

Н. С. Кольева, Е. В. Шевчук

ИНФОРМАТИКА

Жалпы білім беретін мектептің
жаратылыстану-математика бағытындағы
10-сыныбына арналған оқулық

*Қазақстан Республикасының Білім және
ғылым министрлігі бекіткен*



Алматы «Мектеп» 2019

ӘОЖ 373.167.1
КБЖ 32.973я72
К51

Кольева Н.С., Шевчук Е.В.
К51 **Информатика. Жалпы білім беретін мектептің жаратылыстану-математика бағытындағы 10-сыныбына арналған оқулық. — Алматы: Мектеп, 2019. — 184 б.**

ISBN 978—601—07—1103—7

К $\frac{4306020500—101}{404(05)—19}$ 43(1)—19

ӘОЖ 373.167.1
КБЖ 32.973я72

© Кольева Н.С., Шевчук Е.В., 2019
© «Мектеп» баспасы,
көркем безендірілуі, 2019

Барлық құқықтары қорғалған

Басылманың мүлкітік құқықтары
«Мектеп» баспасына тиесілі

ISBN 978—601—07—1103—7

КІРІСПЕ

Қазіргі қоғам ақпарат көлемінің қарқынды өсуімен сипатталатын кезеңде өмір сүруде. Заманауи компьютерлік технологияны пайдалану адамдардың ақпаратты өңдеу жылдамдығынан миллион есе жылдам орындайтын автоматты немесе автоматтандырылған құрылғыларды қолданып, аса күрделі амалдарды орындауға мүмкіндік береді.

Басқаруда батыл қадамдар жасайтын бүгінгі маман жоғары ақпараттанған қоғамның жаңа шарттарымен жұмыс істеуге дайын болуы керек. Ол информатика саласын, компьютерлік техниканы қолданудың практикалық дағдысын, қазіргі ақпараттық технологияларды, ақпараттық жүйелермен жұмыс істеудің негізгі принциптерін, ақпараттың толықтығын және дәлдігін бағалай білуі керек. Осы мақсатта «Информатика» оқулығы өз көмегін тигізеді деген сенімдеміз.

Оқулықта теорияны жақсы меңгеру үшін шартты белгілер берілген.

Информатика пәнін меңгеруде сәттілік тілейміз!

Авторлар

ШАРТТЫ БЕЛГІЛЕР:



— мұны білулерің керек



— өзін-өзі тексеруге арналған сұрақтар



— анықтама



— бақылау сұрақтары мен тапсырмалары



— компьютерде орындалатын практикалық жұмыстар

А ДЕҢГЕЙІ

— оқу-танымдық тапсырмалар

В ДЕҢГЕЙІ

— танымдық-зерттеу тапсырмалары

С ДЕҢГЕЙІ

— шығармашылық тапсырмалар

КОМПЬЮТЕРЛІК ЖЕЛІЛЕР ЖӘНЕ АҚПАРАТТЫҚ ҚАУІПСІЗДІК

I

ТАРАУ

Бұл тараудан білесіңдер:

- ▶ компьютерлік желілердің жұмыс жасау принциптерін;
- ▶ ақпараттық қауіпсіздік негіздерін;
- ▶ ақпаратты қорғау әдістерін;
- ▶ идентификациялау әдістерін;
- ▶ сандарды бір санау жүйесінен екінші санау жүйесіне аударуды;
- ▶ логикалық операцияларын (дизъюнкция, конъюнкция, инверсия);
- ▶ компьютердің логикалық элементтерін;
- ▶ мәтінді ақпаратты кодтау принциптерін.

Мынаны үйренесіңдер:

- ▶ желілік компоненттердің мақсатын сипаттауды (түйіндер, маршрутизаторлар, ажыратқыштар);
- ▶ IP-адресінің мақсаты мен көрсетілуін;
- ▶ домендік атаулар жүйесінің (DNS) мақсатын;
- ▶ «Ақпараттық қауіпсіздік», «құпиялылық», «тұтастық» және «қолжетімділік» терминдерінің мағынасын.

§1

Компьютерлік желілердің жұмыс жасау принциптері. Желі компоненттері

Бүгінгі сабақта:

- ▶ желілік компоненттердің мақсатын сипаттауды (түйіндер, маршрутизаторлар, ажыратқыштар);
- ▶ IP-адресінің мақсаты мен көрсетілуін түсіндіруді;
- ▶ домендік атаулар жүйесінің (DNS) мақсатын үйренесіндер.

Тірек ұғымдар:

- ▶ компьютерлік желілер;
- ▶ қосқыштар (коммутаторлар);
- ▶ маршрутизатор, DNS;
- ▶ торап.



Компьютерлік желілер не үшін керек? Ақпаратты бір компьютерден екіншісіне қалай көшіруге болады? Бірнеше тәсілін атаңдар.

Бұрын ақпаратты бір компьютерден екіншісіне ауыстыру үшін олар сыртқы жады құрылғылары — карталар мен жазылған карталарды, содан кейін дискілер, магниттік таспалар және лазерлік дискілерді пайдаланды. Енді деректермен алмасу компьютерлік желілер арқылы жүзеге асырылады.



Компьютерлік желілер — бұл байланыс желілері қосылған компьютерлер тобы.

Желінің көмегімен файлдарды бір компьютерден екіншісіне жылдам көшіруге болады. Барлық мұғалімдер мектеп туралы, оның ішінде электрондық сынып журналдары туралы ақпаратты сақтайтын «Күнделік» электрондық журналымен жұмыс істейді.

Жергілікті желі жасау үшін барлық кабельдерді жүргізіп, аппараттық және желілік бағдарламаларды конфигурациялау қажет. Ақпарат алмасу мен деректерді қорғау жылдамдығы желі параметрлері мен техникалық қызмет көрсету сапасына байланысты. Мұндай мәселемен желі администраторы айналысады.

