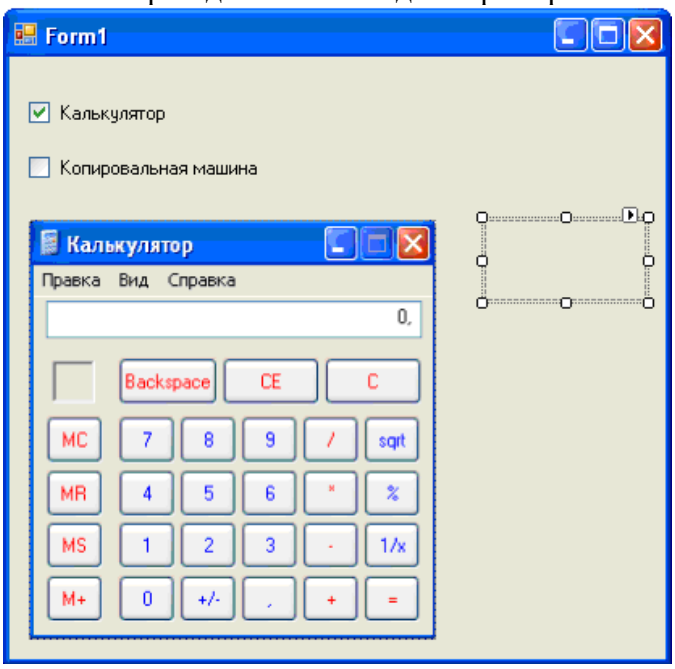
Некоторые элементы управления Visual Basic показывают информацию, а другие собирают информацию от пользователя или обрабатывают данные "за кулисами". В этом упражнении мы будем работать с элементом управления DateTimePicker, который с помощью графического календаря со стрелками прокрутки спрашивает пользователя о дате или времени.

Задание 1: Работа с элементом управления CheckBox

1. Создайте новый проект с именем MyCheckBox
2. В области элементов выберите элемент управления CheckBox.
3. Нарисуйте на форме один над другим два объекта флажка. Флажки CheckBox появляются на форме точно так же, как и другие объекты.
4. Выберите элемент управления PictureBox и нарисуйте под этими двумя флажками два квадратных объекта вывода изображений.
5. Установите для новых объектов следующие свойства.

Объект	Свойство	Значение
CheckBox1	Checked	True
	Text	"Калькулятор"
CheckBox2	Text	"Копировальная машина"
PictureBox1	Image	Укажите путь к рисунку (можно взять здесь)
	SizeMode	StretchImage
PictureBox1	SizeMode	StretchImage

Флажки используются для показа и сокрытия изображений калькулятора и копировальной машины. В свойстве Text объекта CheckBox содержится надпись, связанная с флажком в графическом интерфейсе. Свойство Checked позволяет задать для флажка значение по умолчанию. Если значение Checked равно True, в поле флажка будет стоять галочка, а если False (это значение по умолчанию), то ячейка будет пустой. Чтобы изменить размер изображений так, что они будут растянуты до размеров полей вывода, используйте свойство SizeMode. Форма должна выглядеть примерно так:



6. Чтобы открыть редактор кода на процедуре обработки события CheckBox1_CheckedChanged, дважды щелкните мышью на объекте флажка CheckBox1, а затем введите следующий код программы

```
Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object,
    If CheckBox1.CheckState = 1 Then
        PictureBox1.Image = System.Drawing.Image.FromFile _
("  Путь к рисунку  \calcultr.bmp")
        PictureBox1.Visible = True
    Else
        PictureBox1.Visible = False
    End If
End Sub
```

Процедура обработки события CheckBox1_CheckedChanged вызывается только в том случае, если пользователь щелкает на объекте (ячейке или надписи, то есть на всем поле) первого флажка. В процедуре обработки события используется конструкция If...Then, в которой определяется текущее состояние первого флажка. Если в его поле стоит галочка, то показывается картинка с калькулятором. Если отметка присутствует, то свойство CheckState содержит значение 1, а если она отсутствует, то значение 0. Свойство Visible используется для показа картинки, если отметка присутствует, или ее сокрытия, если отметка отсутствует. Обратите внимание, что длинная строка, которая загружает изображение в объект вывода изображения, перенесена на следующую строку с помощью символа продолжения строки (_).

7. Чтобы снова перейти на форму, в Solution Explorer (Обозревателе решений) нажмите кнопку View Designer (Просмотреть конструктор), а затем дважды щелкните мышью на втором флажке и добавьте в процедуру обработки события CheckBox2_CheckedChanged следующий код, не забыв изменить путь к рисунку с копировальной машиной:

```
Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Obj
    If CheckBox2.CheckState = 1 Then
        PictureBox2.Image = System.Drawing.Image.FromFile _
            (":\copymach.bmp")
        PictureBox2.Visible = True
    Else
        PictureBox2.Visible = False
    End If
End Sub
```

Эта процедура события почти идентична той, которую вы только что ввели; отличаются только имена метафайла Windows (copymach.bmp), объект флажка (CheckBox2) и объект вывода изображения (PictureBox2). Запустите проект.

Задание 2: Использование элемента управления LinkLabel

Во многих Windows-приложениях уже давно предоставляется доступ в интернет, и это уже становится стандартной функцией. В Visual Studio . NET добавить такую функцию к создаваемой программе стало намного проще. Используя шаблоны библиотеки Web Forms и другие возможности, можно создавать сложные приложения для работы по сети Интернет. А чтобы открыть веб-страницу в веб-обозревателе, потребуется всего лишь нескольких строк кода.

В этом упражнении вы научитесь применять элемент управления LinkLabel, который в поле формы показывает текст, являющейся ссылкой на адрес в Интернете.

Элемент управления LinkLabel - новое пополнение в Visual Basic. Используя его вместе с методом Process.Start, можно открывать ссылки на форме с помощью обозревателей Internet Explorer, Netscape Navigator или других. В нашем примере с помощью элемента управления LinkLabel мы подключимся к веб-странице поисковика.

Создание программы WebLink

1. Создайте новый проект Windows Application (Приложение Windows) с именем MyWebLink.
2. В области элементов выберите элемент управления LinkLabel и нарисуйте в форме прямоугольный объект для текста ссылки. Объект текст ссылок выглядит как обычные объекты надпись (неизменяемый текст), но отображаются в форме синим цветом с подчеркиванием.
3. Задайте свойство Text объекта LinkLabel равным значению http://www.ya.ru/). Форма будет выглядеть примерно так.

4. В среде разработки щелкните на поле формы, чтобы ее выделить (на самой форме, а не на объекте ссылки). Это стандартный способ показать настройки по умолчанию объекта Form1 в окне Properties (Свойства). Как и у других объектов в проекте, у формы также есть свойства, которые можно настроить.
5. В свойство Text формы Form1 впишите "Пример с Интернет-ссылкой". Свойство Text управляет тем, что появляется в строке заголовка формы при запуске программы. Эта настройка относится не только к веб-приложениям, в дальнейшем мы будем настраивать строку заголовка в большинстве создаваемых программ.
6. Дважды щелкните мышью на объекте ссылки и в процедуре обработки события LinkLabel1_LinkClicked введите следующий код:

```
Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByV
    ' Изменяем цвет ссылки, установив LinkVisited в значение True.
    LinkLabel1.LinkVisited = True
    ' Используем метод Process.Start, который запускает обозреватель,
    ' указанный в системе по умолчанию
    ' с URL Microsoft Press:
    System.Diagnostics.Process.Start("http://www.ya.ru/")
End Sub
```

В этом фрагменте добавлены комментарии, чтобы дать вам возможность немного попрактиковаться в их вводе. Как только будет введен символ одиночной кавычки ('), Visual Studio изменит цвет строки на зеленый, подсказывая, что эта строка является комментарием. Комментарии предназначены только для документирования - компилятором они не обрабатываются.

Два других оператора программы, которые не являются комментариями, собственно управляют работой ссылки. Присвоение свойству LinkVisited значения True делает ссылку бледно-фиолетовой, что во многих обозревателях указывает на то, что ассоциированный со ссылкой HTML-документ уже был вызван раньше. Хотя для показа веб-страницы настройка этого свойства необязательна, считается хорошим тоном предлагать пользователю информацию в согласовании с другими приложениями.

Второй оператор программы запускает веб-обозреватель по умолчанию (например, Internet Explorer), если он еще не запущен. (Если обозреватель уже запущен, то в нем немедленно загружается этот URL-адрес.) Метод Start класса Process выполняет важную работу, запуская в памяти процесс или сеанс для исполнения программы обозревателя. Класс Process управляет многими аспектами исполнения программ и входит в библиотеку объектов System.Diagnostics, которую программисты на Visual Basic .NET называют "пространство имен System.Diagnostics". Передав в метод Start интернет-адрес (или URL-адрес), мы сообщили программе на Visual Basic, что хотим посмотреть веб-сайт, и Visual Basic оказался достаточно умным, чтобы определить, что наилучшим инструментом для просмотра этого URL-адреса является обозреватель по умолчанию, хотя мы и не указывали его имя.

Метод Process.Start замечателен тем, что с его помощью можно запустить и другие Windows-приложения. Если для просмотра URL-адреса нужно определить конкретный обозреватель по его имени, это можно сделать так (в данном случае вызывается Internet Explorer).

```
System.Diagnostics.Process.Start("IExplore.exe", "http://www.ya.ru/")
```

Если бы требовалось запустить с помощью метода Start другое приложение, то это можно было бы сделать так:

```
System.Diagnostics.Process.Start("C:\Program Files\mp3DirectCut\mp3DirectCut.exe"
```

То есть вы должны точно прописать путь к файлу, запускающему необходимое вам приложение.

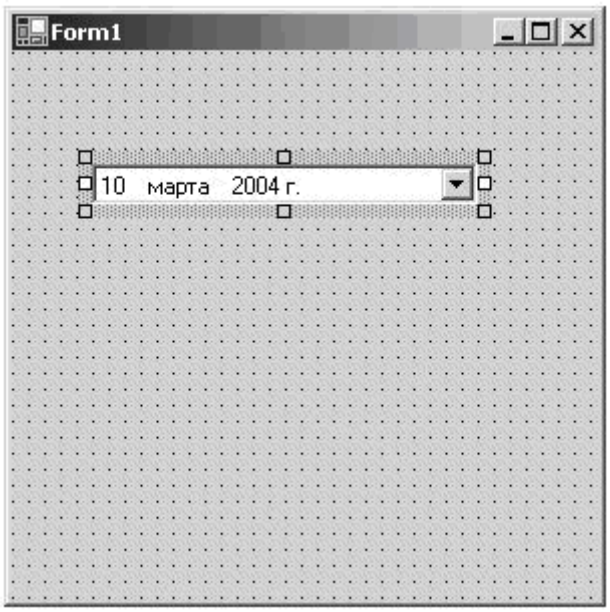
7. Добавьте на форму кнопку, назовите ее именем запускаемого приложения, найдите на вашем компьютере какой-нибудь exe. Файл и пропишите код для его запуска из формы

Задание 3: Создание программы Birthday

1. В меню File (Файл) выберите New (Создать), а затем Project (Проект). Появится диалоговое окно New Project (Создать проект).
2. Создайте новый проект с именем MyBirthday. Новый проект будет создан, и в Windows Forms Designer (Конструкторе Windows Forms) появится пустая форма.
3. В области элементов выберите элемент управления DateTimePicker.

Если вы не видите DateTimePicker в области элементов, скорее всего он находится вне пределов видимости. Чтобы прокрутить список области элементов, нажмите на нижней стрелке прокрутки, расположенной рядом с закладкой Clipboard Ring (Буфер обмена). Чтобы прокрутить список в другую сторону, нажмите стрелку прокрутки рядом с закладкой Windows Forms.

4. В центре формы нарисуйте объект выбора даты и времени, как показано на следующей иллюстрации.



Объект выбора даты и времени по умолчанию показывает текущую дату, но эту настройку можно изменить в свойстве Value. Показ даты при проектировании очень удобен - это позволяет при создании объекта правильно задать его размеры.

5. В области элементов выберите элемент управления Button, и ниже объекта выбора даты и времени добавьте объект кнопки. Эта кнопка будет использована для показа дня рождения и для проверки правильности работы объекта выбора даты и времени.

6. В окне Properties (Свойства) измените свойство Text объекта кнопки на Показать день моего рождения. Теперь нужно добавить в процедуру, связанную с объектом кнопки, несколько строк программного кода. Эта процедура называется процедурой обработки события, так как она запускается тогда, когда в объекте происходит событие, например, щелчок мышью.

7. Дважды щелкните мышью на объекте кнопки, а потом наберите следующий фрагмент программы между операторами Private Sub и End Sub в процедуре события Button1_Click:

```
MsgBox("Ваш день рождения " & DateTimePicker1.Text)
MsgBox("День года: " & _
   DateTimePicker1.Value.DayOfYear.ToString())
MsgBox("Сейчас " & DateTimePicker1.Value.TimeOfDay.ToString())
```

Этот фрагмент программы показывает три последовательных окна сообщения (небольшие диалоговые окна), которые содержат информацию из объекта календаря. В первой строке используется свойство Text календаря для вывода информации о дате рождения, которую пользователь выберет в этом объекте после запуска программы. Функция MsgBox кроме текстового значения из свойства Text календаря показывает строку "Ваш день рождения". Эти два текстовых элемента объединяются в строку с помощью оператора конкатенации (слияния) строк &.

Вторая и третья строки вместе представляют собой один оператор, разделенный на две строки при помощи символа продолжения (_). Это сделано из-за того, что это выражение слишком длинное для печати в методическом указании в одну строку.

В строке `Date.timePicker1.Value.DayOfYear.ToString())` объект календаря используется для вычисления дня года, отсчитывая с 1 января. Это делается с помощью свойства DayOfYear и метода ToString, который переводит числовой результат вычисления даты в текстовое значение, которое гораздо проще показать с помощью функции MsgBox.

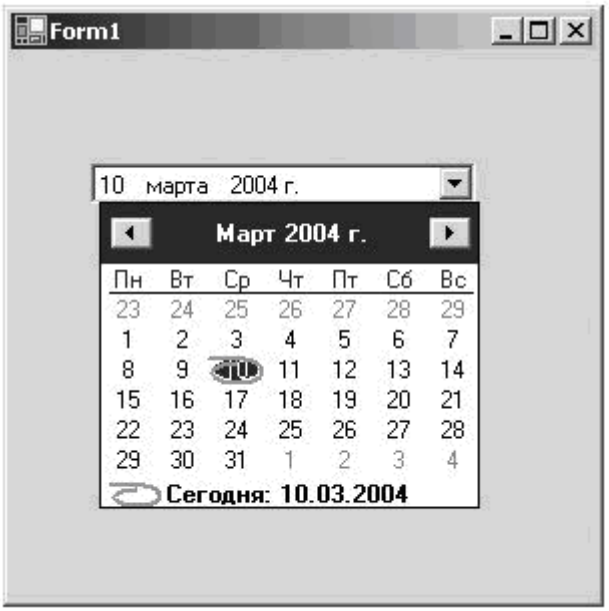
Методы - это специальные выражения, которые выполняют действие или работу для конкретного объекта, например, преобразуют числа в строку или добавляют еще один элемент к списку. Методы отличаются от свойств, которые содержат значение, и процедур событий, которые выполняются при манипуляциях пользователя с объектом. Методы могут использоваться совместно в нескольких объектах, так что когда вы узнаете, как использовать один метод, вы сможете использовать его в различных условиях.

В четвертой строке фрагмента, после перевода значения в строковое (или текстовое) представление, в окне сообщения показывается информация о точном времени.

Совет. Длина строки программы в редакторе кода и Visual Studio может быть более 65 000 символов, но обычно проще всего работать со строками, длина которых не превышает 80 символов. Длинные операторы можно разбить на несколько строк, добавив в нужном месте пробел и символ продолжения строки (_). Эта комбинация должна стоять в конце всех строк выражения, за исключением последней. Для разделения строки, заключенной в кавычки (строковой константы), символ продолжения использовать нельзя. В этом упражнении символ продолжения строки использован для того, чтобы разделить на две части вторую строку кода.

Запуск программы Birthday

1. На стандартной панели инструментов нажмите кнопку Start (Начать). Программа Birthday запустится в среде разработки. В окне объекта выбора даты и времени появится текущая дата.
2. Нажмите стрелку раскрывающегося списка, чтобы вывести на экран представление этого объекта в виде календаря. Форма будет выглядеть как на следующей иллюстрации (дата, разумеется, будет другая).