Қазіргі LAN кабельдік желілерде Ethernet (латынша *aether* — эфир) деп аталатын пакеттік деректер технологиясын пайдаланады. Компьютерлерді бір-бірімен қосу үшін электр кабельдері немесе талшықты-оптикалық кабельдер қолданылады. Желілік карта (желі адаптері, ағылшынша *network interface card*) болса, компьютерді желіге қосуға болады. Заманауи аналық тақта және ноутбуктарда, әдетте, желі картасы бар болады.

Алыс қашықтыққа деректерді беру үшін деректер жарық сәулемен берілетін талшықты-оптикалық кабельдер қолданылады. Жарық шыны немесе пластикалық жарық бағыттаушы цилиндр қабырғаларын көрсететін кабельдің ішіне кіреді.

Коммутаторлар практикада жиі қолданатын «жұлдызша» сызбасына сәйкес компьютерлерді біртұтас желіге біріктіру үшін пайдаланылады (1.1-сурет).



1.1-сурет. Коммутатор

Коммутаторлардың концентраторлардан айырмашылығы ол пакеттік тек қажетті түйінге ғана жібереді. Компьютер коммутаторға RJ-45 қосқышы бар «кабельдік сым» (ағылшынша patching cord — жалғастырушы сым) деп аталатын кабельмен қосылады (1.2-сурет).

Жергілікті желіге қосылған барлық компьютерлер бір байланыс арнасы арқылы Интернетке қосылады. Жергілікті желі Интернетке қосылу үшін маршрутизатор немесе *роутер* қажет. *Маршрутизатордың міндеті* — пакеттің қосымша бағытын анықтау және оны қажетті портқа жіберу. Ол үшін қажетті мекенжайға байланысты пакеттерді қайда жіберу керектігі жазылған бағыт кестесі қолданылады. Жергілікті желідегі маршрутизатордың рөлін бірнеше желілік картасы бар компьютер орындауы мүмкін.

Сымсыз маршрутизатор компьютерлерді сымсыз желіге қосу және Интернетке кіруді қамтамасыз ету үшін пайдаланылады (1.3-сурет).



1.2-сурет. RJ-45
жалғағышы



1.3-сурет. Сымсыз
маршрутизатор

БАҚЫЛАУ СҰРАҚТАРЫ



1. Жергілікті желі дегеніміз не?
2. Сымсыз желіні құру үшін қандай жабдық қажет?
3. Кабельдік желілер үшін қандай жабдық қажет?
4. Коммутатор деген не?
5. Маршрутизатор дегеніміз не? Жергілікті желіде ол қандай рөл атқарады?
6. Компьютерлерді желіге қосудың қандай артықшылықтары мен кемшіліктері бар?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

«Серверлік операциялық жүйелер» тақырыбына қысқаша мәлімет дайындаңдар.

В ДЕҢГЕЙІ

«Блогер болғың келе ме? Неге?» деген тақырыпқа эссе жазыңдар.

С ДЕҢГЕЙІ

Хабарлама (хат) жазып, сыныпта бір-біріңе жергілікті желі арқылы жіберіңдер.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§2

Компьютерлік желілердің жұмыс істеу принциптері. IP-адрес

Бүгінгі сабақта:

- IP-адресінің тағайындалуы мен көрсетілуін түсіндіруді үйренесіңдер.

Тірек ұғымдар:

- компьютерлік желі;
- коммутаторлар;
- маршрутизатор;
- IP-адрес.

Интернетке қосылған барлық компьютерлер бір-бірін автоматты режимде іздеп табады. Хабарламаны жіберу процесіне адамдар мүлде қатыспайды, ол әрбір компьютердің (хост немесе торап) IP-адрес деп аталатын мекенжайына байланысты.



IP-адрес — компьютердің Интернетте орналасқан орнын (мекенжайын) нақты анықтайтын жазба. IP-адрес бір-бірінен нүктемен бөлінген 0-ден 255-ке дейінгі аралықтағы төрт санның жазбасы, мысалы: 192.168.0.13.

Бұл сандарда желі нөмірі мен желідегі компьютер нөмірі кодталған. Екі бөлікті IP-адресінен бөліп алу үшін маска-үлгілері қолданылады. Маска — бұл [0; 255] аралығындағы төрт сан. Маска екілік кодта «N бірлік, содан кейін нөлдер» принципі бойынша жазылады. Мысалы, маска 255.255.255.0 екілік кодта келесі түрде жазылады:

11111111.11111111.11111111.00000000

Алдымен 24 бірлік, содан кейін нөлдер жазылады. Бұл адресінің алғашқы 24 биті — желі нөмірі (192.168.0), ал қалған 8 бит — желідегі компьютер (түйін) нөмірі (13).

Сондай-ақ бұл жазбаны басқаша жазуға болады («/24» — маскада 24 бірлік бар екенін білдіреді).

Мұндай желіде 256 емес 254 түйін болуы мүмкін. Кіші мекенжайы (192.168.0.0) бүкіл желіні белгілеу үшін пайдаланылады, ал үлкені (192.168.0.13) — кеңінен тарату үшін (хабарлама осы желінің барлық компьютерлеріне жіберіледі) қолданылады.

Тораптың мекенжайына үш бит беріледі (маскада үш нөл), бұл желіде тек $2^3 = 8$ мекенжай ғана болады. Олардың екеуі арнайы биттер (желі нөмірлері және кең таратылатын мекенжай), желіге 6 түйінге дейін кіре алады.

Компьютерге ішкі желі маскасы ішкі желі шегарасын, яғни сол желіде кімнің бар екенін, кімнің желіден тысқары екенін анықтау үшін қажет.

Бір желі ішінде компьютерлер ақпаратпен «тікелей», ал басқа желіге жіберу үшін шлюзден шлюзге жіберу арқылы алмасады.

Ішкі желі шегараларын анықтау үшін компьютер жүйеде IP-адрес пен маска арасында көбейтуді орындап, нәтижесінде нөлденген битпен және нөлденген позициялар мекенжайын алады.

215.17.125.177 мекенжайы және 255.255.255.240 маскасы үшін желі нөмірін анықтайық.

IP-адрес: 215.17.125.177 (11010111.00010001.01111101.10110001).

Маска: 255.255.255.240 (11111111.11111111.11111111.11110000).

Биттік көбейтуді қолданып 11010111.00010001.01111101.10110000 аламыз.

Екілік жүйеден ондық жүйеге ауыстырып, желі адресін (мекенжайын) аламыз: 215.17.125.176.

Компьютер нөмірін анықтау үшін IP-адрес және ішкі желі маска байтын (бетпердесін) екілік жүйеге аудару қажет.

IP-адрестегі компьютер нөмірі нөлдік маскада жазылған. Берілген мысалда 0001 екілік жүйеде көрсетілген. Екілік жүйеден ондық жүйеге ауыстырсақ 1 санын аламыз.



IP-адрес компьютерге емес, интерфейс — деректерді беру арнасына (желілік картаға, модемге) беріледі. Сондықтан бір компьютердің бірнеше IP-адресі болуы мүмкін (мысалы, онда екі желілік карта орнатылған болса).

Интернет мекенжайларының қарқынды дамуына байланысты IP хаттамасының жаңа (алтыншы) нұсқасы әзірленді, ол IPv6 деп белгіленеді. Онда әрбір мекенжайға қазіргідей 32 емес, 128 бит беріледі.

Қазіргі уақытта IPv6 қолданылатын 4400-ден астам желі бар.

Үлкен шығынды және барлық ескірген құрылғыларды ауыстыруды талап ететіндіктен IPv6-ға толық көшу жүзеге асырылады.

БАҚЫЛАУ СҰРАҚТАРЫ



1. IP-адрес дегеніміз не?
2. Домендік атауларды құру принципін сипаттаңдар.
3. IP-адресі жадыда қанша орын алады?
4. IP-адресі үшін маска дегеніміз не? Ол қалай құрылады?
5. Компьютердің бірнеше IP-адресі болуы мүмкін бе? Қандай жағдайларда болады?
6. IPv6 хаттамаға көшу неге қажет?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Берілген желінің IP-адресі мен маскасы бойынша желі мекенжайын анықтаңдар:

IP-адрес:	Маска:
а) 12.16.196.10	255.255.224.0
ә) 145.92.137.88	255.255.240.0
б) 217.16.246.2	255.255.252.0
в) 146.212.200.55	255.255.240.0
г) 148.8.238.3	255.255.248.0

В ДЕҢГЕЙІ

Берілген желінің IP-адресі және маскасы бойынша компьютердің желідегі нөмірін анықтаңдар.

IP-адрес:	Маска:
а) 162.198.0.157	255.255.255.224
ә) 156.128.0.227	255.255.255.248
б) 192.168.156.235	255.255.255.240
в) 10.18.134.220	255.255.255.192
г) 122.191.12.220	255.255.255.128
ғ) 156.132.15.138	255.255.252.0
д) 112.154.133.208	255.255.248.0

С ДЕҢГЕЙІ

Әрбір келтірілген мекенжай үшін желі нөмірін, торап нөмірін, желідегі компьютерлердің ең көп санын анықтаңдар:

- а) 192.168.104.109/30
- ә) 172.16.12.12/29
- б) 193.25.5.136/28
- в) 10.10.40.15/27

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§3

Компьютерлік желілердің жұмыс істеу принциптері. DNS

Бүгінгі сабақта:

- ▶ домендік атаулар жүйесінің мақсатын түсіндіруді үйренесіңдер.

Тірек ұғымдар:

- ▶ компьютерлік желі;
- ▶ коммутаторлар;
- ▶ маршрутизатор;
- ▶ IP-адрес;
- ▶ DNS.

DNS домендік атаулар. IP-адрес компьютерлерді Интернетте атау үшін ыңғайлы, бірақ пайдаланушылық жұмыс үшін қолайсыз (көрнекі емес, қиын есте сақталады, енгізу кезінде қате жіберу мүмкін). Сондықтан сандық IP-адрес орнына DNS деп аталатын домендік атаудың әріптік жүйесі қолданылады. Бұл жүйеге сәйкес әрбір Web-сервер атауы нүкте арқылы бөлінген сөздер тізбегінен тұрады, ол есте жеңіл сақталады.

Домендік атау Интернетте серверді анықтайды және келесі иерархиялық принцип бойынша құрылады:

— ең жоғары деңгейде (жоғары деңгейдегі домен), әдетте, елдің атауы орналасқан, мысалы, kz (Қазақстан), ru (Ресей), uk (Ұлыбритания). Бірақ көп жағдайда ел атауының орнына домен тиесілі ұйымның түріне жауап беретін қысқарту қойылады: com (коммерциялық), gov (үкіметтік), mil (әскери), edu (білім беру), net (желілік), org (басқа ұйымдар). Демек,

— жоғары деңгейлі доменнің сол жағында нүкте арқылы қаланың, штаттың немесе ұйымның белгісі жазылады;

— қаланың (ұйымның) атауының сол жағына нүктеден кейін домен атауында сол жақ позицияны иеленетін сервер атауы жазылады.

Нәтижесінде сервердің домендік атауы (басқаша айтқанда, домен) келесі түрде беріледі:

Beeline.kz — ұялы байланыс операторының сервері.

IP-адресі мен домендік атаулар арасындағы сәйкестік арнайы DNS серверлердегі деректер қоры арқылы орнатылады. DNS серверлері — домендік атау жүйелерінің жұмыс істеуі үшін қажетті күнделікті жұмысты орындайтын электронды роботтар.

Браузердің адрестік жолына сайт мекенжайын (домендік атауын) енгізгенде, алдымен DNS-серверге сұраныс жіберіледі, оның мақсаты — сервердің IP-адресін анықтау. Егер бұл мүмкін болса, веб-бетті

алуға сұраныс жіберіледі және IP хаттама драйвері домен атауын емес, алынған IP-адресі пайдаланады.