3. Чтобы увидеть предыдущие месяцы календаря, нажмите на левой стрелке прокрутки. Обратите внимание, что текстовое поле объекта при прокрутке дат также изменяется. Однако значение Сегодня в нижней части календаря

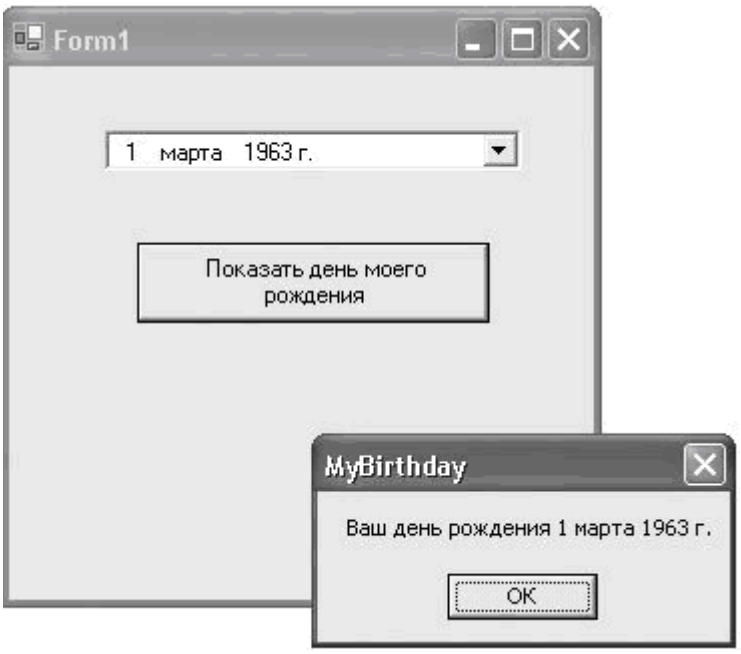
остается неизменным. Хотя календарь можно промотать назад до даты вашего рождения, вряд ли у кого хватит терпения прокручивать месяц за месяцем. Чтобы быстрее перейти к году вашего рождения, выделите в текстовом поле значение года и введите новое значение.

4. В текстовом поле объекта выбора даты и времени выделите четыре цифры года. Как только вы это сделаете, календарь закроется.

5. Вместо выделенного года введите год вашего рождения, а затем снова щелкните на стрелке раскрывающегося списка. Появится календарь, который будет открыт на годе вашего рождения.

6. Чтобы найти месяц, в котором вы родились, снова нажмите левую или правую стрелки прокрутки, а затем щелкните на день вашего рождения. Когда вы выберете дату, календарь закроется, и в текстовом поле будет показан день вашего рождения. Чтобы увидеть, как эта информация становится доступной для других объектов вашей формы, нажмите кнопку.

7. Нажмите кнопку Показать день моего рождения. Visual Basic исполнит введенный вами код программы и покажет окно с сообщением, содержащим день и дату вашего рождения. Обратите внимание на соответствие двух дат.



8. В окне сообщения нажмите OK. Появится второе окно сообщения, указывающее, в какой день года вы родились.

9. Нажмите OK, чтобы показать последнее окно сообщения. Появятся текущие дата и время - программа работает! Вы обнаружите, что объект выбора даты и времени очень удобен - он не только помнит новую, введенную вами информацию о дате или времени, но также отслеживает текущие дату и время и может показывать эту информацию в различных форматах. **Совет.** Чтобы настроить объект выбора даты и времени для показа времени, а не даты, установите свойство Format этого объекта равным Time.

10. Чтобы закрыть окно сообщения, щелкните на OK, а затем на кнопке формы Закрыть. На этом мы пока закончим работу с элементом управления DateTimePicker.

Контрольные вопросы

1. Опишите механизмы Visual Basic для приема в программе входных данных
2. Назначение свойства Checked объекта CheckBox, его значение
3. Назначение свойства SizeMode, что означает значение StretchImage
4. Назначение объекта LinkLabel
5. Опишите работу метода Start
6. Назначение элемента управления DateTimePicker
7. Назначение оператора конкатенации &, пример его использования
8. Объясните работу фрагмента кода

```
Date.timePicker1.Value.DayOfYear.ToString())
```

.

Практическая работа 9. Создание простых продуктов средствами VB.Net

Цель: научиться создавать редактор текста;
изучить возможность интеграции с MS Excel.

Задание 1. Создание простейшего текстового редактора.

В ходе выполнения задания необходимо создать редактор текста, который может открывать, сохранять файлы в RTF- и TXT-форматах, изменять стиль шрифта, устанавливая его полужирным или курсивным, а также специфицировать размер шрифта. Кроме того, если в открываемом документе имеется гиперссылка, то щелчок на ней вызовет загрузку браузера соответствующей Web- страницы.

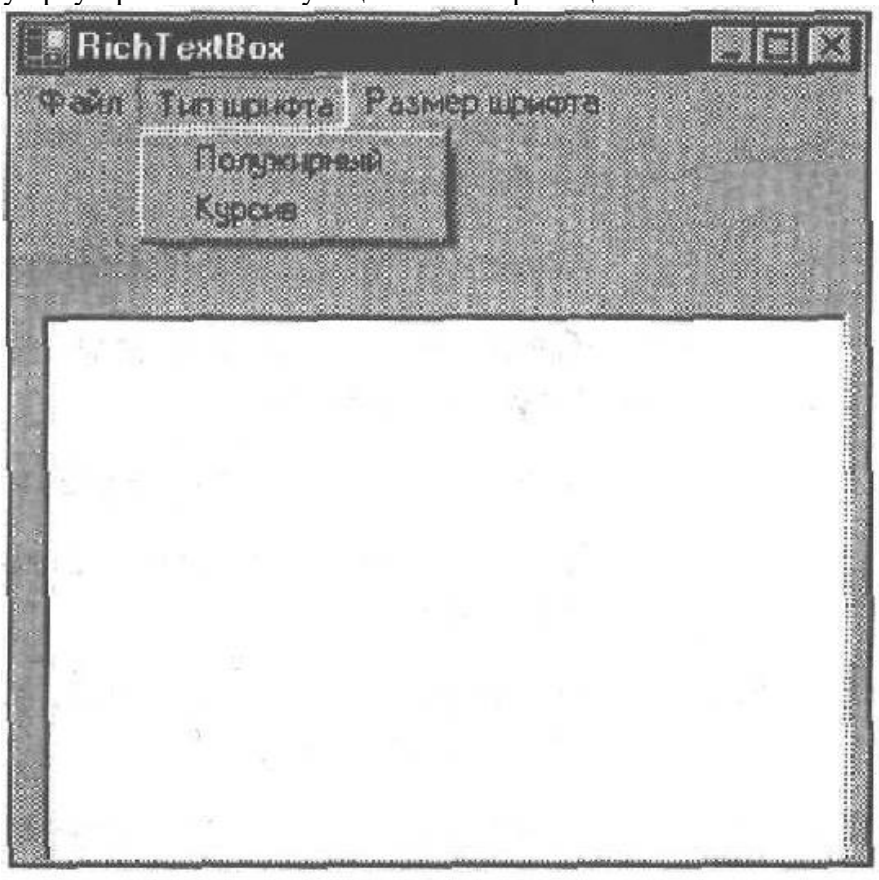


Рис. 11.48. Простейший редактор текста

Для инициализации данного проекта создайте форму, на которой расположите расширенное поле ввода RichTextBox, которое является экземпляром класса RichTextBox и меню MainMenu (работа с данным элементом осуществляется по аналогии с Delphi). Элемент управления RichTextBox предназначен для отображения, ввода и редактирования текста и обладает всеми возможностями поля ввода.

Используя мастер меню, скопируйте главное меню, состоящее из трех выпадающих подменю: Файл, Типы шрифта и Размер шрифта. В меню Файл создайте 2 команды: Открыть и Сохранить, в меню Тип шрифта – Полужирный и Курсивный, в меню Размер шрифта – 10 и 12. Используя окно Properties, установите этим командам значения свойства Name равным itmOpen, itmSave, itmBold, itmItalic, itm10. itm12 соответственно. Кроме того, установите значение свойств Name расширенного поля ввода равным rtbText. В код, сгенерированным мастером проекта, добавьте следующие инструкции:

```
Public Class Form1
    Private oldFont As Font
    Private newfont As Font

    Private Sub itmOpen_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles itmOpen.Click
        Dim ofd As New OpenFileDialog()
        ofd.DefaultExt = "*.rtf"
        ofd.Filter = "RTF Files|*.rtf|TXT Files|*txt"
        ofd.FilterIndex = 1
        If ofd.ShowDialog() = Windows.Forms.DialogResult.OK Then
            Try
                rtbText.LoadFile(ofd.FileName, selectedformat(ofd.FilterIndex))
            Catch ex As Exception
                MessageBox.Show("Ошибка при загрузке файла")
            End Try
        End If
    End Sub

    Private Sub itmSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles itmSave.Click
        Dim sfd As New System.Windows.Forms.SaveFileDialog()
        sfd.Filter = "RTF Files|*.rtf|TXT Files|*.txt"
        sfd.FilterIndex = 1
        sfd.RestoreDirectory = True
        If sfd.ShowDialog() = Windows.Forms.DialogResult.OK Then
            Try
                rtbText.SaveFile(sfd.FileName, SelectedFormat(sfd.FilterIndex))
            Catch ex As Exception
                MessageBox.Show("Ошибка при записи файла")
            End Try
        End If
    End Sub
```

```
Private Function SelectedFormat(ByVal idx As Integer) As Integer
    Select Case idx
        Case 1
            Return RichTextBoxStreamType.RichText
        Case 2
            Return RichTextBoxStreamType.PlainText
    End Select
End Function

Private Sub itmBold_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles itmBold.Click
    oldFont = rtbText.SelectionFont
    If oldFont.Bold Then
        newfont = New Font(oldFont, oldFont.Style And Not FontStyle.Bold)
    Else
        newfont = New Font(oldFont, oldFont.Style Or FontStyle.Bold)
    End If
    rtbText.SelectionFont = newfont
    rtbText.Focus()
End Sub

Private Sub itmItalic_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles itmItalic.
    oldFont = rtbText.SelectionFont
    If oldFont.Italic Then
        newfont = New Font(oldFont, oldFont.Style And Not FontStyle.Italic)
    Else
        newfont = New Font(oldFont, oldFont.Style Or FontStyle.Italic)
    End If
    rtbText.SelectionFont = newfont
    rtbText.Focus()
End Sub

Private Sub itm10_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles itm10.Click
    setfontsize(10)
End Sub

Private Sub itm12_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles itm12.Click
    setfontsize(12)
End Sub

  Private Sub setfontsize(ByVal fs As Integer)
      Dim ff As FontFamily
      ff = rtbText.SelectionFont.FontFamily
      oldFont = rtbText.SelectionFont
      newfont = New Font(ff, fs, oldFont.Style)
      rtbText.SelectionFont = newfont
      rtbText.Focus()
  End Sub


  Private Sub rtbText_LinkClicked(ByVal sender As Object, ByVal e As System.Windows.Forms.LinkClickedEventArgs)
      Handles rtbText.LinkClicked
      System.Diagnostics.Process.Start(e.LinkText)
  End Sub
End Class
```

Запустите проект и проверьте работоспособность всех функций редактора.

Задание 2. Интеграция с MS Excel
Нахождение значений арифметических выражений

В качестве примера использования MS Excel для создания на VisualBasic .NET Windows-приложения рассмотрим приложение, которое по введенному в поле арифметическому выражению вычисляет его значение.

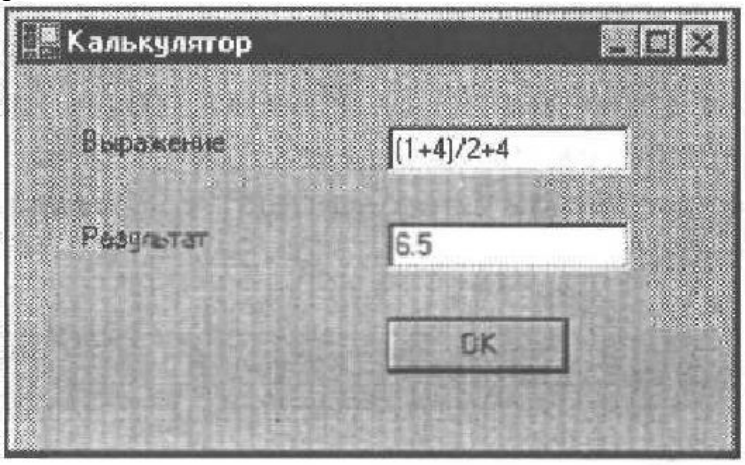


Рис. 13.2. Окно **Калькулятор**

Для создания такого приложения создайте форму с двумя полями ввода (TextBox), двумя надписями (Label) и кнопкой (Button). Использую окно Properties, установите значения свойств элементов управления, перечисленные ниже:

Элемент управления	Свойство	Значение
Надпись	Text	Выражение
Поле ввода	Name	txtExpr
	Text	""
Надпись	Text	Результат
Поле ввода	Name	txtRes
	Text	""
Кнопка	Name	btnOK
	Text	OK

Выберите команду Project/Add Reference. На экране отобразится окно Add Reference. В списке COM выберите Microsoft Excel 9.0 Object Library, нажмите на кнопку OK. Далее к коду, сгенерированному мастером проекта, добавьте следующую инструкцию:

```
Private Sub btnOK_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnOK.Click
  Dim objExcel As Excel.Application = Nothing
  Dim expr As String
  Dim res As Double
  Try
    expr = txtExpr.Text.Trim
    expr = "=" & expr
    objExcel = New Excel.Application()
    If objExcel Is Nothing Then
      MessageBox.Show("Excel не открылся")
      Exit Sub
    End If
    res = objExcel.Evaluate(expr)
    txtRes.Text = res.ToString
    objExcel.Quit()
  Catch ex As Exception
    MessageBox.Show("Ошибка в выражении")
  End Try
End Sub
```

Запустите проект и проверьте его работоспособность.

Практичекая работа 10: Разработка руководства программиста.

Введение

Руководство программиста должно состоять из следующих частей:

-Титульной;

-Информационной;

-Основной.

Титульная часть оформляется согласно ГОСТ 19.104-78 ЕСПД. Основные надписи.

Информационная часть должна состоять из аннотации и содержания. В аннотации приводят сведения о назначении документа и краткое изложение основной части.

Содержание включает перечень записей о структурных элементах основной части документа.

Основная часть руководства программиста должна содержать следующие разделы: (ГОСТ 19.504-79 ЕСПД. Руководство программиста. Требования к содержанию и оформлению)

-Назначение и условия применения программы содержит функции, выполняемые программой и условия, необходимые для выполнения программы: объём оперативной памяти, требования к составу и параметрам периферийных устройств, требования к программному обеспечению.

-Характеристики программы описывают временные характеристики, режимы работы, средства контроля правильности выполнения и самовосстанавливаемости программы.

-Обращение к программе представляет собой

описание

процедур

вызова

программы
,
способов
передачи
параметров
.
-
Входные
и
выходные
данные
должен
содержать
описание
организации
используемой
входной
и
выходной
информации
.
-
Сообщения
содержит
тексты
сообщений
,
выдаваемых
программисту
или
оператору
,
в
ходе
выполнения
программы
,
описание
их
содержания
и
действий
,
которые
необходимо
предпринять
по
этим
сообщениям
.

Задание

Составить руководство программиста в соответствии с ГОСТ19.504-79 ЕСПД. Руководство программиста. Требования к содержанию и оформлению.

Требования к отчёту

Отчёт должен содержать титульный лист, аннотацию, содержание и основную часть, оформленную в соответствии с ГОСТ19.504-79 ЕСПД. Руководство программиста. Требования к содержанию и оформлению.

Раздел 3. «Отладка, тестирование и сопровождение программных продуктов»

Практическая работа 11. Тестирование программ методом «белого ящика»

Цель работы: отработать навыки составления и тестирования программ как «белого ящика»; освоить на практике метода базового пути.

Ход работы:

Теоретические сведения:

Тестирование – это процесс многократного выполнения программы с целью обнаружения максимального количества ошибок. Программа тестируется для того, чтобы повысить уровень надежности программы. Тестирование является составной частью отладки и включает в себя два вида деятельности: разработка тестов и непосредственное тестирование по ним.

Особенность тестировании программ как «белый ящик» заключается в следующем:

Известна: внутренняя структура программы.

Исследуются: внутренние элементы программы и связи между ними (рис. 6.3).

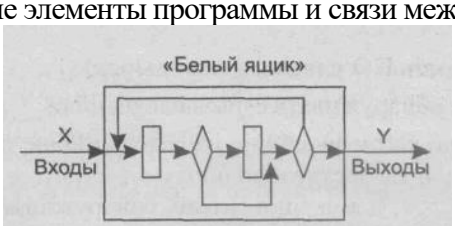


Рис. 6.3. Тестирование «белого ящика»

Объектом тестирования здесь является не внешнее, а внутреннее поведение программы. Проверяется корректность построения всех элементов программы и правильность их взаимодействия друг с другом.

Методы белого ящика

1. Метод покрытия операторов

Критерием этого метода является выполнение каждого оператора хотя бы один раз. Этот критерий является достаточно слабым, т.к. выполнение каждого оператора хотя бы один раз условие необходимое, но недостаточное для результирующего условия.

2. Метод покрытия решений

Согласно этому методу должно быть написано достаточное кол-во тестов, такое, что каждое решение на этих тестах по крайне мере один раз и при этом каждый оператор д. выполняться хотя бы один раз.

3. Метод покрытия условий

В том случае записывают число тестов достаточное для того, что бы все возможные

результаты каждого условия в решении выполнялись по крайне мере один раз.

4. Метод покрытия решений \ условий

Этот метод требует достаточного набора тестов, чтобы возможные результаты каждого условия в решении выполнялись по крайне мере один раз и каждой точке входа передавалось управлениеие по крайне мере один раз.

Практическая работа 12. Тестирование программ методом «черного ящика»

Цель работы: отработать навыки составления и тестирования программ как «белого ящика»; освоить на практике метод способ анализа граничных значений.

Ход работы:

Теоретические сведения:

Тестирование – это процесс многократного выполнения программы с целью обнаружения максимального количества ошибок. Программа тестируется для того, чтобы повысить уровень надежности программы.

Особенность тестировании программ как «черный ящик» заключается в следующем:

Известны: функции программы.

Исследуется: работа каждой функции на всей области определения.

Как показано на рис., основное место приложения тестов «черного ящика» — интерфейс ПО.

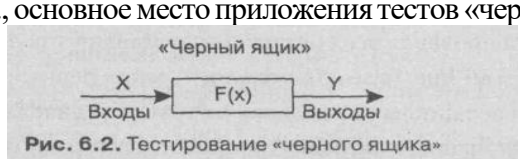


Рис. 6.2. Тестирование «черного ящика»

Эти тесты демонстрируют:

- как выполняются функции программ;
- как принимаются исходные данные;
- как вырабатываются результаты;
- как сохраняется целостность внешней информации.

При тестировании «черного ящика» рассматриваются системные характеристики программ, игнорируется их внутренняя логическая структура.

Методы черного ящика

1. Метод эквивалентных разбиений

Его основу составляют 2 положения:

- Каждый тип должен включать столько различных входных и выходных условий, сколько необходимо для того, что бы минимизировать общее число необходимых тестов.
- Необходимо разбивать входную область программы на конечное число классов эквивалентности

Класс эквивалентности выделяется путем выбора каждого входного условия и разбиением его на две или более групп.

2. Анализ граничных решений (АГР)

Граничные условия –ситуация, возникающая непосредственно на границе, выше или ниже границ входных или выходных элементов класса эквивалентности (КЭ)

АГР предполагает:

- Выбор любого элемента в КЭ в качестве представительного при АГЗ осущ-ся т.о., чтобы проверить тестом каждую границу этого класса.

- При разработке тестов рассматриваются не только входные условия, но и выходные.

3. Метод функциональных диаграмм

Недостатком метод граничных решений и метод эквивалентных разбиений является то, что они не исследуют комбинации входных условий.

Метод функциональных диаграмм помогает создать высоко результирующие тесты. Функциональная диаграмма представляет собой формальный язык, на который транслируется спецификация, написанная на естественном языке.

Построение тестов осуществляется в несколько этапов:

- спецификация разбивается на несколько участков
- спецификация определяется причиной и следствием

Причины и следствия определяются путем последовательного чтения спецификации, каждой причине и следствию присваивается отдельный номер.

Например: Используя операторы if/else решить квадратное у равнение вида:

$$c \cdot x^2 + a \cdot x + b = 0$$

Текст программы на языке Си+ выглядит следующим образом:

```
#include<iostream.h>
#include<math.h>
void main()
{
        double a,c,b,D,x1,x2,x,x3;
        cout<<"Vvedite Chisllo c - ";
        cin>>c;
        cout<<"Vvedite Chisllo a - ";
        cin>>a;
        cout<<"Vvedite Chisllo b - ";
        cin>>b;
        D=a*a-4*c*b;
//      if (a=0)
//        else

                if (c==0)
                        if(a==0)
                        {       cout<<"Net kornei";}
                        else
                                if (b==0)
                                {
                cout<<"Mnogestvo rewenii";
                                }
        else
        {       x=-b/a;
                cout<<"\n x="<<x;
        }
                else
                        if (D<0)
                        {
                                cout<<"\n Net kornei";
                        }
                        else
                                //if (b=0)
                                if (D>0)
```

```
                {
                        x1=(-a*a+sqrt(D)/2*c);
                        x2=(-a*a-sqrt(D)/2*c);
                        cout<<"\n x1= "<<x1;
                        cout<<"\n x2= "<<x2;
                }
                else
                {
                        x3=-a/2*c;
                        cout<<"x3= "<<x3;
                }
        cout<<"\n D: "<<D;
        cout<<"\n Press any key to EXIT";
}
```

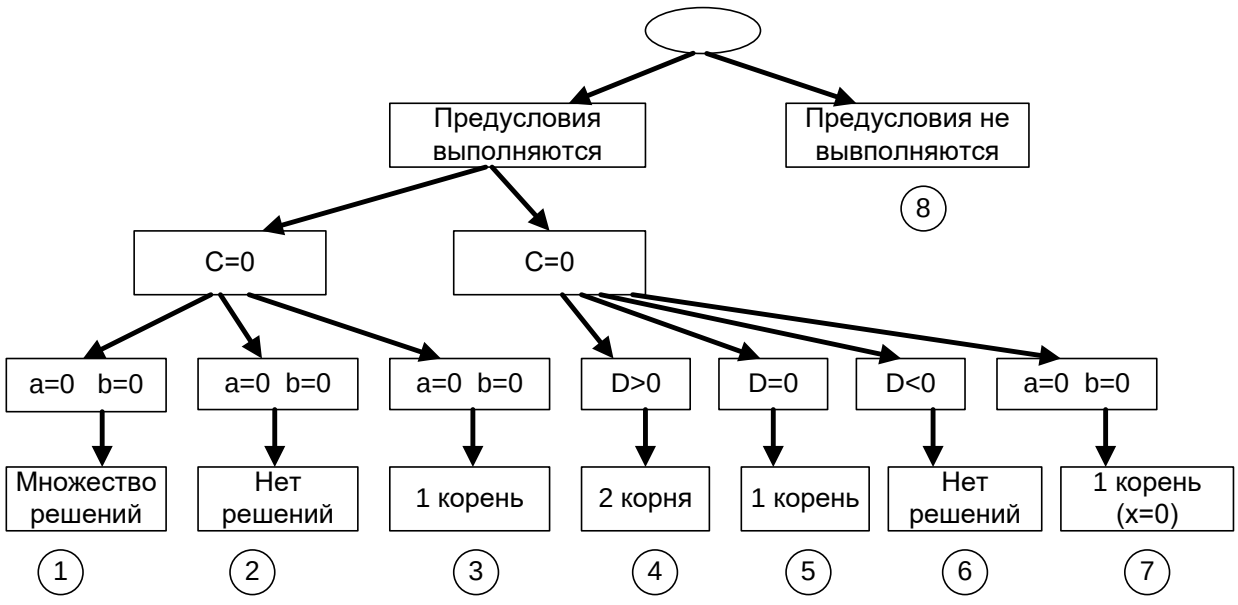
Предусловия:

1) C=0
2) C>0
3) D>0
4) D=0
5) D<0
6) b=0, a=0
7) a=0, $b \neq 0$
8) $a \neq 0$, $b \neq 0$

Постусловия:

1) Находится 2 корня
2) Находится 1 корень
3) Нет решения (корней)
4) Бесконечное множество решений

Дерево разбиений области данных бинарного поиска.



Практическое задание:

Задание

Выполнение работы предусматривает следующую последовательность действий:

1. Определение причин(условий ввода) и следствий;
2. Построение графа причинно-следственных связей;
3. Создание таблиц решений;
4. Построение тестовых вариантов;
5. Оформление результатов тестирования.

В отчет по лабораторной работе входят:

1. Перечень причин и следствий;
2. Граф причинно-следственных связей;
3. Таблица решений;
4. Тестовые варианты;
5. Результаты тестирования.

Варианты задания

Построить таблицу значений функции *y=f(x)*, *x* изменяется от x_{min} до x_{max} с шагом *dx*. Проконтролировать правильность ввода x_{min}, x_{max}, *dx* и корректность вычисляемого выражения.

Примечание. В протоколе необходимо указать порядок выполнения операций в соответствии с их приоритетом.

1. $y = \frac{a+20b}{x^3} * \ln 2x - \frac{1}{(a-1)^2}$

2. $y = 3\sqrt{\frac{5x-9}{7.5ab}+18} + e^{2x+\frac{0.5}{a}}$

3. $y = \frac{x^4 - a^3 - b^2}{\sqrt{19x-3.5} + \ln a}$

4. $y = \sin\frac{e^x - 3a}{a^2+b^2} + \frac{10}{x^3}$

5. $y = \cos\frac{(x-a)^2}{x-2a} - \frac{3.5}{\sqrt{xb}}$

6. $y = \sin\frac{a+\cos^2 x}{\cos x^2 - b} + 2.5a\sqrt{b}$

7. $y = \frac{tg3a - 20|b| - \sqrt{ab}}{x^2+b^3}$

8. $y = 2arctg\frac{25a}{b} + 3\cos^2\frac{9xb}{b-x}$

9. $y = \left|x^2 - a^2\right| + \frac{9x^3}{(b-x)^3} * \sin\frac{x}{a}$

10. $y = \left|(2a-7.5x)^3\right| + e^{\frac{2b}{x} - \frac{a}{2b}}$

11. $y = \sin 3x + \cos^2\frac{x}{2a+b} - \frac{2x}{a}$

12. $y = \ln\frac{3x^3 - 2x^2 + x}{(a^2+b)^2} + \frac{a}{x^3 - 4x^2 - x}$

13. $y = e^{|\sin(3ax+b)|} * \frac{x}{(\sqrt{ax}+bx^2)^3}$

14. $y = \sin\frac{|x|}{2\sqrt{a}} + \cos^2\frac{x^3}{a+b}$

15. $y = 2\text{ctg}\dfrac{x^3 - 2x^2 + |x|}{(a+\sqrt{b})^3} - \dfrac{1}{x}$

16. $y = \sin 5e^{\frac{x}{a}+\frac{b}{2}-\frac{3}{x}} + \cos^2 3ax$

Практическая работа 13. Отладка программ

Цель: научиться находить ошибки в программе и удалять их.

Теоретическая часть:

Ошибки, которые могут быть допущены человеком при написании программы можно разделить на три вида:

1.**синтаксические** ошибки - вызов команды, не входящей в систему команд конкретного языка программирования, обычно их обнаруживает компилятор или интерпретатор данного языка программирования;

2.**семантические** ошибки - вызов команды в ситуации, когда эта команда не может быть исполнена, эти ошибки приводят к отказу компилятора или интерпретатора работать;

3.**логические** ошибки - ЭВМ выполнила программу, но цель, поставленная человеком, не достигнута. Эти ошибки не фиксируются ни ЭВМ, ни компилятором или интерпретатором, важно понимать, что "безотказное" выполнение программы еще не означает его правильность.

Контроль правильности написанной программы состоит, как правило, из трех этапов:

1.**Просмотр**. Текст программы просматривается на предмет обнаружения описок и расхождений с алгоритмом решения. Просмотреть организацию циклов, убедиться в правильности команд, задающих количество повторений. Просмотреть логические выражения в конструкциях ветвления и выбора, вызов процедур и функций.

2.**Проверка**. При проверке программы необходимо постараться мысленно восстановить вычислительный процесс, определяемый программой, и сравнить его с требуемым процессом.

3.**Прокрутка**. Пошаговое выполнение программы вручную человеком. Для выполнения прокрутки необходимо задать исходные данные и производить над ними необходимые вычисления. Исходные данные должны подбираться так, чтобы в прокрутку вовлекалось большинство ветвей алгоритма.

Для **отладки программы** обычно пользуются приемами:

1.пошаговое выполнение программы;

Выполнение по шагам - это простейший способ выполнения программы по элементарным фрагментам. Выбор команды **Пуск|Обойти подпрограмму (клавиша F8)** вызывает выполнение отладчиком всего кода в подсвеченной строке, включая любые вызываемые на ней процедуры или функции. После этого подсвеченная строка перемещается на следующую строку программы. Выполнение по шагам позволяет проследить логику выполнения программы.

Часто при выполнении программы по шагам требуется отслеживать значения переменных. Для этого можно открыть окно просмотра значений переменных, выполнив команду **Отладка | Вывод.**

2.установка точек останова.

Точка останова - это обозначенная в коде программы позиция, в которой вы хотите прекратить выполнение программы и вернуть выполнение отладчику. В этом смысле точка останова работает аналогично команде **Go to Cursor**, при которой программа выполняется обычным путем до достижения определенной точки. Основное различие состоит в том, что вы можете задать несколько точек останова и точки останова, которые будут срабатывать не при каждом их достижении.
Чтобы установить в программе точку останова, переместите курсор на ту строку, где вы хотите остановиться. Строка должна содержать выполняемый код и не может быть комментарием, описанием или пустой строкой. Выбор команды **Точка Останова** в локальном меню окна редактирования или нажатие комбинации клавиш **Ctrl+F8.**
Для ручной настройки точек останова используется пункт меню **Отладка / Точки останова**, где самостоятельно указываете номер строки, до которой будет выполняться программа.

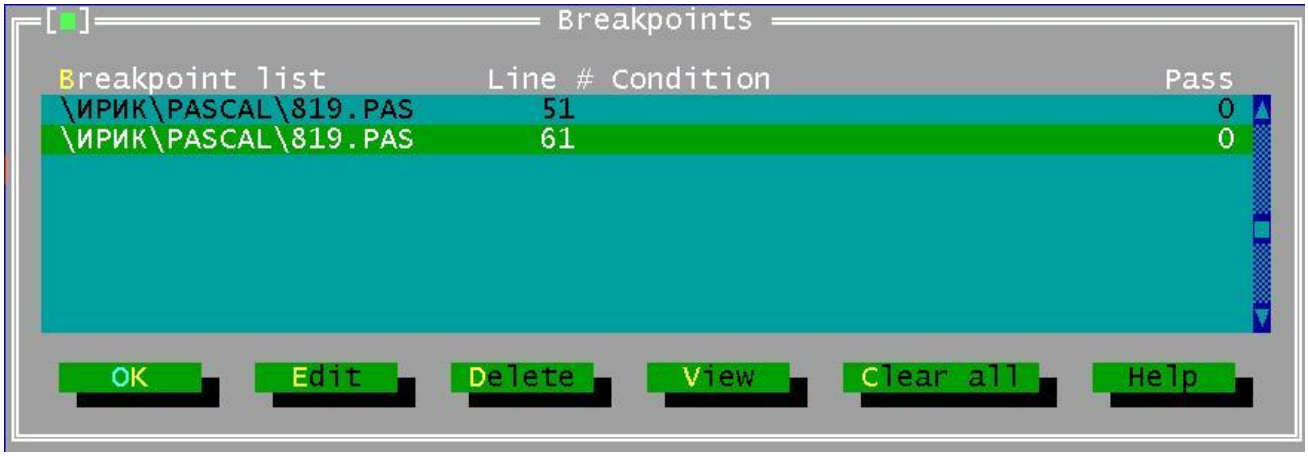


Рис.1. Окно с точками останова

Чтобы задать в программе точку, до которой вы хотите ее выполнить, а затем остановиться, используйте команду **Пуск | Выполнение до курсора** или клавишу **F4**. Позиционируйте курсор на той строке, где вы хотите возобновить управление отладкой, затем нажмите клавишу **F4**. Заметим, что вы можете сделать это как в начале сеанса отладки, так и когда уже выполните часть программы по шагам.
Даже если вы не установите точек останова, то все равно сможете выйти в отладчик при выполнении программы из IDE. В любой момент работа программы нажмите клавиши **Ctrl+Break**. Отладчик находит позицию в исходном коде, где вы прервали программу. Как и в случае обычно точки останова вы можете затем выполнить программу по шагам, трассировать ее, отследить или вычислить выражения.

Трассировка. F7
Пуск/Войти в подпрограмму Вы можете выполнить одну строку вашей программы, затем прерваться и посмотреть на результаты. При вызове процедуры или функции внутри вашей программы, Вы можете задать режим выполнения вызова как одного шага или режим трассировки этой процедуры или функции строка за строкой.
Вы можете так же трассировать вывод Вашей программы строка за строкой. Вы можете так же установить, чтобы экран переключался по необходимости или использовать два монитора. Вы можете так же установить экран вывода в отдельном окне.