Бір домендік атау бірнеше IP-адресіне сәйкес келуі мүмкін. Мұндай әдіс оқырмандары көп сайттарға жүктемені бөлу үшін қолданылады. Осылайша домендік атаулар мен IP-адресстердің арасындағы сәйкестікті «көптің көпке» деп сипаттауға болады. Демек, бір IP-адреске бірнеше домендік атаулар және керісінше байланысты болуы мүмкін.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Домендік атауларды құру принципін сипаттаңдар.
2. DNS сервері дегеніміз не?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Атаулардың түбірлік серверлерінің болжамды географиялық орналасуын анықтаңдар (бұл үшін whois қызметін пайдаланыңдар).

В ДЕҢГЕЙІ

DNS серверін орнатыңдар (forward сервер ретінде 192.168.128.1 немесе 192.168.128.5 DNS серверін пайдаланыңдар).

С ДЕҢГЕЙІ

Бапталған сервистің жұмыс қабілеттілігін тексеріңдер. Wireshark снифері арқылы nslookup утилитасының жұмыс істеу механизмін зерттеңдер.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 4

Компьютерлік желілердің жұмыс істеу принциптері. Жеке виртуалды желі

Бүгінгі сабақта:

- ▶ домен атаулары жүйесінің мақсатын түсіндіруді;
- ▶ жеке виртуалды желінің мақсатын түсіндіруді үйренесіңдер.

Тірек ұғымдар:

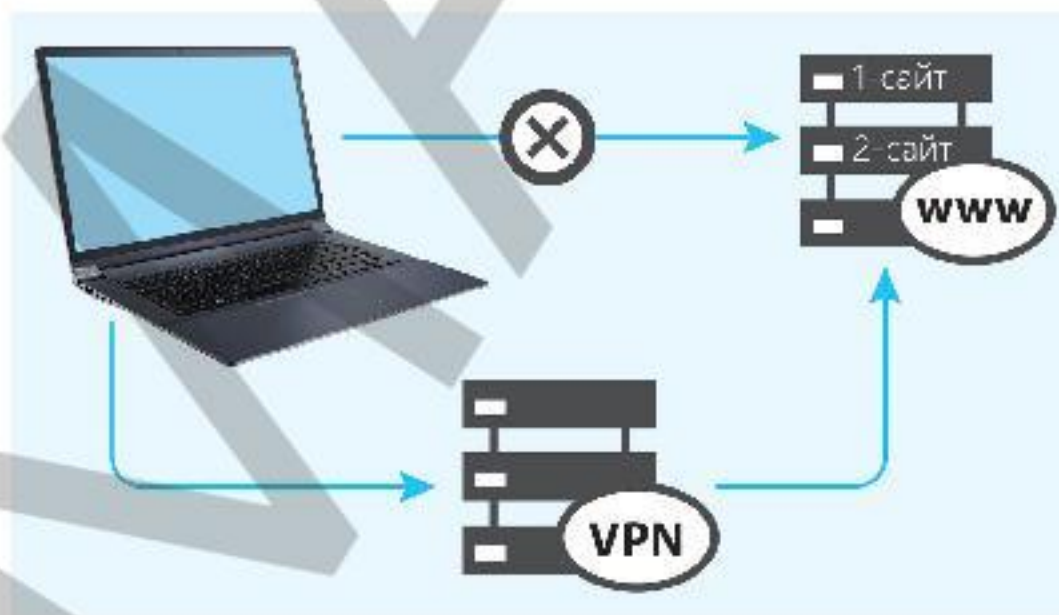
- ▶ компьютерлік желі;
- ▶ коммутаторлар;
- ▶ маршрутизатор;
- ▶ IP-адресі;
- ▶ DNS;
- ▶ жеке виртуалды желі.

Виртуалды жеке желі (VPN) — ұйымдар мен компанияларға қолданыстағы жалпы желі, мысалы, Интернет желіні кеңейтуге мүмкіндік береді. VPN желілік деректер ағынын басқаруға мүмкіндік береді және деректерді сәйкестендіру және қорғау сияқты маңызды функцияларды ұсынады.



Виртуалды жеке желі (VPN) — бұл басқа желінің үстінен бір немесе бірнеше желілік қосылулар қамтамасыз етуге мүмкіндік беретін технологиялар.

Бұл байланыс пайдаланушының компьютері мен қашықтағы серверді тікелей байланыстыратын — шифрланған туннельді береді және ол нақты IP-адресі жасыру ғана емес, сонымен қатар өз трафигін шифрлеуге мүмкіндік береді (4.1-сурет). Басқаша айтқанда, ешкімге білдірмей кез келген ақпаратты жүктеу мүмкіндігі бар.



4.1-сурет. VPN сұлбасы

Бірақ VPN-ге тек қана ірі кәсіпорындар (мұнай немесе газ компаниялары және т.б.) қол жеткізе алады. Желілік инфрақұрылымды дамыту үшін жеке желілер құрылады. Қазіргі уақытта Интернет желісі қызметтерінің сапасы мен олардың құны арасындағы тиімділігі және қолжетімділігінің арқасында ілгері қадам басып, VPN желісін ығыстырып отыр.

Сондай-ақ VPN кемшіліктеріне оны баптаудың күрделілігін жатқызуға болады. VPN-ді пайдалану әртүрлі өндірушілердің жабдықтарымен қолдау табады, алайда NETGEAR компаниясы дайындамаған өнімге қосылу баптау рәсімін қиындатады.

Виртуалды жеке желінің артықшылықтары:

- үйден жұмыс компьютеріне кіру мүмкіндігі;
- VPN-туннель арқылы тасымалданатын ақпаратты ұстап қалу ықтималдығының төмен болмауы.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Виртуалды жеке желі (VPN) дегеніміз не?
2. Виртуалды жеке желінің артықшылықтары мен кемшіліктерін атаңдар.