3.просмотр значения любой переменной или нахождение значения любого выражения.

Задание: Решить и отладить следующую задачу. Отладку выполнить каждым из указанных способом.

Написать отчет и вывод о проделанной работе. В отчете указать результат используемого метода отладки и его применение.

Варианты заданий:

I.Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над обыкновенными дробями вида $\frac{P}{Q}$, где P - целое, Q – натуральное:

1) сложения;
2) вычитания;
3) умножения;

Используя этот модуль, решить задачу:

Дан массив A – массив обыкновенных дробей. Найти сумму всех дробей, результат представить в виде несократимой дроби. Вычислить среднее арифметическое всех дробей, результат представить в виде несократимой дроби.

II.Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над одномерными массивами:

1) заполнение массива;
2) вывод на экран массива;
3) добавление элемента в k-ю позицию;
4) удаление k-го элемента;

Используя этот модуль, решить следующую задачу:

В одномерном массиве все отрицательные элементы переместить в начало массива, а остальные – в конец с сохранением порядка следования.

III.Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над двумерными массивами:

1) заполнение массива;
2) вывод на экран массива;
3) добавление элемента в k-ю позицию;
4) суммирование элементов массива.

Используя этот модуль, решить следующую задачу:

В данной квадратной матрице порядка *n* найти сумму элементов строки, в которой расположен элемент с наименьшим значением. Предполагается, что такой элемент единственен.

IV. Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над натуральными числами в шестнадцатеричной системе счисления:

1) сложения;
2) умножения;
3) перевода из шестнадцатеричной системы счисления в десятичную;

Используя этот модуль, решить задачу.

Возвести число в степень (основание и показатель степени записаны в шестнадцатеричной системе счисления). Результат выдать в шестнадцатеричной и десятичной системах счисления.

V.Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над множествами:

1) заполнение и определение количества элементов в множестве;
2) объединение множеств;
3) пересечение множеств;
4) принадлежность элемента множеству;

Используя этот модуль, решить следующую задачу:

Даны N множеств. Найти их пересечение, объединение.

VI. Реализовать в виде модуля набор подпрограмм для выполнения следующих операций с квадратными матрицами:

1) сложения двух матриц;
2) нахождения транспонированной матрицы;
3) вычисления определителя матрицы.

Матрицу описать следующим образом:

```
Const NMax=10;
Type Matrica = Array[1..NMax,1..NMax] of Real;
```

Используя этот модуль, решить следующую задачу:

Задан массив величин типа Matrica. Отсортировать этот массив в порядке возрастания значений определителей матриц.

VII. Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над числами:

1) сложения;
2) вычитания;
3) умножения;
4) деления;

Используя этот модуль, решить следующую задачу:

Дан массив чисел A[M]. Получить матрицу B[N], каждая строка которой получается умножением на число, равное номеру этой строки, данного массива.

VIII. Реализовать в виде модуля набор подпрограмм для выполнения следующих операций над длинными числами:

1) сложения;
2) вычитания;
3) умножения;
4) нахождения частного и остатка от деления одного числа на другое;
5) функций, реализующих операции отношения (равно, не равно, больше или равно, меньше или равно, больше, меньше).

Длинное число представить следующим типом:

```
Type  Tsifra=0..9; Chislo=Array[l..1000]  Of  Tsifra;
```

Используя этот модуль, решить задачу.

Возвести число в степень (основание и показатель степени — длинные числа).

Практическая работа 14. Оптимизация программы

Цель: Научиться оптимизировать программы, используя методы и законы оптимизации.

Теоретические сведения:

Оптимизация-это улучшение программы.

Оптимизация бывает двух видов: глобальная и локальная.

Глобальная оптимизация – это оптимизация, при которой экономия ресурсов достигается путём изменения алгоритма всей программы или большей ее части.

Локальная оптимизация – это оптимизация, при которой экономия ресурсов достигается путём изменения участка программы.

Способы локальной оптимизации:

- Чистка программы
- Разгрузка участков повторения
- Реализация действий
- Упрощение действий
- Экономия памяти
- Сокращение программы

Основные законы оптимизации:

1. Делать проверки всего того, что вводит пользователь.
2. Не делать проверки внутри цикла.
   Так как циклы это слабые места любой программы, и оптимизацию нужно начинать с них.

3. Надо знать принцип работы компьютера. Чем лучше будете знать как компьютер выполняет код программы, тем лучше можно оптимизировать этот код.

Задание:

Решите задачу, используя только элементарные конструкции (последовательность, ветвления, циклы). Программа должна быть рабочей!

Оптимизировать программу: можно использовать процедуры или функции.

Внимание! Оптимизированная программа должна содержать проверки всех переменных, которые вводятся с клавиатуры.

Отчет по лабораторной работе должен содержать задачу, решенную двумя способами (два кода программы), а так же блок – схему для варианта с подпрограммой (если это возможно).

Варианты:

1. Даны натуральные числа $n,a_1,\ldots,a_n$. Определить количество членов a_k последовательности $a_1,\ldots,a_n$:
а) являющихся нечетными числами;
б) кратных 3 и не кратных 5;
в) имеющих четные порядковые номера и являющихся нечетными числами.
2. Даны натуральные числа $n,q_1,\ldots,q_n$. Найти те члены q_i, последовательности $q_1,\ldots,q_n$, которые
а) являются удвоенными нечетными числами;
б) при делении на 7 дают остаток 1, 2 или 5;
в) обладают тем свойством, что корни уравнения x^2+3q_i-5 действительны и положительны.
3. Дано натуральное число n.Получить сумму тех чисел вида i^3-3in^2+n (i=1,2,…,n), которые являются утроенными нечетными
4. Даны целые числа A1, ..., A80. Получить сумму тех чисел данной последовательности, которые
а) кратны 5;
б) нечетны и отрицательны;
в) удовлетворяют условию $|Ai]<i^2$.
5. Даны натуральное число *n,* целые числа A1, ..., An. Найти количество и сумму тех членов

данной последовательности, которые делятся на 5 и не делятся на 7.
6. Даны натуральные числа *n, p,* целые числа A1 ..., An,. Получить произведение членов последовательности A1, ..., An, кратных *p.*
7. Даны целые числа *p, q,* A1 ...,A67 (p>q>=0). В последовательности A1 …A67 заменить нулями члены, модуль которых при делении на *p* дает в остатке *q.*
8. Даны натуральное число n, действительные числа A1., An. Получить удвоенную сумму всех положительных членов последовательности A1,…An.
9. Даны натуральное число *n,* действительные числа A1 ..., An. Вычислить обратную величину произведения тех членов Ai последовательности *A1* ...,An, для которых выполнено i+ 1 < Ai <i!.
10. Даны натуральное число *n,* действительные числа A1, ...,An. В последовательности A1, ...,An все отрицательные члены увеличить на 0.5, а все неотрицательные заменить на 0.1.
11. Даны натуральное число *n,* действительные числа X1 ..., Xn. В последовательности X1, . . ., Xn все члены, меньшие двух, заменить нулями. Кроме того, получить сумму членов, принадлежащих отрезку [3, 7], а также, число таких членов.
12. Даны натуральное число *n,* действительные числа A1 ...,An. В последовательности A1 ..., An все неотрицательные члены, не принадлежащие отрезку [1, 2], заменить на единицу. Кроме того, получить число отрицательных членов и число членов, принадлежащих отрезку [1, 2].
13. Даны натуральное число *n,* целые числа A1, ...An Получить сумму положительных и число отрицательных, членов последовательности A1..., An.
14. Даны целые числа A1 ....An. Получить число отрицательных членов последовательности A1 ., An и число нулевых членов- всей последовательности A1....An.
15. Даны натуральное число n, целые числа a, X1...,Xn. Если в последовательности X1, ..., Xn есть хотя бы один член, равный *a,* то получить сумму всех членов, следующих за первым таким членом; в противном случае ответом должно быть число —10.

Раздел 4. Коллективная разработка программных средств

Тема1: Коллективная работа над созданием программного обеспечения. Обоснование выбора среды реализации.

КОЛЛЕКТИВНАЯ РАБОТА ПО СОЗДАНИЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ Разработка программного обеспечения, кроме достаточно сложного технического аспекта, имеет сложный организационный или даже пси. ... Бизнес-аналитик — это представитель заказчика, ответственный за создание системы. Он досконально знает бизнес-процесс своего предприятия и обязан предоставлять полную и достоверную информацию. Сторона разработчика: менеджер программы, аналитик, программист, тестировщик. Менеджер программы — это главный технический специалист проекта, он является хранителем спецификации и заинтересован в ее минимальных изменениях во время разработки.

Коллективная разработка Коллективная разработка – это создание программного продукта коллективом разработчиков. Один из основных вопросов коллективной разработки является разделение труда. Один человек не способен создать приложение масштаба предприятия. ... Основной моделью разделения труда в наше время является иерархическая структура. 6 слайд.

Иерархическая модель Если в современных производственных средах один менеджер проекта (начальник) отвечает за все тонкости разработки и принимает все важные решения, возникает множество проблем, ведущих к провалу проекта.

Коллективная разработка программного обеспечения. Преобладающим методом коллективной разработки является организация бригад. Метод предметных связей используется реже, тогда когда ведется разработка не тиражируемого продукта силами организации, которая будет и вести его эксплуатацию. С административной точки зрения, бригада - структурная производственная единица, которая может быть образована в рамках существующего структурного подразделения для выполнения определенной работы, имеющей конкретный результат.

Авторская разработка – принцип создания программных продуктов, при котором весь жизненный цикл разработки поддерживается одним единственным человеком. Этот принцип был распространен в 70 – 80-е годы XX века. Сейчас он применяется редко. ... - исключения работ по разбиению проекта на составляющие, по распределению их между исполнителями, по координации деятельности исполнителей и контролю за их работой. ... 2. Коллективная разработка. Одним из основных вопросов коллективной разработки является разделение труда – от равноправных соисполнителей до организации в виде жесткой иерархии (например, бригада главного программиста).

Работа начинается с создания тестов модуля, она должна предшествовать программированию модуля. Тесты необходимо помещать в библиотеку кодов вместе с кодом, который они тестируют. Тесты делают возможным коллективное создание кода и защищают код от неожиданных изменений. ... Программная инженерия как некоторое направление возникло и формировалось под давлением роста стоимости создаваемого программного обеспечения. Главная цель этой области знаний – сокращение стоимости и сроков разработки программ. Программная инженерия прошла несколько этапов развития, в процессе которых были сформулированы фундаментальные принципы и методы разработки программных продуктов.

Обоснование выбора инструментальной среды разработки программного обеспечения. В качестве языка программирования для реализации данной работы был выбран язык C++. Этот выбор обусловлен следующими особенностями языка: 1) возможность генерации высокоэффективного программного кода; 2) поддерживаются различные стили и технологии программирования, включая традиционное директивное

программирование, ООП, обобщённое программирование, метапрограммирование (шаблоны, макросы)

Тема2: Коллективная разработка программного продукта. Реализация программного продукта.

Объем программного продукта, выполненного методом авторской разработки, в 5-20 раз меньше по сравнению с индустриальными аналогами. Авторская разработка предполагает достижение профессионального успеха, известности и славы в одиночку. Это реально, если правильно выбрать профессиональную «нишу», область ведения разработки. 2. Коллективная разработка. Одним из основных вопросов коллективной разработки является разделение труда – от равноправных соисполнителей до организации в виде жесткой иерархии (например, бригада главного программиста). Равноправные соисполнители.

Авторская разработка — принцип создания программных продуктов, при котором весь жизненный цикл разработки поддерживается одним — единственным человеком. Этот принцип был достаточно широко распространен в 70 — 80-е годы XX века. Сейчас он применяется редко. ... Коллективная разработка программного обеспечения. Автор: Пользователь скрыл имя, 28 Октября 2013 в 13:43, реферат. Описание работы. Авторская разработка — принцип создания программных продуктов, при котором весь жизненный цикл разработки поддерживается одним — единственным человеком. Этот принцип был достаточно широко распространен в 70 — 80-е годы XX века. Сейчас он применяется редко.

Технология коллективной разработки. Классификация методов разработки программного обеспечения в зависимости от количества участников и типов взаимоотношений в коллективе. Оптимальный состав группы управления, специалистов по инженерии программирования и отладке программного обеспечения. Рубрика. ... Объем программного продукта, выполненного методом авторской разработки, в 5-20 раз меньше по сравнению с индустриальными аналогами. Авторская разработка предполагает достижение профессионального успеха, известности и славы в одиночку. Такое вполне реально, следует только правильно выбрать профессиональную "нишу", область ведения разработки.

Коллективная разработка Коллективная разработка – это создание программного продукта коллективом разработчиков. Один из основных вопросов коллективной разработки является разделение труда. Один человек не способен создать приложение масштаба предприятия. ... И у коллективного подхода есть недостатки: разрозненная связь с внешними источниками информации;

Коллективная разработка программного обеспечения. Преобладающим методом коллективной разработки является организация бригад. Метод предметных связей используется реже, тогда когда ведется разработка не тиражируемого продукта силами организации, которая будет и вести его эксплуатацию. С административной точки зрения, бригада - структурная производственная единица, которая может быть образована в рамках существующего структурного подразделения для выполнения определенной работы, имеющей конкретный результат.

Тема3: Разработка пользовательского интерфейса.Проектная деятельность

Разработка пользовательского интерфейса. Для обеспечения успешного взаимодействия с пользователем требуется сбалансированный подход на протяжении всего жизненного цикла разработки. Чтобы обеспечить этот баланс, необходимо не только сосредоточиться на реализации функций, необходимых для выполнения задачи, но и о том, как задача предоставляется через пользовательский интерфейс. Помните, что пользовательский интерфейс должен быть не только функциональным, но и может быть пригоден для использования.

Разработка пользовательского интерфейса видео редактора. .7 Структура проекта. .8 Запуск и отладка сервера. ... Программы позволяющие редактировать видео, достаточно востребованы как для профессиональной деятельности, так и для создания любительских видео роликов. В связи с этим, существует достаточно большое количество приложений, позволяющих обрабатывать, редактировать и конвертировать видео, как профессионалам, так и обычным пользователям.

Проектирование вычислительных систем охватывает широкий спектр проектных действий — от проектирования аппаратных средств до проектирования интерфейса пользователя. Организации-разработчики часто нанимают специалистов для проектирования аппаратных средств и очень редко для проектирования интерфейсов. Интерфейс. Текстовый. Графический. ... На схеме изображен итерационный процесс проектирования пользовательского интерфейса. Наиболее эффективным подходом к проектированию интерфейса пользователя является разработка с применением моделирования пользовательских функций. Изучение и анализ действий. пользователя.

Разработка пользовательского интерфейса. Для обеспечения успешного взаимодействия с пользователем требуется сбалансированный подход на протяжении всего жизненного цикла разработки. Чтобы обеспечить этот баланс, необходимо не только сосредоточиться на реализации функций,

необходимых для выполнения задачи, но и о том, как задача предоставляется через пользовательский интерфейс. Помните, что пользовательский интерфейс должен быть не только функциональным, но и может быть пригоден для использования.

роектирование графического интерфейса пользователя. Интерфейсы *. Из песочницы. Введение. В современном мире миллиарды вычислительных устройств. Еще больше программ для них. И у каждой свой интерфейс, являющийся «рычагами» взаимодействия между пользователем и машинным кодом. Не удивительно, что чем лучше интерфейс, тем эффективнее взаимодействие. ... Какие ЭИ создать? Разработка интерфейса обычно начинается с определения задачи или набора задач, для которых продукт предназначен. Простое должно оставаться простым. Не усложняйте интерфейсы.

Тема4: Разработка прототипа проекта. Проектная деятельность.

Прототипы — это схематичный набросок, эскиз страниц сайта с изображенными на нем элементами. К примеру, кнопки, «меню», формы и вариации. Прототипом можно назвать промежуточный и упрощенный вариант программной системы, позволяющий представить Заказчику, как будет выглядеть и работать законченная система. ... Скептики могут возразить, что прототипирование не представляет достаточного интереса, так как многие проекты имеют почти аналогичную схему расположения блоков и элементов, их просто можно продублировать. Но это далеко не так. Все дело в том, что потребители отличаются друг от друга — их восприятие и мышление может быть разным.

Как мы разрабатываем прототипы. Прототипирование ускоряет создание продукта и помогает значительно уменьшить количество ошибок и переделок при дальнейшей разработке. Делимся своим опытом в прототипировании. Нравится. 0. ... Прототип — это быстрое визуальное представление будущего продукта. Прототип позволяет увидеть проектируемую систему на самом раннем этапе. Протипирование может занимать половину времени разработки интерфейса проекта. На этом этапе происходит утрясание концепции, поиск решений и корректировка первоначальных планов. Прототип сразу дает представление о продукте, экономит деньги и время при разработке. Прототип плюс ТЗ.

Прототипы, как средство быстрой визуализации идей, в последние годы получили большое распространение и оказали позитивное влияние на качество цифровых продуктов в целом. Вместе с тем, взяв на вооружение инструменты прототипирования, рынок все еще не до конца усвоил, что прототип – часть юзабилити-процесса, а не проджект-менеджмента. За

последние несколько лет я общался со многими дизайнерами, у которых в портфолио есть пара десятков прототипов, при этом часто они не только не понимают до конца процесс создания прототипов, но и не всегда уверенно могут объяснить зачем они вообще нужны.

Зачем делают прототипы Типы прототипов Этапы прототипирования Инструменты для разработки макетов. Быстрый старт в email-маркетинге. ... Прототипирование — это один из начальных этапов разработки, в ходе которого создается предварительный дизайн сайта, лендинга, приложения или другого проекта. В ходе прототипирования создается макет, который имитирует взаимодействие пользователя с интерфейсом проекта. Прототип нужен для презентации проекта заказчику и оценки его юзабилити. Тестирование такого макета позволяет заранее выявить и устранить ошибки, прежде чем вкладывать деньги в разработку конечного дизайнерского решения и кода.

Прототип — это интерактивный проект, который создан без кода, с целью проверки концепции перед запуском. С помощью прототипирования можно анализировать различные дизайнерские решения без участия разработчиков. Это также позволяет тестировать прототипы с реальными пользователями, чтобы выявить проблемы юзабилити перед выпуском конечного продукта. Этот процесс помогает дизайнерам сэкономить много времени, денег и усилий на протяжении всего жизненного цикла проектирования и разработки.

1 Сұрақ
Вопрос
Проектирование программного обеспечения - это

A. комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
B. набор компьютерных программ, процедур, а также связанных с ними документации и данных.
C. совокупность процессов и методов создания программного продукта.
D. полный набор (программное обеспечение) или часть (программные средства) программ, процедур, правил и связанной с ними документации системы обработки информации

2 Сұрақ
Вопрос
Система (system) - это

A. комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
B. набор компьютерных программ, процедур, а также связанных с ними документации и данных.
C. совокупность процессов и методов создания программного продукта.
D. полный набор (программное обеспечение) или часть (программные средства) программ, процедур, правил и связанной с ними документации системы обработки информации

3. Сұрақ
Вопрос
Нотация (notation) - это

A. комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
B. набор компьютерных программ, процедур, а также связанных с ними документации и данных.
C. система графических обозначений для записи промежуточных и конечного результатов разработки ПС
D. полный набор (программное обеспечение) или часть (программные средства) программ, процедур, правил и связанной с ними документации системы обработки информации

4. Сұрақ
Вопрос
Организационные процессы жизненного цикла - это

A. комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
B. это процессы, предназначенные для создания в некоторой организации и совершенствования организационных структур, охватывающих процессы жизненного цикла и соответствующий персонал.
C. это процессы, которые реализуются под управлением основных сторон, участвующих в жизненном цикле программных средств.

D. это процессы, являющиеся целенаправленными составными частями других процессов и предназначенные для обеспечения успешной реализации и качества выполнения

5. Сұрақ
Вопрос
Каскадная стратегия разработки программных средств и систем

A. комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
B. представляет собой однократный проход этапов разработки. Данная стратегия основана на полном определении всех требований к разрабатываемому программному средству или системе в начале процесса разработки. Каждый этап разработки начинается после завершения предыдущего этапа. Возврат к уже выполненным этапам не предусматривается. Промежуточные продукты разработки в качестве версии программного средства (системы) не распространяются
C. представляет собой многократный проход этапов разработки с запланированным улучшением результата. Данная стратегия основана на полном определении всех требований к разрабатываемому программному средству (системе) в начале процесса разработки. Однако полный набор требований реализуется постепенно в соответствии с планом в последовательных циклах разработки. Результат каждого цикла называется инкрементом.
D. представляет собой многократный проход этапов разработки. Данная стратегия основана на частичном определении требований к разрабатываемому программному средству или системе в начале процесса разработки. Требования постепенно уточняются в последовательных циклах разработки. Результат каждого цикла разработки обычно представляет собой очередную поставляемую версию программного средства или системы.

6. Сұрақ
Вопрос
Стандарт – это

A. исходный документ на проектирование технического объекта. Устанавливает основное назначение разрабатываемого объекта, его технические характеристики, показатели качества и технико-экономические требования, предписание по выполнению необходимых стадий создания документации (конструкторской, технологической, программной и т. д.) и её состав, а также специальные требования.
B. анализ требований, предъявляемых к системе; определение спецификаций; проектирование; кодирование; тестирование; комплексное; эксплуатация и сопровождение.
C. Нормативный документ, который разработан на основе согласия сторон и утвержденный уполномоченным органом, в котором определяются для длительного и постоянного пользования правила, характеристики или общие принципы, затрагивающие разные виды деятельности или их результат. Задача этого документа достичь наилучшей степени упорядочения в заданной области.
D. система графических обозначений для записи промежуточных и конечного результатов разработки ПС

7. Сұрақ
Вопрос
Справочник по применению ПС

A. Предназначен для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для избирательного поиска отдельных деталей.
B. Предназначена для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для ее изучения.
C. Предназначено для администраторов ПС. Оно должно детально предписывать, как устанавливать системы в конкретной среде, в частности, должно содержать описание компьютерно-считываемого носителя, на котором поставляется ПС, файлы, представляющие ПС, и требования к минимальной конфигурации аппаратуры.
D. Дает краткую характеристику функциональных возможностей ПС. Предназначено для пользователей, которые должны решить, насколько необходимо им данное ПС

8. Сұрақ
Вопрос
База данных - это:
A. совокупность данных, организованных по определенным правилам;
B. совокупность программ для хранения и обработки больших массивов информации;
C. интерфейс, поддерживающий наполнение и манипулирование данными;
D. определенная совокупность информации.

9. Сұрақ
Вопрос
Как соотносятся диаграммы кооперации и диаграммы объектов?
A. диаграмма объектов и диаграмма кооперации полностью взаимозаменяемы
B. UML-модель не может содержать диаграммы кооперации и диаграммы объектов одновременно
C. диаграмма объектов показывает статику, а диаграмма взаимодействия описывает динамические аспекты системы
D. использование диаграммы кооперации или диаграммы объектов зависит только от особенностей стиля проектировщика

10. Сұрақ
Вопрос
Какие артефакты пришли на смену техническому заданию?
A. диаграммы компонентов
B. диаграммы классов
C. диаграммы прецедентов
D. диаграммы развертывания

11. Сұрақ
Вопрос
Сколько конечных состояний может содержать диаграмма активностей?
A. не больше двух
B. больше одного
C. только одно
D. столько же, сколько на диаграмме начальных состояний

12. Сұрақ
Вопрос
Разновидностью какой диаграммы UML являются диаграммы активностей?
A. диаграммы классов
B. диаграммы состояний

C. диаграммы последовательностей
D. диаграммы развертывания

13. Сұрақ
Вопрос
Программный модуль (software unit) - это
A. отдельно компилируемая часть программного кода (программы).
B. набор компьютерных программ, процедур, а также связанных с ними документации и данных.
C. совокупность процессов и методов создания программного продукта.
D. полный набор (программное обеспечение) или часть (программные средства) программ, процедур, правил и связанной с ними документации системы обработки информации

14. Сұрақ
Вопрос
Жизненный цикл программного средства или системы - это
A. комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
B. набор компьютерных программ, процедур, а также связанных с ними документации и данных.
C. совокупность процессов, работ и задач, включающая в себя разработку, эксплуатацию и сопровождение программного средства или системы и охватывающая их жизнь от формулирования конценции до прекращения использования.
D. полный набор (программное обеспечение) или часть (программные средства) программ, процедур, правил и связанной с ними документации системы обработки информации

15. Сұрақ
Вопрос
Основные процессы жизненного цикла
A. внешние и внутренние
B. заказ; поставка; разработка; эксплуатация; сопровождение.
C. документирование; управление конфигурацией; обеспечение качества; верификация; аттестация; совместный анализ; аудит; решение проблем.
D. управлепие; создание инфраструктуры; усовершенствование; обучение.

16. Сұрақ
Вопрос
Инкрементная стратегия разработки программных средств и систем
A. комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
B. представляет собой однократный проход этапов разработки. Данная стратегия основана на полном определении всех требований к разрабатываемому программному средству или системе в начале процесса разработки. Каждый этап разработки начинается после завершения предыдущего этапа. Возврат к уже выполненным этапам не предусматривается. Промежуточные продукты разработки в качестве версии программного средства (системы) не распространяются
C. представляет собой многократный проход этапов разработки с запланированным улучшением результата. Данная стратегия основана на полном определении всех

требований к разрабатываемому программному средству (системе) в начале процесса разработки. Однако полный набор требований реализуется постепенно в соответствии с планом в последовательных циклах разработки. Результат каждого цикла называется инкрементом.

D. представляет собой многократный проход этапов разработки. Данная стратегия основана на частичном определении требований к разрабатываемому программному средству или системе в начале процесса разработки. Требования постепенно уточняются в последовательных циклах разработки. Результат каждого цикла разработки обычно представляет собой очередную поставляемую версию программного средства или системы.

17. Сұрақ

Вопрос

Спецификация требований программного обеспечения – это

A. исходный документ на проектирование технического объекта. Устанавливает основное назначение разрабатываемого объекта, его технические характеристики, показатели качества и технико-экономические требования, предписание по выполнению необходимых стадий создания документации (конструкторской, технологической, программной и т. д.) и её состав, а также специальные требования.

B. анализ требований, предъявляемых к системе; определение спецификаций; проектирование; кодирование; тестирование; комплексное; эксплуатация и сопровождение.

C. нормативный документ, который разработан на основе согласия сторон и утвержденный уполномоченным органом, в котором определяются для длительного и постоянного пользования правила, характеристики или общие принципы, затрагивающие разные виды деятельности или их результат. Задача этого документа достичь наилучшей степени упорядочения в заданной области.

D. законченное описание поведения программы, которую требуется разработать, включает ряд пользовательских сценариев, которые описывают все варианты взаимодействия между пользователями и программным обеспечением.

18. Сұрақ

Вопрос

Руководство по управлению ПС

A. Предназначен для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для избирательного поиска отдельных деталей.

B. Предназначено для администраторов ПС. Оно должно описывать сообщения, генерируемые, когда ПС взаимодействует с другими системами, и как должен реагировать администратор на эти сообщения. Кроме того, если ПС использует системную аппаратуру, этот документ может объяснять, как сопровождать эту аппаратуру.

C. Предназначено для администраторов ПС. Оно должно детально предписывать, как устанавливать системы в конкретной среде, в частности, должно содержать описание компьютерно-считываемого носителя, на котором поставляется ПС, файлы, представляющие ПС, и требования к минимальной конфигурации аппаратуры.

D. Дает краткую характеристику функциональных возможностей ПС. Предназначено для пользователей, которые должны решить, насколько необходимо им данное ПС

19. Сұрақ

Вопрос

Наиболее распространенными в практике являются:

A. распределенные базы данных
B. иерархические базы данных
C. сетевые базы данных
D. реляционные базы данных

20. Сұрақ
Вопрос
Программные средства, программное обеспечение (software)- это
A. комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
B. набор компьютерных программ, процедур, а также связанных с ними документации и данных.
C. совокупность процессов и методов создания программного продукта.
D. полный набор (программное обеспечение) или часть (программные средства) программ, процедур, правил и связанной с ними документации системы обработки информации

21. Сұрақ
Вопрос
Программный продукт (software product)- это
A. комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
B. набор компьютерных программ, процедур, а также связанных с ними документации и данных.
C. совокупность процессов и методов создания программного продукта.
D. полный набор (программное обеспечение) или часть (программные средства) программ, процедур, правил и связанной с ними документации системы обработки информации

22. Сұрақ
Вопрос
Процессы жизненного цикла
A. внешние и внутренние
B. основные, вспомогательные, организационные
C. нотация, программный модуль
D. полный набор (программное обеспечение) или часть (программные средства) программ, процедур, правил и связанной с ними документации системы обработки информации

23. Сұрақ
Вопрос
Вспомогательные процессы жизненного цикла
A. внешние и внутренние
B. заказ; поставка; разработка; эксплуатация; сопровождение.
C. документирование; управление конфигурацией; обеспечение качества; верификация; аттестация; совместный анализ; аудит; решение проблем.
D. управлепие; создание инфраструктуры; усовершенствование; обучение.

24. Сұрақ
Вопрос

Эволюционная стратегия разработки программных средств и систем

A. комплекс, состоящий из процессов, технических и программных средств, устройств и персонала, обладающий возможностью удовлетворять установленным потребностям или целям.
B. представляет собой однократный проход этапов разработки. Данная стратегия основана на полном определении всех требований к разрабатываемому программному средству или системе в начале процесса разработки. Каждый этап разработки начинается после завершения предыдущего этапа. Возврат к уже выполненным этапам не предусматривается. Промежуточные продукты разработки в качестве версии программного средства (системы) не распространяются
C. представляет собой многократный проход этапов разработки с запланированным улучшением результата. Данная стратегия основана на полном определении всех требований к разрабатываемому программному средству (системе) в начале процесса разработки. Однако полный набор требований реализуется постепенно в соответствии с планом в последовательных циклах разработки. Результат каждого цикла называется инкрементом.
D. представляет собой многократный проход этапов разработки. Данная стратегия основана на частичном определении требований к разрабатываемому программному средству или системе в начале процесса разработки. Требования постепенно уточняются в последовательных циклах разработки. Результат каждого цикла разработки обычно представляет собой очередную поставляемую версию программного средства или системы.

25. Сұрақ
Вопрос

Общее функциональное описание ПС

A. Предназначен для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для избирательного поиска отдельных деталей.
B. Предназначена для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для ее изучения.
C. Предназначено для администраторов ПС. Оно должно детально предписывать, как устанавливать системы в конкретной среде, в частности, должно содержать описание компьютерно-считываемого носителя, на котором поставляется ПС, файлы, представляющие ПС, и требования к минимальной конфигурации аппаратуры.
D. Дает краткую характеристику функциональных возможностей ПС. Предназначено для пользователей, которые должны решить, насколько необходимо им данное ПС

26. Сұрақ
Вопрос

Руководство по сопровождению ПС

A. Предназначен для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для избирательного поиска отдельных деталей.
B. Предназначено для администраторов ПС. Оно должно описывать сообщения, генерируемые, когда ПС взаимодействует с другими системами, и как должен реагировать администратор на эти сообщения. Кроме того, если ПС использует системную аппаратуру, этот документ может объяснять, как сопровождать эту аппаратуру.
C. Предназначено для администраторов ПС. Оно должно детально предписывать, как устанавливать системы в конкретной среде, в частности, должно содержать описание

компьютерно-считываемого носителя, на котором поставляется ПС, файлы, представляющие ПС, и требования к минимальной конфигурации аппаратуры.

D. Описывает особенности реализации ПС (в частности, трудности, которые пришлось преодолевать) и как учтены возможности развития ПС в его строении (конструкции). В нем также фиксируются, какие части ПС являются аппаратно- и программно-зависимыми.

27. Сұрақ
Вопрос
Этапы разработки программного обеспечения

A. внешние и внутренние
B. анализ требований, предъявляемых к системе; определение спецификаций; проектирование; кодирование; тестирование; комплексное; эксплуатация и сопровождение.
C. основные, вспомогательные, организационные
D. каскадный; инкрементный; эволюционный.

28. Сұрақ
Вопрос
Руководство по инсталяции ПС

A. Предназначен для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для избирательного поиска отдельных деталей.
B. Предназначена для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для ее изучения.
C. Предназначено для администраторов ПС. Оно должно детально предписывать, как устанавливать системы в конкретной среде, в частности, должно содержать описание компьютерно-считываемого носителя, на котором поставляется ПС, файлы, представляющие ПС, и требования к минимальной конфигурации аппаратуры.
D. Дает краткую характеристику функциональных возможностей ПС. Предназначено для пользователей, которые должны решить, насколько необходимо им данное ПС

29. Сұрақ
Вопрос
Стадии тестирования ПС

A. внешние и внутренние
B. заказ; поставка; разработка; эксплуатация; сопровождение.
C. основные, вспомогательные, организационные
D. автономное; комплексное и системное.

30. Сұрақ
Вопрос
Начало какого этапа жизненного цикла ПО знаменует собой создание диаграммы классов?

A. внедрения
B. анализа
C. тестирования
D. проектирования

31. Сұрақ
Вопрос
Что такое требование к ПО?