ПРАКТИКУМ

А ДЕҢГЕЙІ

Зерттеу жұмысын жүргізіп, кестені толтырыңдар.

Аты	Артықшылықтары	Кемшіліктер
PPTP виртуалды жеке желі		
OpenVPN виртуалды жеке желі		
L2TP виртуалды жеке желі		

В ДЕҢГЕЙІ

Интернет және басқа да қосымша көздерді пайдалана отырып, виртуалды желілердің жіктелуін құрыңдар. Нәтижені сұлба, кесте, блок-схема түрінде мәтіндік немесе графикалық редакторда ұсыныңдар.

С ДЕҢГЕЙІ

Тегін Azure есеп жазбасымен бұлтта жеке виртуалды желі жасаңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§5 Ақпараттық қауіпсіздік

Бүгінгі сабақта:

- ▶ «ақпараттық қауіпсіздік», «күпиялық», «тұтастық» және «қолжетімділік» терминдерінің мағынасын түсінесіңдер.

Тірек ұғымдар:

- ▶ ақпараттық қауіпсіздік;
- ▶ күпиялық;
- ▶ тұтастық;
- ▶ қолжетімділік.

Интернеттен эссе туралы ақпаратты табуға, әуенді тыңдауға, сүйікті кітабыңды оқуға, пойыз немесе ұшақ билеттерін сатып алуға және көптеген форумдарда маңызды тақырыпты талқылауға болады. Интернет оқу немесе достармен араласу үшін қолайлы және пайдалы құрал болып саналады.



Компьютерлер желіге қосылған кездегі деректердің қауіпсіздігі туралы не айтуға болады?

Бірақ Интернеттің пайдасымен қатар зиянды әсері де болуы мүмкін. Жиі кездесетін зиянды жақтары:

- зиянды вирустарды бағдарламалық жасақтама (SOFTWARE) арқылы жұқтыру қаупі. Хакерлер трояндық вирустарды тарату үшін Интернеттен, электрондық поштадан, флэш-дискілерден және т.б. жүктеген файлдарды жиі пайдаланады;
- сөйлесу топтары немесе электрондық пошта арқылы бөтен адамдармен байланыс (киберқылмыскерлер балалардан жеке ақпаратты алу үшін осы ақпаратты пайдаланады);
- Интернеттегі ойын-сауықты іздеу (мысалы, ойындар). Кейде жаңа ойын сайтын іздеп жүргенде, бейтаныс карточкалық серверге кіріп кетіп, көлемді қаржыны ұтқызу мүмкіндігі.



Ақпараттық қауіпсіздікті қамтамасыз ету — ақпараттық ресурстарды сақтау және ақпарат саласындағы жеке және қоғамның заңды құқықтарын қорғау.

Ақпараттық қауіпсіздік — ақпараттың құпиялылығын, тұтастығын және қолжетімділігін қамтамасыз ету процесі.

Құпиялылық — ақпаратты тек уәкілетті пайдаланушыларға ұсыну.

Адалдық — ақпараттың сенімділігі мен толықтығын және оны өңдеу әдістерін қамтамасыз ету.

Қолжетімділік — қажетіне қарай қолданушылардың ақпаратына және тиісті активтеріне қолжетімділікті қамтамасыз ету.

Компьютерлік желілерде ақпараттың қауіпсіздігі дербес компьютермен салыстырғанда төмен, себебі:

- желіде көптеген пайдаланушылар жұмыс істейді, олардың құрамы өзгереді;
- желіге заңсыз қосылу мүмкіндігі бар;
- желілік бағдарламалық жасақтама кемшіліктер бар;
- желі арқылы заңсыз, зиянды бағдарламалардың шабуылдары болуы мүмкін.

Ақпараттың қауіпсіздігіне байланысты мәселелерді Қазақстан Республикасының «Ақпараттық қауіпсіздікті қамтамасыз ету» Заңы реттейді.

Ақпаратты қорғаудың техникалық құралдары — бұл құлыптар, дабыл жүйесі және бейнебақылау жүйелері, ақпараттың сыртқа заңсыз таралу арналарын бұғаттайтын немесе оларды анықтауға мүмкіндік беретін басқа да құрылғылар.

Бағдарламалық құралдар құпия сөз бойынша деректерге қолжеткізуді, ақпаратты шифрлеуді, уақытша файлдарды жоюды, зиянды бағдарламалардан қорғауды және т.б. қамтамасыз етеді.

Ұйымдастыру құралдары:

- байланыс желілерін зиянкестердің оларға қол жеткізу мүмкіндігі болмайтындай етіп орналастыруды;
- ұйымның қауіпсіздік саясатын қамтиды.

Серверлер, негізінен, жеке (күзетілетін) бөлмеде орнатылады және тек желі администраторына ғана қолжетімді болады. Маңызды ақпараттар іркіліс болған жағдайда оны сақтау үшін резервтік тасымалдағыштарға көшіріліп отырады.

Кез келген қорғаудың ең әлсіз буыны — адам. Негізінен, ақпараттың заңсыз таралуы инсайдерлермен байланысты (ағылшынша *inside* — ішкі), яғни инсайдер — фирмадағы жауапкершілігі төмен қызметкерлер. Көп жағдайда құпия ақпараттың заңсыз таралуы көмекші персонал (хатшылар, жинаушылар және т.б.) арқылы жүзеге

асырылуы мүмкін. Сондықтан ешбір адам қалпына келтірмейтіндей (деректерді жою, ұрлау немесе өзгерту, жабдықты істен шығару) зиян тигізе алмауы тиіс.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Интернетте қандай қауіпсіздік шаралары бар?
2. Әлеуметтік желідегі парақшаң бұзылған жағдайда не істеу қажет?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Сұрақтарға жауап беріңдер:

1. Ақпараттың қауіпсіздік аспектілері:
 - а) дискретті;
 - ә) тұтастығы;
 - б) қолжетімділік;
 - в) өзектілігі;
 - г) айқындық.
2. Уәкілеттікке қол жеткізу үшін қолжетімді болуы керек деректердің меншік құқығына кез келген уақыттағы қажеттілік:
 - а) құпиялық;
 - ә) тұтастығы;
 - б) қолжетімділік;
 - в) түпнұсқалық;
 - г) сенімхат.
3. Рұқсатсыз кіру дегеніміз:
 - а) жүйеде қол жеткізуді бақылау ережелерін бұза отырып, нысанаға кіру мүмкіндігі;
 - ә) ұйымдағы сақтық көшірмелерін жасау;
 - б) құпия сөздерді қорғауды айналып өту үшін ұйым өзірлеген ережелер;
 - в) ұйым басшысының келісімінсіз жүйеге ену;
 - г) қажетсіз ақпаратты жою.
4. Ақпараттың тұтастығы деген не?
 - а) қандай да бір субъектінің мүмкіндігіне байланысты ақпараттың сипаты;

- ө) бір ғана пайдаланушы өзгертуі мүмкін ақпараттың сипаты;
- б) файлдардың бірыңғай жиынтығынан тұратын ақпараттың қасиеті;
- в) ақпараттың өзгеріссіз қалпын сақтау қасиеті;
- г) ақпараттың шифрленген түрде сақталуы.

5. Өзге біреулердің деректеріне зиян келтіруі мүмкін ықтимал оқиға, әрекет немесе процесс:

- а) қауіп;
- ө) қауіпті;
- б) ниет;
- в) ескерту;
- г) бұзу.

В ДЕҢГЕЙІ

Ақпараттың қауіпсіздігіне қауіпті ықтимал әрекеттерді тауып, кестені толтырыңдар.

Қауіпті жағдайлар	
Кездейсоқ әрекеттер	Арнайы жасалатын әрекеттер

С ДЕҢГЕЙІ

Төмендегі үлгі бойынша компаниялардың (мысалы: емхана, сақтандыру компаниясының кеңсесі, адвокат кеңсесі, қонақүй, интернет-дүкен т.б.) ақпараттық қауіпсіздік концепцияларын дайындаңдар:

- Ақпараттық қауіпсіздік жүйесінің мақсаты.
- Ақпараттық қауіпсіздік жүйесінің міндеттері.
- Ақпараттық қауіпсіздік нысандары.
- Ықтимал бұзушылар.
- Ақпараттық қауіпсіздік қауіптерінің негізгі түрлері.
- Ақпараттық қауіпсіздік жөніндегі іс-шаралар.
- Антивирустық қорғау бағдарламалық жасақтаманы ұсыныңдар (салыстырмалы баға, мүмкіндіктер және т.б.).

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 6

Ақпаратты қорғау әдістері

Бүгінгі сабақта:

- ▶ мәліметтерді шифрлау қажеттілігін бағалауды үйренесіңдер.

Тірек ұғымдар:

- ▶ ақпараттық қауіпсіздік;
- ▶ ақпаратты қорғау;
- ▶ ақпараттың құпиялылығы;
- ▶ ақпараттың тұтастығы;
- ▶ ақпараттың түпнұсқалығы.

Әлемде, дәл солай Қазақстанда да ғылым, техника және экономиканың дамуында ақпарат маңызды рөл атқарады. Ақпарат елдің және жеке кәсіпорынның экономикалық әлеуетін сипаттайтыны белгілі. Соңғы уақытта ақпаратқа шабуыл жасау туралы хабарламалар барлық бұқаралық ақпарат жүйелерін толтырды. Банктік шоттарды рұқсатсыз бұзып кіру, вирустық бағдарламалардың көмегімен шабуыл жасау — бұл ақпараттық шабуылдың бірнеше мысалы ғана. Мұндай шабуылдармен күресу үшін мемлекет үлкен шығындар жұмсайды. Әрине, ақпаратты қорғауға жұмсалған қаражат қорғалатын ақпарат құндылығымен өлшемдес болу қажет.



Ақпараттық қауіпсіздік — нәтижесінде ақпарат иесіне немесе пайдаланушыларға зиян тиетін кез келген әрекеттерден ақпараттың қорғалуы.

Зиян келтіру, ол ақпаратты жоғалту, оны бұрмалау, сонымен қатар заңсыз бұзып кіруді білдіреді. Ең алдымен мемлекеттік және әскери құпияны қорғау қажет. Сонымен қатар коммерциялық, заң жүзінде, банктік және жеке ақпаратты (төлқұжат деректері, сайттағы құпия сөздер және т.б.) қорғау қажет.

Осыған орай ақпаратқа шабуыл дереккөздерінің бірнеше топтарын ерекшелейік (6.1-сурет):

- вандализм, негізінде, оны әуесқой-хакерлер өзін-өзі таныту мақсатында жасайды;
- жекетұлға мен коммерциялық құрылымдардың ақшалай қаражатын ұрлау мақсатында жасалынатын қылмыстық шабуылдар;
- коммерциялық тыңшылық;
- ақпаратты бұрмалау, жалған ету;
- үкімет мүшелерінің тапсырмасы бойынша тыңшылық (барлау).



6.1-сурет

Қорғау — ақпаратты қорғау механизмін сипаттайтын жалпы термин. Ақпаратты қорғау механизмдеріне мыналар жатады:

- ақпаратты жоғалту мен бұрмалауды болдырмау;
- пайдаланушыға өзінің бағдарламалары мен деректерін қорғау құралдарымен қамтамасыз ету.



Шифрлеу — бұл ашық ақпаратты басқа адамдар түсінбеу үшін шифрленген ақпаратқа түрлендіру.