A. формальные критерии соответствия системы желаниям заказчика
B. желаемая функциональность, свойство или поведение системы
C. формальное описание внутреннего устройства будущей системы
D. условия, ограничивающие функциональность будущей системы

32. Сұрақ
Вопрос
Базовые стратегии разработки ПО
A. внешние и внутренние
B. заказ; поставка; разработка; эксплуатация; сопровождение.
C. основные, вспомогательные, организационные
D. каскадная; инкрементная; эволюционная.

33 Сұрақ
Вопрос
Техническое задание – это
A. исходный документ на проектирование технического объекта. Устанавливает основное назначение разрабатываемого объекта, его технические характеристики, показатели качества и технико-экономические требования, предписание по выполнению необходимых стадий создания документации (конструкторской, технологической, программной и т. д.) и её состав, а также специальные требования.
B. анализ требований, предъявляемых к системе; определение спецификаций; проектирование; кодирование; тестирование; комплексное; эксплуатация и сопровождение.
C. основные, вспомогательные, организационные
D. каскадный; инкрементный; эволюционный.

34. Сұрақ
Вопрос
Инструкция по применению ПС
A. Предназначен для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для избирательного поиска отдельных деталей.
B. Предназначена для ординарных пользователей. Содержит необходимую информацию по применению ПС, организованную в форме удобной для ее изучения.
C. Предназначено для администраторов ПС. Оно должно детально предписывать, как устанавливать системы в конкретной среде, в частности, должно содержать описание компьютерно-считываемого носителя, на котором поставляется ПС, файлы, представляющие ПС, и требования к минимальной конфигурации аппаратуры.
D. Дает краткую характеристику функциональных возможностей ПС. Предназначено для пользователей, которые должны решить, насколько необходимо им данное ПС

35. Сұрақ
Вопрос
Сбой системы — это
A. явление, связанное с нарушением системой установленных на нее спецификаций.
B. алгоритмический дефект, который создает выброс
C. данные, при обработке которых правильными алгоритмами системы происходит сбой
D. выполнение доказательств, что программа удовлетворяет своим спецификациям.

36. Сұрақ
Вопрос

Что такое диаграмма взаимодействия?

A. диаграмма, на которой представлено взаимодействие, состоящее из сообщений, которыми обмениваются элементы модели
B. диаграмма, на которой представлено взаимодействие, состоящее из множества объектов одного класса и сообщений, которыми они обмениваются
C. диаграмма, на которой представлено взаимодействие, состоящее из множества объектов одного класса и его подклассов и сообщений, которыми они обмениваются
D. диаграмма, на которой представлено взаимодействие, состоящее из множества объектов и отношений между ними, включая и сообщения, которыми они обмениваются

37. Сұрақ
Вопрос
Программный продукт – это
A. программа для удовлетворения нужд разработчиков, предназначенная для продажи
B. комплекс взаимосвязанных программ для решения определенной проблемы массового спроса, подготовленный к реализации как любой вид промышленной продукции
C. программная реализация решения задачи на компьютере
D. результат разработки какого-либо технического задания

38. Сұрақ
Вопрос
Отличительной особенностью программных продуктов является
A. системность
B. простота
C. универсальность
D. надежность

39. Сұрақ
Вопрос
Сопровождение программного продукта – это

A. снабжение программного продукта необходимой документацией
B. обнаружение и исправление ошибок
C. поддержка работоспособности программного продукта, переход на его новые версии, внесение изменений, исправление обнаруженных ошибок и т.д.
D. проверка работоспособности каждой разработанной функции, процедуры, модуля

40. Сұрақ
Вопрос
Мобильность программных продуктов – это

A. независимость от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п.
B. точность выполнения предписанных функций обработки
C. способность к внесению изменений
D. обеспечение дружественного интерфейса для работы конечного пользователя, наличие контекстно-зависимой подсказки или обучающей системы в составе программного средства

41. Сұрақ
Вопрос
В условиях существования рынка программных продуктов важными его характеристиками являются:

A. количество продаж, наличие программ-конкурентов, длительность продаж
B. стоимость, количество продаж, время нахождения на рынке, известность фирмы-разработчика и программы
C. внешний интерфейс программы, количество продаж, наличие программ-конкурентов
D. модифицируемость, надежность, универсальность, известность фирмы - разработчика

42. Сұрақ
Вопрос
Утилитарные программы выполняют роль…
A. сервиса
B. клиента
C. сервера
D. программного средства разработки приложений

43. Сұрақ
Вопрос
Основными показателями качества программных продуктов является:
A. алгоритмическая сложность, полнота и системность функций обработки, объем файлов программы
B. стоимость, количество продаж, наличие программных продуктов аналогичного назначения
C. мобильность, надежность, эффективность, модифицируемость, коммуникативность, учет человеческого фактора
D. модифицируемость, надежность, наличие программных продуктов аналогичного назначения

44. Сұрақ
Вопрос
Функциональные задачи – это
A. задачи, требующие решения при реализации функций управления в рамках информационных систем предметных областей
B. основа для разработки сервисных средств ПО (утилиты, библиотеки)
C. совокупность связанных между собой функций и задач управления, с помощью которых достигается выполнение поставленных целей
D. задачи, которые ставятся и решаются при организации технологического процесса обработки информации на компьютере

45. Сұрақ
Вопрос
Алгоритм – это
A. комплекс математических вычисление для решения задачи
B. последовательность команд, предназначенных для решения задач
C. программная реализация на компьютере решения определенной задачи
D. результат интеллектуального труда, для которого характерно творчество

46. Сұрақ
Вопрос
При индивидуальной разработке фирма-разработчик создает программный продукт для…
A. конкретного заказчика
B. массового использования
C. внедрения в специальные организации

D. для удовлетворения собственных нужд

47. Сұрақ
Вопрос
Модифицируемость программных продуктов означает…
A. независимость от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п.
B. точность выполнения предписанных функций обработки
C. способность к внесению изменений, например расширение функций обработки, переход на другую техническую базу обработки и т.п.
D. обеспечение дружественного интерфейса для работы конечного пользователя, наличие контекстно-зависимой подсказки или обучающей системы в составе программного средства

48. Сұрақ
Вопрос
Жизненный цикл программы – это
A. временной интервал, начиная с момента замысла программы и кончая прекращением всех видов его пользований
B. временной интервал, начиная с момента введения программы в эксплуатацию
C. промежуток времени, который определяет наиболее эффективное использование создаваемой программы
D. временная характеристика разработки программного продукта

49. Сұрақ
Вопрос
Программы малого Жизненного Цикла – это программы
A. когда время разработки программы значительно меньше времени эксплуатации программы
B. когда время разработки программы значительно больше времени использования программы
C. когда время разработки программы равно времени эксплуатации программы
D. нет правильного ответа

50. Сұрақ
Вопрос
Выбрать правильный ответ
A. На этапе сбора и анализа требований заказчик должен выяснить, прежде всего, необходимость обеспечения безопасности системы и данных
B. На этапе сбора и анализа требований заказчик должен выяснить, прежде всего, функции, которые должен выполнять программный продукт
C. На этапе сбора и анализа требований заказчик должен выяснить, прежде всего, сроки написания программы
D. На этапе сбора и анализа требований заказчик должен собрать литературу по разрабатываемому программному продукту

51. Сұрақ
Вопрос
Самая распространенная модель Жизненного цикла программного продукта это
A. итерационная
B. V - образная
C. спиральная

D. каскадная

52. Сұрақ
Вопрос
Классическая модель ЖЦПО характеризуется следующими основными особенностями
A. последовательным выполнением входящих в ее состав этапов
B. наличием обратных связей между этапами
C. отсутствием временного перекрытия этапов
D. отсутствием (или определенным ограничением) возврата к предыдущим этапам
E. наличием результата после каждого этапа разработки

53. Сұрақ
Вопрос
Выберите правильную последовательность этапов спиральной модели жизненного цикла программного продукта:
A. техническое проектирование, сопровождение ПП, сбор и анализ требований заказчика, кодирование, уточнение функциональных характеристик, тестирование и отладка
B. кодирование, техническое проектирование, уточнение функциональных характеристик, сопровождение ПП, тестирование и отладка
C. кодирование, техническое проектирование, уточнение функциональных характеристик, тестирование и отладка
D. определение требований, анализ, реализация и тестирование, внедрение

54. Сұрақ
Вопрос
V – образная модель ЖЦ разработки ПО предполагает:
A. отсутствие временного перекрытия этапов
B. наличие обратной связи
C. возможность сокращения времени разработки ПО
D. возможность увеличения жизненного цикла программного продукта

55. Сұрақ
Вопрос
На втором этапе каскадной модели ЖЦ разработки ПО (Требования ПО) осуществляется…
A. составление концептуальной структуры системы
B. определение функциональности программного компонента
C. составление детальной спецификации архитектуры системы
D. составление набора тестовых данных

56. Сұрақ
Вопрос
Проверка корректности требований при использовании V – образной модели ЖЦ разработки ПО осуществляется…
A. после каждого этапа разработки
B. после разработки всей системы
C. после разработки черновой версии системы
D. после разработки набора тестовых данных
57. Сұрақ
Вопрос

Выберите правильную последовательность этапов жизненного цикла программного продукта:

A. техническое проектирование, сопровождение ПП, сбор и анализ требований заказчика, кодирование, уточнение функциональных характеристик, тестирование и отладка

B. сбор и анализ требований, проектирование системы, кодирование, создание программной документации, сопровождение

C. кодирование, сбор и анализ требований заказчика, техническое проектирование, уточнение функциональных характеристик, сопровождение ПП, тестирование и отладка

D. сбор и анализ требований заказчика, уточнение функциональных характеристик, техническое проектирование, кодирование, тестирование и отладка, сопровождение ПП

58. Сұрақ

Вопрос

Во вспомогательные процессы ЖЦ программного продукта входит:

A. документирование, верификация, аттестация, обеспечение качества, совместная оценка, разрешение проблем, аудит
B. управление, создание инфраструктуры, усовершенствование, обучение
C. разработка, приобретение, поставка, эксплуатация, сопровождение
D. кодирование, тестирование, сопровождение

59. Сұрақ

Вопрос

Метод получения информации при проектировании программного продукта путем анализа материала подразумевает:

A. изучение материала, с которым будет осуществляться работа с использованием данного ПП
B. изучение работы одного из исполнителей с учетом того, что другие исполнители будут выполнять те же действия и операции
C. накопление опыта разработки программного продукта
D. накопление информации в том случае, если были получены противоречивые сведения

60. Сұрақ

Вопрос

Одним из достоинств классического жизненного цикла программного продукта является

A. дает план и временной график по всем этапам проекта
B. в конце всей работы заказчику будут доступны результаты проекта
C. системный анализ каждого элемента программы
D. отсутствие временного перекрытия этапов разработки программного продукта

61. Сұрақ

Вопрос

Итерационная модель ЖЦПО характеризуется следующими основными особенностями:

A. последовательным выполнением входящих в ее состав этапов
B. наличием обратных связей между этапами
C. отсутствием временного перекрытия этапов
D. отсутствием (или определенным ограничением) возврата к предыдущим этапам
E. возможность проведение корректировки после каждого этапа

62. Сұрақ

Вопрос

Программа является совместимой, если…

A. она работает должным образом не только автономно, но и как часть информационной системы;
B. ее качества могут быть продемонстрированы на практике;
C. она допускает быструю модификацию с целью приспособления к изменяющимся условиям функционирования.
D. Нет правильного ответа

63 Сұрақ
Вопрос

В конце каждого витка спирали спиральной модели ЖЦ разработки ПО получаем…

A. готовый программный продукт
B. одну версию программного продукта
C. версию программного продукта с набором тестовых данных
D. черновую модель программного продукта

64. Сұрақ
Вопрос

Спиральная модель ЖЦ разработки ПО предполагает:

A. отсутствие временного перекрытия этапов
B. наличие обратной связи
C. возможность сокращения времени разработки ПО
D. нет правильного ответа

65. Сұрақ
Вопрос

На втором этапе каскадной модели ЖЦ разработки ПО (Требования к ПО) осуществляется…

A. определение функциональности программного компонента
B. составление детальной спецификации архитектуры системы
C. составление концептуальной структуры системы
D. написание программного кода

66. Сұрақ
Вопрос

Какую модель жизненного цикла разработки ПО целесообразнее использовать, если нет четко определенных требований к будущей системе?

A. каскадную
B. спиральную
C. V – образную
D. итерационную

67. Сұрақ
Вопрос

В каких годах разработана основная часть документов единой системы программной документации?

A. 60-70-х гг
B. 70-80-х гг
C. 80-90-х гг
D. 60-90-х гг

68. Сұрақ

Вопрос

На сколько частей можно разделить ЕСПД?

A. 6
B. 8
C. 5
D. 10

69. Сұрақ

Вопрос

Группа «0» ЕСПД указывает на …

A. резервные группы
B. общее положение
C. основополагающие стандарты
D. прочие стандарты

70. Сұрақ

Вопрос

Обозначение стандарта ЕСПД должно состоять из трех частей, где первое число обозначает…

A. присвоение к классу стандартов ЕСПД
B. код классификационной группы стандарта
C. год регистрации стандарта
D. номер документа в реестре

71. Сұрақ

Вопрос

Техническое задание – это

A. набор правил, по которым строится ПП
B. задание, которое необходимо выполнить на ПК по техническим характеристикам
C. набор правил эксплуатации программного продукта
D. совокупность требований к программным средствам, которые могут использоваться как критерий проверки и приемки разработанного ПП

72. Сұрақ

Вопрос

ГОСТ 19.102-77 называется:

A. «Правила внесения изменений в программные документы, выполняемые непечатным способом»
B. «Стадии разработки»
C. «Техническое задание. Требования к содержанию и оформлению»
D. «Основные надписи»

73. Сұрақ

Вопрос

Техническое задание, как одно из стадий разработки, состоит из трех частей.Каких?

A. научно-исследовательская работа, разработка эскизного проекта, разработка технического проекта
B. разработка программной документации, утверждение эскизного и технического проектов, подготовка и передача программы
C. обоснование необходимой разработки программы, научно-исследовательская работа, разработка и утверждение технического задания

D. разработка программной документации, утверждение выбранных методов разработки, утверждение технического проекта, кодирование

74. Сұрақ
Вопрос
Объектный модуль – это…
A. отдельная программа, независима от других выполняемых программ
B. модуль специальной структуры, созданный при компилировании программы
C. обычный текстовый файл с нужным расширением
D. средство языка программирования служащее для увеличения уровня языка программирования

75. Сұрақ
Вопрос
Одной из составных частей рабочего проекта является
A. разработка программ и программной документации
B. разработка технического задания;
C. выбор языка программирования
D. разработка технического проекта

76. Сұрақ
Вопрос
Системное программное обеспечение – это
A. комплекс взаимосвязанных программ для решения задач определенного класса конкретной предметной области
B. совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ
C. совокупность программ и программных комплексов, обеспечивающих технологию разработки, отладки и внедрения создаваемых программных продуктов

77. Сұрақ
Вопрос
Предметная (прикладная) область – это
A. проблема, подлежащая решению
B. совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей
C. программная реализация решения задачи
D. точная формулировка решения задачи на компьютере с описанием входной и выходной информации

78. Сұрақ
Вопрос
Задача – это
A. проблема, подлежащая решению
B. совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей
C. программная реализация решения задачи
D. точная формулировка решения задачи на компьютере с описанием входной и выходной информации

79. Сұрақ
Вопрос

Программное средство - это

A. программа для удовлетворения нужд разработчиков, предназначенная для продажи
B. программа, предназначенная для многократного применения на различных объектах и разработанная любым способом
C. программная реализация решения задачи на компьютере
D. результат разработки какого-либо технического задания

80. Сұрақ
Вопрос

Технологические задачи – это

A. задачи, требующие решения при реализации функций управления в рамках информационных систем предметных областей
B. основа для разработки сервисных средств ПО (утилиты, библиотеки)
C. совокупность связанных между собой функций и задач управления, с помощью которых достигается выполнение поставленных целей
D. задачи, которые ставятся и решаются при организации технологического процесса обработки информации на компьютере

81. Сұрақ
Вопрос

Дайте расшифровку аббревиатуре ЕСПД.

A. Единая схема проектирования документов
B. Единая система программной документации
C. Единая схема программных документов
D. Единственная система программной документации

82. Сұрақ
Вопрос

На современном этапе выделяют 2 основных подхода к проектированию ПП. Какие?

A. структурный и процедурный
B. объектно-ориентированный и структурный
C. метод проектирования Джексона и объектно-ориентированный
D. иерархический и сетевой

83. Сұрақ
Вопрос

Методами структурного проектирования являются

A. модульное программирование, нисходящее проектирование, кодирование и тестирование, структурное проектирование;
B. интегрированное и модульное проектирование;
C. функционально – ориентированное и объектно-ориентированное проектирование
D. структурное программирование, модульное проектирование, тестирование и кодирование

84. Сұрақ
Вопрос

Что не использует структурный подход проектирования программного продукта?

A. диаграммы декомпозиции
B. интегрированную структуру данных предметной области
C. структурные схемы
D. анализ предметной области

85. Сұрақ
Вопрос

Объектно-ориентированный подход проектирования программного продукта основан на:

A. проектировании

B. кодировании и тестировании
C. создании иерархии классов, наследовании свойств объектов и методов их обработки
D. выделении классов объектов

86. Сұрақ
Вопрос
Проектирование – это
A. итерационный процесс, при помощи которого требования к программным средствам транслируются в инженерное представление
B. процесс построения модели будущего программного средства
C. инженерное представление программного продукта на каком-либо языке программирования
D. представление программного продукта совокупностью объектов и их свойств

87. Сұрақ
Вопрос
Дополните фразу: предварительное проектирование программного продукта формирует…
A. уточнение абстракций и добавляет подробности алгоритмического уровня
B. абстракцию архитектурного уровня
C. идентификацию подсистемы и определение основных принципов управления подсистемами
D. набор тестовых данных

88. Сұрақ
Вопрос
Какие модели можно использовать при структурировании системы?
A. модель абстракционной машины, трехуровневую модель, модель хранилища данных, модель клиент-сервер
B. модель событийного управления, модель хранилища данных, модель потока данных, трехуровневую модель
C. модель объекта, модель централизованного управления, модель хранилища данных, модель абстракционной машины
D. модель объекта, модель централизованного управления, модель абстрактной машины

89. Сұрақ
Вопрос
Назовите виды моделей управления.
A. модель потока данных и модель хранилища данных
B. модель клиент-сервер и модель управления прерываниями
C. модель централизованного и событийного управления
D. модель централизованного и периферийного управления

90. Сұрақ
Вопрос
При разбиении программного средства на отдельные модули можно выделить 2 модели:
A. модель потока данных и модель событий
B. модель потока данных и модель объекта
C. модель объекта и модель управления
D. модель управления и модель событий

91. Сұрақ
Вопрос
В основе модели потока данных лежит –
A. сцепление компонентов
B. разделение данных
C. разбиение по функциям

D. выделение отдельных компонентов и их свойств

92. Сұрақ

Вопрос

К классическим методам проектирования ПС ориентированных на процедурную реализацию относят:

A. структурный метод проектирования и метод проектирования Джексона
B. метод проектирования Джексона и объектно-ориентированный метод проектирования
C. объектно-ориентированный метод проектирования и структурный метод проектирования
D. иерархический и структурный методы

93. Сұрақ

Вопрос

Структурный метод проектирования ПП основан на:

A. разбиении единой системы на автономные объекты реального мира
B. выделении объектов и их методов
C. разбиении всей программы на отдельные логические части
D. последовательной декомпозиции всей системы на отдельные компоненты

94. Сұрақ

Вопрос

Исходными данными для структурного метода проектирования ПП являются

A. логические компоненты, составляющие программное средство
B. отдельные компоненты модели анализа программных средств
C. специальные компоненты, имеющие в основе собственный набор данных
D. объекты системы, их свойства и методы

95. Сұрақ

Вопрос

Проектирование для потоков типа преобразования состоит из

A. 3 уровней
B. 5 уровней
C. 4 уровней
D. 7 уровней

96. Сұрақ

Вопрос

Модуль — это…

A. самостоятельная часть программы, имеющая определенное назначение и обеспечивающая заданные функции обработки автономно от других программ
B. упорядоченный набор команд, обеспечивающий выполнение определенных функций
C. алгоритм построения программного продукта
D. нет правильного ответа

97. Сұрақ

Вопрос

Головной модуль –

A. обеспечивает вызов других модулей на обработку
B. управляет запуском программного продукта
C. выполняет функции обработки
D. осуществляет обслуживающие функции

98. Сұрақ

Вопрос

Синтаксическая спецификация модуля программного продукта позволяет

A. построить на используемом языке программирования синтаксически правильное обращение к модулю

B. описать семантику функций, выполняемых этим модулем по каждому из его входов
C. описать древовидную структуру модуля
D. нет правильного ответа

99. Сұрақ

Вопрос

Метод восходящей разработки программного продукта заключается в

A. первоначальном построении модульной структуры в виде дерева затем проектируется каждый модуль в отдельности начиная с нижнего уровня
B. первоначальном построении модульной структуры в виде дерева затем проектируется каждый модуль в отдельности начиная с головного уровня
C. проектировании всех модулей одновременно и последующая их сборка в единую систему
D. нет правильного ответа

100. Сұрақ

Вопрос

Конструктивный подход в разработке ПП основывается на

A. восходящей разработке ПП
B. архитектурной разработке ПП
C. нисходящей разработке ПП
D. является комбинированным способом проектирования

101. Сұрақ

Вопрос

Спецификация программного модуля содержит

A. общие требования к ПП
B. функциональную и синтаксическую спецификацию
C. требования к каждому модулю ПП
D. нет правильного ответа

102. Сұрақ

Вопрос

Для контроля структуры программы можно использовать

A. смежный контроль
B. тестовый контроль
C. итоговый контроль
D. сквозной контроль

103. Сұрақ

Вопрос

В основе структуры программного продукта лежит

A. функциональная организация продукта и его функций
B. модульная организация продукта и его функций
C. пошаговое программирование всех функций продукта
D. нет правильного ответа

104. Сұрақ

Вопрос

Рабочий модуль

A. обеспечивает вызов других модулей на обработку
B. управляет запуском программного продукта
C. выполняют функции обработки
D. осуществляют обслуживающие функции

105. Сұрақ

Вопрос

Модульную структуру ПП можно представить

A. в виде сетевой структуры

B. в виде древовидной структуры
C. в реляционной структуры
D. нет правильного ответа

106. Сұрақ

Вопрос

При нисходящем тестировании первым тестируется

A. управляющий модуль
B. головной модуль
C. рабочий модуль
D. исполняемый модуль

107. Сұрақ

Вопрос

При конструктивном подходе к разработке ПП головной модуль программируется исходя из

A. спецификаций модуля
B. спецификаций программы в целом
C. общих требований к программному продукту
D. нет правильного ответа

108. Сұрақ

Вопрос

В классическом методе нисходящей разработки программного продукта рекомендуется

A. сначала запрограммировать весь программный продукт, а затем начинать нисходящее их тестирование
B. сначала все модули разрабатываемой программы запрограммировать, а затем начинать нисходящее их тестирование
C. сначала запрограммировать весь программный продукт, а затем начинать детально прорабатывать каждый модуль
D. нет правильного ответа

109. Сұрақ

Вопрос

Цели структуризации программного продукта.

A. контролировать трудозатраты и стоимость проектных работ
B. разрозненное выполнение отдельных функций программного продукта
C. распределить работы по исполнителям, обеспечив приемлемую их загрузку и требуемые сроки разработки программных продуктов
D. нет правильного ответа

110. Сұрақ

Вопрос

Управляющий модуль -

A. обеспечивает вызов других модулей на обработку
B. управляет запуском программного продукта
C. выполняют функции обработки
D. осуществляют обслуживающие функции

111. Сұрақ

Вопрос

Функциональная спецификация модуля позволяет

A. построить на используемом языке программирования синтаксически правильное обращение к модулю
B. описать семантику функций, выполняемых этим модулем по каждому из его входов
C. описать древовидную структуру модуля
D. нет правильного ответа

112. Сұрақ
Вопрос
Метод нисходящей разработки программного продукта заключается в
A. проектировании всех модулей одновременно и последующая их сборка в единую систему
B. первоначальном построении модульной структуры в виде дерева затем проектируется каждый модуль в отдельности начиная с нижнего уровня
C. первоначальном построении модульной структуры в виде дерева затем проектируется каждый модуль в отдельности начиная с головного уровня
113. Сұрақ
Вопрос
Архитектурный подход в разработке ПП основывается на
A. восходящей разработке ПП
B. нисходящей разработке ПП
C. является комбинированным способом проектирования
D. нет правильного ответа
114. Сұрақ
Вопрос
В рамках конструктивного подхода сначала реализуются
A. простейшие модули
B. более сложные модули
C. разрабатывается программа целиком
D. нет правильного ответа
115. Сұрақ
Вопрос
Для контроля структуры программы можно использовать
A. смежный контроль
B. тестовый контроль
C. модульный контроль
D. сквозной контроль
116. Сұрақ
Вопрос
Модули бывают
A. главные, вспомогательные, сервисные, управляющие
B. головные, управляющие, рабочие, сервисные
C. главные, побочные, рабочие
D. первичные, вторичные, локальные, глобальные
117. Сұрақ
Вопрос
Сервисный модуль
A. обеспечивает вызов других модулей на обработку
B. управляет запуском программного продукта
C. выполняет функции обработки
D. осуществляет обслуживающие функции
118. Сұрақ
Вопрос
Назовите методы разработки структуры программного продукта.
A. нисходящий, восходящий, конструктивный и архитектурный метод разработки
B. разработка сверху вниз и восходящая разработка
C. нисходящая, восходящая, детальная разработка
D. конструктивна, восходящая, нисходящая разработка
119. Сұрақ

Вопрос

При восходящем тестировании программного продукта для каждого модуля необходимо

A. создавать отдельный набор тестирующих данных и проверять взаимодействие модуля с остальными модулями и головной программой
B. создавать ведущую программу, которая должна подготовить для тестируемого модуля необходимое состояние информационной среды и произвести требуемое обращение к нему
C. создавать дополнительный модуль, который тестирует исходный модуль

120. Сұрақ

Вопрос

Важным при архитектурном подходе проектирования программного продукта является

A. разработка конкретной программы
B. разработка конкретной функции программы
C. повышение уровня используемого языка программирования
D. возможность разработки архитектуры программного продукта после проектирования основных модулей

121. Сұрақ

Вопрос

Достоинством целенаправленной конструктивной реализации является то,

A. что на ранней стадии создается протестированный программный продукт
B. что уже на ранней стадии создается работающий вариант разрабатываемой программы
C. что ПП создается поэтапно
D. нет правильного ответа

122. Сұрақ

Вопрос

Сквозной контроль – это

A. мысленное прокручивание структуры программы при выполнении заранее разработанных тестов
B. контроль со стороны разработчиков архитектуры и внешнего описания ПС
C. проверка каждого действия разработчика
D. проверка структуры программы, набора разработанных тестов и последующее применение тестов к программе

123. Сұрақ

Вопрос

Какую связь устанавливает пользовательский интерфейс?

A. серверную
B. доменную
C. клиент-серверную
D. клиентскую

124. Сұрақ

Вопрос

Выберите правильную характеристику системного программиста.

A. системные программисты знают тонкости построения ПС и могут ее модифицировать
B. системные программисты умеют строить алгоритмы и на основе их реализовывать программу
C. системные программисты способны разрабатывать базовые методы и средства оснащения ПО
D. системные программисты не имеют опыт работы с ПК

125. Сұрақ

Вопрос

Каким базовым требованиям должен отвечать интерфейс, чтобы он был удобным?

A. понятность, эстетичность, устойчивость к неполадкам
B. простота, надежность, предсказуемость, адаптивность, стандартность
C. легкая восприимчивость пользователями, несхожесть с другими интерфейсами
D. предсказуемость, модифицируемость, адаптивность, эффективность

126. Сұрақ

Вопрос

Интерактивный режим работы с программами заключается в том, что…

A. действия пользователей ограничивает задание
B. пользователь вводит команды и получает результат
C. оба варианта правильны
D. нет правильного ответа

127. Сұрақ

Вопрос

Одной из особенностей диалогового режима работы программы является:

A. предназначен для непрограммистов
B. широко применяется в современных ИС
C. программа занимает мало ресурса, в том числе и памяти
D. применяется в MSDOS

128. Сұрақ

Вопрос

Пакетный режим работы программ предназначен для…

A. работы с самой программой
B. коммерческих целей
C. работы системных программистов и администраторов сети
D. нет правильного ответа

129. Сұрақ

Вопрос

Язык программирования – это…

A. некоторая структура, в которой указаны правила синтаксиса и семантики
B. множество текстов некоторого алфавита, удовлетворяющих правилам синтаксиса и семантики
C. шаблон построения программных продуктов, с заранее указанными правилами синтаксиса и семантики
D. совокупность правил описания алгоритма

130. Сұрақ

Вопрос

Алфавит языка программирования включает в себя

A. буквы и специальные знаки
B. зарезервированные слова
C. оба варианта правильны
D. нет правильного ответа

131. Сұрақ

Вопрос

Семантика языка программирования – это…

A. правила, определяющие какие операции, и в какой последовательности должна выполнять ЭВМ
B. правила, определяющие какие операции должна выполнять ЭВМ
C. набор правил, которым должна удовлетворять любая программа
D. описание допустимых элементов, используемых языком программирования

132. Сұрақ

Вопрос

Одной из сравнительных характеристик языка программирования является уровень языка, который характеризуется

A. разнообразием создаваемых программ
B. сложностью задач с использованием данного языка программирования
C. свойствами совокупности понятий, служащих для описания данного языка программирования
D. сложностью тестирования задач, разработанных на данном языке программирования

133. Сұрақ

Вопрос

Гибкость языка программирования обеспечивает –

A. описание задачи с использованием определенной предметной области
B. легкость выражения данном языке программирования, необходимое для решения задачи
C. независимость языка от других программных и аппаратных средств
D. возможность тестирования программы на другом языке программирования

134. Сұрақ

Вопрос

Полнота языка программирования обеспечивает –

A. легкость восприятия пользователем программы
B. легкость понимания семантики языков конструкций
C. простоту разработки программного продукта
D. описание задачи с использованием определенной предметной области

135. Сұрақ

Вопрос

Что такое консольное приложение?

A. это монитор и клавиатура, рассматриваемые как единое устройство
B. устройством вывода
C. программа, предназначенная для работы в операционной системе MS-DOS
D. обычное приложение C++

136. Сұрақ

Вопрос

Модульная программа – это программа, в которой…

A. модульную часть логической структуры можно изменить, не внося изменений в основную часть программы
B. модульную часть программы можно оставить без изменений, а внести изменения в саму программу
C. модульная часть функционирует независимо от программы
D. нет правильного ответа

137. Сұрақ

Вопрос

Модуль – это

A. отдельная единица, которая представляет собой «шаблон» для построения программных продуктов
B. отдельно компилируемая часть программы
C. одна из составляющих программного средства, которая функционирует при подключении к программе
D. именованная часть программы, которая создается с использованием языка программирования и его настроек

138. Сұрақ

Вопрос

Основным свойством модуля является:

A. надежность
B. независимость
C. взаимодействие с программными средствами
D. наличие внешних связей

139. Сұрақ
Вопрос
Логика модуля означает

A. что делает модуль
B. как модуль связан с другими модулями программы
C. наличие внутренних и внешних потоков информации
D. как реализован модуль

140. Сұрақ
Вопрос
Внутренняя связь модуля характеризуется…

A. понятием прочности модуля
B. понятием надежности модуля
C. понятием взаимозависимости модуля
D. логикой модуля

141. Сұрақ
Вопрос
Загрузочный модуль – это…

A. отдельная программа, независима от других выполняемых программ
B. выполняемый *.exe модуль, фактически отдельная программа
C. обычный текстовый файл с нужным расширением
D. средство языка программирования служащее для увеличения уровня языка программирования

142. Сұрақ
Вопрос
Подпрограмма – это…

A. средство языка программирования, служащее для увеличения уровня языка программирования
B. набор откомпилированных программ, собранных в специально форматированный файл
C. практическая часть текса программы, которая подставляется при каждом вызове, увеличивая ее размер
D. обычный текстовый файл с нужным расширением

143. Сұрақ
Вопрос
Одним из источников ошибок в программе может быть

A. перевод программы;
B. взаимопонимание;
C. оба варианта правильны.
D. Нет правильного ответа

144. Сұрақ
Вопрос
Качество ПП - это

A. совокупность свойств этого продукта, которые удовлетворяют определенным потребностям пользователей в соответствии с его назначением;
B. те свойства данного продукта, благодаря которым программный продукт может функционировать в любой программной среде;

C. совокупность свойств программного продукта, которые удовлетворяют требованиям ЕСПД и базовым международным стандартам.
D. Нет правильного ответа

145. Сұрақ
Вопрос
Сложность программы может заключаться в…
A. сложность в построении неформальной модели предметной области;
B. реализации программы, сложность в самой предметной области;
C. сложность в создании загрузочного модуля.
D. Нет правильного ответа