Нәтижесінде:

- *ақпараттың құпиялылығын* — ол ақпаратқа тек ресми пайдаланушылар қол жеткізе алатынын білдіреді;
- *тұтастықты* — ол ақпарат бастапқы түрінде болады және оны жіберу немесе сақтау кезінде рұқсатсыз өзгерістер болмағанын білдіреді;
- *түпнұсқалықты қамтамасыз ету* — ақпарат авторы ретінде хабарланған тұлға ақпарат дереккөзі болатынын білдіреді.

Қазақстанда ақпаратты қорғаумен байланысты сұрақтарды «Ақпараттандыру туралы» Заң реттейді.

Ақпаратты қорғау бағыттарына *антивирусология және жүйелік қорғау* жатады.



Антивирусология — компьютерлік вирустармен және басқа өздігінен таралатын бағдарламалармен күресу жолдары туралы ғылым.



Жүйелік қорғау — техниканың істен шығу, қате әрекеттер және басқа апатты жағдайларда деректердің тұтастығын және қолжетімділікті қамтамасыз етуге бағытталған аппараттық және бағдарламалық құралдар жиынтығы.

Қорғау жүйесінің өлсіз жағы адам болып табылады. Мысалы, жауапкершілігі төмен қызметкерлер. Құпия ақпараттың хатшылар, кеңсені жинастыратын және басқа да көмекші қызметкерлер арқылы таралу жағдайлары белгілі. Осыған дейін құжатты (ақпаратты тасымалдаушыны) жекетұлғамен қорғау, оны арнайы курьермен жіберу қолданылып келді. Бірақ құжатты ұрлауға, курьерді жолда ұстап немесе пара ұсынуға болады. Қазіргі уақытта жүйелік қорғауды жүзеге асыру үшін заманауи телекоммуникациялық байланыс арналары қолданылады. Бірақ бұл тәсілдер қомақты қаржы салуды қажет етеді. Ғылым мен техника дамуының заманауи деңгейінде қашықтықтағы абоненттер арасында үлкен көлемді ақпаратты бірнеше рет жіберу қиындық тудыруы мүмкін.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Ақпараттық қауіпсіздік дегеніміз не?
2. Неге ақпаратты қорғау қажет?
3. «Ақпаратты қорғау» ұғымы нені қамтиды?
4. Ақпаратты қашықтықта құпия түрінде жіберу тәсілдерін атаңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

§7

Тұлғаны сәйкестендіру (идентификациялау) әдістері

Бүгінгі сабақта:

- ▶ пайдаланушы деректерінің қауіпсіздік шарасын қолдануды үйренесіңдер.

Тірек ұғымдар:

- ▶ құпия сөз;
- ▶ кілт;
- ▶ есеп жазбасы;
- ▶ сәйкестендіру;
- ▶ түпнұсқалық;
- ▶ биометриялық түпнұсқалық.

Пайдаланушылар деректерінің қауіпсіздігі үшін құпия сөздер, есеп жазбалары қолданылады. Windows операциялық жүйесі бір компьютерде бірнеше пайдаланушыға есеп жазбасы арқылы жұмыс істеуге мүмкіндік беретінін білесіңдер.

Есеп жазбасы кіруге рұқсат етілген программалық құралдарды, сыртқы бейнесін және т.б. анықтайтын компьютерді баптау жиынтығын ұсынады. Есеп жазбасының аты (логин) операциялық жүйеде сәйкестендіру үшін қолданылады. Әрине, үй компьютерінде, негізінде, бір есеп жазбасы қолданылады. Бірақ әрбір пайдаланушының өзінің есеп жазбасы болғаны дұрыс. Қалай ойлайсыңдар, ол не үшін қажет?

Біріншіден, ол компьютерді тәжірибесі аз пайдаланушылардан қорғайды. Тәжірибелі пайдаланушының әкімшілік құқығы, ал басқаларында жай әдеттегі қорғау түрі болады. Сонда компьютерге ешкім зиян келтіре алмайды.

Екіншіден, ол компьютерде деректер мен әрекеттердің құпиялылығын қамтамасыз етеді. Әрбір пайдаланушы өзінің файлдарын қарай алады, ал басқа құжаттарды ашу үшін құпия сөз енгізу қажет.



Қалай ойлайсыңдар, «авторизация», «сәйкестендіру» және «аутентификация» ұғымдары арасында қандай айырмашылық бар?

Алдымен сәйкестендірудің (идентификация) не екенін қарастырайық.



Сәйкестендіру — идентификатор арқылы субъектіні тану процедурасы. Басқаша айтқанда, ол аты, логин немесе нөмірін және т.б. анықтау.

Мысалы, операциялық жүйеге немесе электрондық поштаға кіру кезінде сәйкестендіру орындалады.

Сәйкестендірудің не екенін мысалмен қарастырайық. Егер белгісіз нөмірден қоңырау шалса, «Бұл кім?» деген сұрақ туындайды. Бұл жағдайда қоңырау шалушының жауабы идентификация болып табылады.

<https://vk.com/id186301730>

7.1-сурет. «Контакт» әлеуметтік желісінде идентификатор мысалы

Сәйкестендіруге телефон нөмірі, төлқұжат нөмірі, e-mail, әлеуметтік желідегі парақша нөмірі және т.б. жатады (7.1-сурет).

Сәйкестендіруден (идентификациядан) кейін түпнұсқалық (аутентификация) орындалады.



Түпнұсқалық (аутентификация) — түпнұсқалықты тексеру процедурасы. Басқаша айтқанда, пайдаланушыны құпия сөз арқылы, хатты электрондық пошта арқылы және т.б. тексереді.

Түпнұсқалықты анықтау үшін бірнеше факторлар қолданылады:

- құпия сөз — тек біз білетініміз (сөз, PIN-код, графикалық кілт, сейф үшін код және т.б.);
- құрылғы — сенде бар болады (пластикалық карточка, құлыптың кілті, USB-кілт);
- биометрика — саусақ ізі, портрет, көз торы т.б. бөліктер.

Сонымен құлыпқа кілт салғанда, құпия сөз енгізгенде немесе саусақ ізі сенсорына саусақты басқанда түпнұсқалықтан өтеді (7.2-сурет).