146. Сұрақ
Вопрос
Изучаемость ПП включает в себя:
A. удобочитаемость, тестируемость, информативность;
B. внедряемость, понятность, удобочитаемость;
C. документированность, понятность, удобочитаемость.
D. Нет прасильного ответа
147. Сұрақ
Вопрос
Функциональная пригодность программного продукта включает в себя:
A. точность, защищенность, надежность;
B. эффективность и внедряемость;
C. понятность, стабильность, надежность.
D. Нет прасильного ответа
148. Сұрақ
Вопрос
Программа является надежной, если…
A. выдаваемый результат работы имеет допустимые значения отклонений от аналогичных отклонений;
B. она продолжает свою работу при возникновении сбоев;
C. она при всех одинаково вводимых данных обеспечивает полную повторяемость результата.
D. Нет прасильного ответа
149. Сұрақ
Вопрос
Программа является эффективной, если…
A. она правильно работает при любых допустимых вариантах исходных данных;
B. объем требуемых ресурсов для ее выполнения не превышает допустимой границы;
C. она работает должным образом не только автономно, но и как часть информационной системы.
D. Нет прасильного ответа

150. Сұрақ
Вопрос
Что такое цикл?
A. Оператор выбора
B. Оператор условия
C. Оператор повторений
D. Оператор перехода

151. Сұрақ
Вопрос
Под ошибкой подразумевается
A. место в программе, где искажение проявляется или становится очевидным
B. неправильность, погрешность или неумышленное искажение объекта или процесса
C. место в программе, где возникают условия для появления искажений
D. исправление выявленных искажений в процессе тестирования программы
152. Сұрақ
Вопрос
Источником ошибок в программе может быть
A. недостаточная квалификация специалиста
B. сложность программы
C. большой объем программы
D. недостаточное знание заказчиком предметной области
153. Сұрақ
Вопрос
Структурный подход к разработке программы является методом борьбы с…
A. переводом программы
B. неквалифицированностью специалиста
C. взаимопониманием
D. сложностью программы
154. Сұрақ
Вопрос
Одним из признаков классификации ошибок является
A. уровень сложности и устойчивости
B. степень зацикливания
C. правильность описания программы
D. возможность описания программы
155. Сұрақ
Вопрос
Процесс отладки включает следующие подпроцессы:
A. выявление ошибок, диагностика и локализация ошибок, исправление ошибок
B. выявление ошибок и их локализация
C. диагностика ошибок, исправление ошибок и повторное тестирование программы
D. выявление ошибки, исправление ошибки
156. Сұрақ
Вопрос
Отладка начинается с того момента как
A. не выдается сообщение об ошибках
B. не выдается сообщения о синтаксических ошибках
C. программа полностью описана
D. прописаны отдельные модули программы
157. Сұрақ
Вопрос
Точка обнаружения – это…
A. место в программе, где ошибка себя проявляет или становится очевидной
B. неправильность, погрешность или неумышленное искажение объекта или процесса
C. место в программе, где ошибку можно локализовать
D. место в программе, где возникают условия для появления ошибки
158. Сұрақ
Вопрос
Что может являться источником ошибки в программе?

A. перевод программы
B. недостаточная квалификация специалиста
C. модульное программирование
D. объектно-ориентированное программирование

159. Сұрақ

Вопрос

Контроль (проверка, испытания) программы является методом борьбы с…

A. переводом программы
B. взаимопониманием
C. сложностью программы
D. описанием программы

160. Сұрақ

Вопрос

Выделяют следующие виды ошибок программ:

A. синтаксические, семантические, первичные
B. ошибки анализа, общего и физического характера
C. ошибки анализа, первичные и вторичные
D. ошибки описания, определения функций и кодирования

161. Сұрақ

Вопрос

Под отладкой понимается процесс

A. нахождения и исправления ошибок
B. позволяющий получить программу, которая функционирует с требующими характеристиками
C. оптимизации программы
D. тиражирования программы

162**.** Сұрақ

Вопрос

Для тестирования программы используют

A. простые тестовые данные
B. просчитанные данные
C. сложные данные
D. произвольные данные

163. Сұрақ

Вопрос

Точка происхождения – это…

A. место в программе, где ошибка себя проявляет или становится очевидной
B. неправильность, погрешность или неумышленное искажение объекта или процесса
C. место в программе, где возникают условия для появления ошибки
D. место в программе, где ошибку можно локализовать

164. Сұрақ

Вопрос

Выберите возможные источники ошибки в программе.

A. модульное программирование
B. трудность во взаимопонимании между заказчиком и разработчиком
C. сложность понимания языка программирования
D. объектное описание программы

165. Сұрақ

Вопрос

Переход на формальные стороны взаимодействия является методом борьбы с…

A. переводом программы
B. взаимопониманием

C. сложностью программы
D. пониманием языка программирования

166. Сұрақ
Вопрос
Одним из признаков классификации ошибок является
A. синтаксис и семантика
B. степень зацикливания
C. первичные и побочные ошибки
D. первостепенные и второстепенные ошибки
167. Сұрақ
Вопрос
Отладка бывает:
A. ручная и семантическая
B. ручная и автоматизированная
C. разрушающая и неразрушающая
D. разрушающая, семантическая, оптимизирующая
168. Сұрақ
Вопрос
Тестирование – это…
A. оптимизация программ
B. действие, направленное на выявление ошибок
C. регистрация программы
D. исправление выявленных ошибок
169. Сұрақ
Вопрос
Тестирование – это…
A. процесс создания загрузочного файла программы
B. запуск программы на выполнение
C. процесс многократного выполнения программы с целью обнаружения максимального количества ошибок
D. процесс нахождения и исправления ошибок
170. Сұрақ
Вопрос
Тестовый набор данных должен включать
A. входные, промежуточные и выходные данные
B. входные и выходные данные
C. все промежуточные результаты проверки тестов и конечный результат выполнения каждой функции
D. входные, выходные данные и результаты проверки каждого условия
171. Сұрақ
Вопрос
Тестирование бывает
A. нисходящее, восходящее, промежуточное, завершенное
B. структурное, функциональное, промежуточное, полное
C. нисходящее, восходящее, структурное, полное
D. нисходящее, восходящее, структурное, функциональное
172. Сұрақ
Вопрос
Что известно при тестировании «черного ящика»?
A. функции программы
B. внутренняя структура программы

C. работа каждой функции на всей области определения
D. внутренние элементы программы и связи между ними

173. Сұрақ

Вопрос

При тестировании «белого ящика» исследуется…

A. функции программы
B. внутренняя структура программы
C. работа каждой функции на всей области определения
D. внутренние элементы программы и связи между ними

174. Сұрақ

Вопрос

К методам «белого ящика» относятся…

A. метод покрытия решений, метод граничных решений, метод функциональных диаграмм, метод покрытия условий
B. метод эквивалентных разбиений, метод функциональных диаграмм, анализ граничных решений
C. метод покрытия условий, метод покрытия операторов, метод покрытия решений, анализ граничных решений
D. метод покрытия условий, метод покрытия операторов, метод покрытия решений, метод покрытия решений и условий

175. Сұрақ

Вопрос

Метод эквивалентных разбиений основан на…

A. разработке такого числа эквивалентных тестов, достаточного для того, что бы все возможные результаты каждого условия в решении выполнялись по крайне мере один раз
B. разбиении входной области программы на классы по определенным признакам
C. разработке достаточного количества тестов, чтобы каждое решение на этих тестах выполнялось по крайне мере один раз
D. выполнении каждого оператора хотя бы один раз

176. Сұрақ

Вопрос

Метод покрытия условий основан на…

A. разработке такого числа эквивалентных тестов, достаточного для того, что бы все возможные результаты каждого условия в решении выполнялись по крайне мере один раз
B. разбиении входной области программы на классы по определенным признакам
C. разработке достаточного количества тестов, чтобы возможные результаты каждого условия в решении выполнялось по крайне мере один раз
D. выполнении каждого оператора хотя бы один раз

177. Сұрақ

Вопрос

Тестирование «черного ящика» выполняется

A. на ранних этапах разработки программы
B. когда разработан весь программный продукт и протестированы отдельные его модули
C. на поздних стадиях тестирования программы
D. на ранних стадиях тестирования программы

178. Сұрақ

Вопрос

Техника «черного ящика» ориентирована на…

A. выявление класса ошибок
B. выявление отдельных ошибок

C. сокращение количества тестовых вариантов
D. увеличение количества тестовых наборов

179. Сұрақ

Вопрос

Тестирование включает в себя …

A. создание текстового, загрузочного файла и их проверка
B. разработка тестов и непосредственное тестирование по ним
C. проверка разработанного набора тестов на исполняемом файле
D. составление алгоритма решения задачи, текста программы, набора тестовых данных и их проверка

180. Сұрақ

Вопрос

Чему равна вероятность наличия необнаруженных ошибок в какой-то части программы?

A. обратно пропорциональна числу ошибок обнаруженных в программе
B. количеству обнаруженных в программе ошибок
C. пропорциональна числу ошибок обнаруженных в программе
D. 1/3 числу обнаруженных ошибок

181. Сұрақ

Вопрос

Что известно при тестировании «белого ящика»?

A. функции программы
B. внутренняя структура программы
C. работа каждой функции на всей области определения
D. внутренние элементы программы и связи между ними

182. Сұрақ

Вопрос

При тестировании «черного ящика» исследуется…

A. функции программы
B. внутренняя структура программы
C. работа каждой функции на всей области определения
D. внутренние элементы программы и связи между ними

183. Сұрақ

Вопрос

К методам «черного ящика» относятся…

A. метод покрытия решений, метод граничных решений, метод функциональных диаграмм, метод покрытия условий
B. метод эквивалентных разбиений, метод функциональных диаграмм, анализ граничных решений
C. метод покрытия условий, метод покрытия операторов, метод покрытия решений, анализ граничных решений
D. метод покрытия условий, метод покрытия операторов, метод покрытия решений, метод покрытия решений и условий

184. Сұрақ

Вопрос

Метод покрытия операторов при тестировании программ основан на…

A. разработке такого числа эквивалентных тестов, достаточного для того, что бы все возможные результаты каждого условия в решении выполнялись по крайне мере один раз
B. разбиении входной области программы на классы по определенным признакам
C. разработке достаточного количества тестов, чтобы каждое решение на этих тестах выполнялось по крайне мере один раз

D. выполнении каждого оператора хотя бы один раз

185. Сұрақ

Вопрос

Граничные условия – это

A. условия, ситуация, возникающая непосредственно на границе выше или ниже границ входных или выходных элементов класса эквивалентности
B. ситуация, возникающая непосредственно на промежуточных элементах класса эквивалентности
C. условия, ситуация, возникающие внутри программы, когда выполнены все тестовые наборы

186. Сұрақ

Вопрос

При тестировании программ методами «черного ящика» необходимо разрабатывать набор тестов, который…

A. показывает нормальное функционирование программы
B. выявляет все ошибки программы и по ним позволяет оптимизировать программу
C. показывает нормальное и аномальное функционирование программы
D. нет правильного ответа

187. Сұрақ

Вопрос

Тестирование «белого ящика» выполняется

A. на ранних этапах разработки программы
B. когда разработан весь программный продукт и протестированы отдельные его модули
C. на поздних стадиях тестирования программы
D. на ранних стадиях тестирования программы

188. Сұрақ

Вопрос

Тестирование «черного ящика» обеспечивает поиск следующих категорий ошибок:

A. ошибок во внутренних структурах данных
B. ошибок интерфейса
C. ошибок во внешних структурах данных
D. ошибок в циклах и ветвлениях
E. ошибок характеристик

189. Сұрақ

Вопрос

К программным средствам защиты программного продукта не относят….

A. криптографическую защиту
B. ограничение доступа к программному продукту
C. патентную защиту
D. нестандартное форматирование диска, на котором находится программный продукт

190. Сұрақ

Вопрос

Лицензирование программного продукта относится к…

A. правовой защите ПП
B. программной защите ПП
C. технической защите ПП
D. физической защите ПП

191. Сұрақ

Вопрос

Каким знаком обозначается авторское право на программный продукт?

A. ©
B. ™

C. ®
D. X
192. Сұрақ
Вопрос
Каким знаком обозначается регистрация права на программный продукт?
A. ©
B. ™
C. ®
D. X
193. Сұрақ
Вопрос
Этап Эволюции при сопровождении программного продукта предполагает…
A. выявление и устранение обнаруженных ошибок, тиражирование, контроль за распространением версии, введение новых функций программы и т.д
B. внесение изменения в программу в ответ на изменившиеся условия
C. использование всех возможных и невозможных способов для поддержания жизни в старой и распадающейся на части программной системе
D. проектирование программного продукта, тестирование, тиражирование и утилизацию
194. Сұрақ
Вопрос
Этап Сохранение при сопровождении программного продукта предполагает…
A. выявление и устранение обнаруженных ошибок, тиражирование, контроль за распространением версии, введение новых функций программы и т.д
B. внесение изменения в программу в ответ на изменившиеся условия
C. использование всех возможных и невозможных способов для поддержания жизни в старой и распадающейся на части программной системе
D. проектирование программного продукта, тестирование, тиражирование и утилизацию
195. Сұрақ
Вопрос
Этап Чистое сопровождение при сопровождении программного продукта предполагает…
A. выявление и устранение обнаруженных ошибок, тиражирование, контроль за распространением версии, введение новых функций программы и т.д
B. внесение изменения в программу в ответ на изменившиеся условия
C. использование всех возможных и невозможных способов для поддержания жизни в старой и распадающейся на части программной системе
D. проектирование программного продукта, тестирование, тиражирование и утилизацию
196. Сұрақ
Вопрос
Существует две основные модели организации коллектива при разработке ПО:
A. иерархическая модель и модель группы
B. структурная и объектная модель
C. иерархическая и объектная модель
D. модель группы и сетевая модель
197. Сұрақ
Вопрос
Какая модель коллективной разработки программного продукта определяет структуру коллектива с точки зрения отдела кадров?
A. модель группы
B. иерархическая модель
C. структурная модель
D. сетевая модель

198. Сұрақ
Вопрос
Какая модель коллективной разработки программного продукта не определяет структуру коллектива с точки зрения отдела кадров?
A. модель группы
B. иерархическая модель
C. структурная модель
D. сетевая модель
199. Сұрақ
Вопрос
Кто из членов группы при коллективной разработке программных продуктов участвует в создании пользовательского интерфейса, сокращая тем самым затраты на сопровождение продукта и поддержку пользователей?
A. менеджер продукта
B. разработчик
C. инструктор
D. тестер
200. Сұрақ
Вопрос
Какие задачи необходимо решить, чтобы проект считался удачным?
A. удовлетворить требования заказчика
B. соблюсти ограничения
C. спроектировать систему по объектно-ориентированному методу
D. выполнить спецификации, основанные на требованиях пользователей
E. выпустить продукт только после выявления и устранения всех проблем
F. выполнить программный продукт с учетом ситуации на рынке программ
G. гарантировать простоту развертывания и управления

1. C	2. A	3. C	4. B	5. B	6. C
7. A	8. A	9. C	10. C	11. B	12. B
13. A	14. C	15. B	16. C	17. D	18. B
19. D	20. D	21. B	22. B	23. C	24. D
25. D	26. D	27. B	28. C	29. D	30. D
31. B	32. D	33. A	34. B	35. A	36. D
37. C	38. c	39. C	40. A	41. D	42. A
43. C	44. C	45. B	46. A	47. C	48. A
49. B	50. B	51. C	52. A	53. D	54. A
55. B	56. D	57. D	58. A	59. A	60. B
61. E	62. A	63. D	64. C	65. B	66. C
67. B	68. A	69. D	70. A	71. D	72. C
73. B	74. A	75. B	76. B	77. A	78. D
79. C	80. D	81. B	82. A	83. D	84. B
85. C	86. C	87. C	88. B	89. B	90. B
91. D	92. A	93. C	94. A	95. D	96. A
97. A	98. A	99. B	100.C	101.C	102.B
103.A	104.A	105.B	106.C	107.A	108.B
109.C	110.B	111.B	112.B	113.B	114.A
115.B	116.B	117.D	118.A	119.B	120.D
121.A	122.D	123.D	124.C	125.A	126.B
127.B	128.C	129.C	130.C	131.A	132.C
133.C	134.B	135.C	136.A	137.C	138.C
139.B	140.D	141.B	142.C	143.A	144.C
145.A	146.C	147.A	148.C	149.A	150.C
151.A	152.A	153.D	154.B	155.A	156.D
157.D	158.B	159.C	160.A	161.A	162.B
163.C	164.B	165.B	166.A	167.A	168.B
169.D	170.D	171.D	172.B	173.C	174.D
175.A	176.A	177.B	178.B	179.B	180.C
181.A	182.D	183.B	184.D	185.A	186.B
187.B	188.A	189.A	190.A	191.C	192.A
193.B	194.B	195.A	196.A	197.A	198.B
199.D	200.D				