Түпнұсқалық тексерілгеннен кейін ғана файлға кіруге рұқсат беріледі, яғни авторизация орындалады.



7.2-сурет. Түпнұсқалық кезінде құпия сөз ретінде саусақтың ізін қолдануға болады



Авторизация — қандай да бір ресурсқа кіруге рұқсат беру (мысалы, құпия сөз енгізгеннен кейін электрондық поштаға кіру, саусақтың ізін түсіргеннен кейін смартфонды бұғаттан шығару және т.б.).



Бұл үш ұғым бір-бірімен қалай байланысты (7.3-сурет)?



7.3-сурет

1. Алдымен сәйкестендіру — атын анықтайды (логин немесе нөмір).
2. Содан кейін түпнұсқалық (аутентификация) — құпия сөзді тексереді (кілт немесе саусақтың ізі).
3. Және авторизация — рұқсат береді.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Есеп жазбалары не үшін құрастырылады?
2. Сәйкестендіру, түпнұсқалық дегеніміз не?
3. Биометриялық түпнұсқалықты қолдану салаларын атаңдар.

ПРАКТИКАЛЫҚ ЖҰМЫС

Жобалық жұмыс

1. Қауіпті программалар және заң.
2. Тегін антивирустық бағдарламалық жасақтама.
3. Шифрлеу және заң.
4. Шифрлеудің криптотұрақтылығы.
5. Жиілікке талдау жасау.

Жобаны бағалау критерийлері

Өзектілігі — таңдалған зерттеу тақырыбының өзектілігіне негізделген.

Дербестік — жобалау қызметінің барлық кезеңдерін жоспарлау және орындау.

Мәселелік — проблеманы, проблемалық жағдайды тұжырымдай білу.

Мазмұндылығы — ақпараттылық деңгейі, жобаның мағыналық сыйымдылығы.

Ғылымилық — нақты ғылыми терминдерді пайдалану және олармен жұмыс істеу мүмкіндігі.

Ақпаратпен жұмыс — ақпаратпен жұмыс істеу деңгейі, жаңа ақпаратты іздеу тәсілі.

Жүйелілік — жалпылама әрекет тәсілін ерекшелеу және оны жұмыста тапсырмаларды шешу кезінде қолдану.

Интегративтілік — білімнің әртүрлі салаларының байланысы.

Коммуникативтік — нақты стилистикалық сауатты жұмыстың нәтижесін баяндауы.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

II ТАРАУ

ДЕРЕКТЕРДІ ҰСЫНУ

Бұл тараудан білесіңдер:

- ▶ сандарды бір санау жүйесінен екінші санау жүйесіне аударуды;
- ▶ «логика» ұғымын;
- ▶ логикалық функциялардың жазылу тәртібін;
- ▶ ақиқат кестесін құрастыруды.

Мынаны үйренесіңдер:

- ▶ ондық жүйедегі бүтін сандарды екілік санау жүйесіне және кері аударуды;
- ▶ логикалық операцияларды қолдануды (дизъюнкция, конъюнкция, инверсия);
- ▶ берілген логикалық элементтер үшін ақиқат кестесін құруды;
- ▶ функциялар мен процедураларды қолданып, бағдарламалау тілінде код жазуды;
- ▶ Unicode және ASCII таңбаларын кодтау кестесін салыстыруды.

§ 8

Сандарды бір санау жүйесінен екінші санау жүйесіне аудару

Бүгінгі сабақта:

- ондық жүйедегі бүтін сандарды екілік санау жүйесіне және кері аударуды үйренесіңдер.

Тірек ұғымдар:

- кеңейтілген және жинақталған түрде;
- негізгі санау жүйелері;
- екілік, сегіздік, ондық, он алтылық, позициялық;
- позициялық емес санау жүйелері.

Он (10^1), жүз (10^2), мың (10^3), миллион (10^6) сандарды санауды ыңғайлы болу үшін оның дәрежесі түрінде жазамыз. Мұндай есептеу тәсілі және дөңгелектеуді қабылдау аданың анатомиялық ерекшеліктеріне де байланысты берілген. Адамның 10 саусағы бар, ал саусақтар, ең алдымен, санау құралы болғандықтан 10 саны негіз болып қалыптасқан. Сондықтан бұны «ондық» санау жүйесі деп атайды.

Ондық жүйеде сандардың реттік тізбектері көрсетіледі 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.



Санау жүйесі — сандар деп аталатын белгілі бір алфавиттік символдардың санын көрсету тәсілі.

Барлық сандар жүйесі *позициялық және позициялық емес* болып бөлінеді.

Позициялық санау жүйесінде цифрмен белгіленген мән (шама) санның позициясына байланысты болады.

Ең көп таралған позициялық санау жүйелері: ондық, екілік, сегіздік, он алтылық (8.1-кесте).

8.1-кесте

Негізгі санау жүйесінің алфавиті

Санау жүйесі	Негізі	Сандар алфавиті
Позициялық		
Ондық	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Екілік	2	0, 1
Сегіздік	8	0, 1, 2, 3, 4, 5, 6, 7
Он алтылық	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
Позициялық емес		
Римдік		I(1), V(5), X(10), L(50), C(100), D(500), M(1000)

Ондық жүйедегі сандар жазылуының ережелерін толығырақ қарастырайық. Санды жазып жатқанда, оны сан жүйесінің негізі 10 санының дәрежелерінің қосындысы ретінде көрсетеміз.

Мысалы, 777 саны жеті жүздік, жеті ондық және жеті бірліктің қосындысының кеңейтілген түрі былай жазылады:

$$777 = 7 \cdot 10^2 + 7 \cdot 10^1 + 7 \cdot 10^0,$$

яғни, бір сан енгізудегі позицияға байланысты түрлі мәндерді көрсетеді. Ал 777 саны — бұл санның жинақталған түрде жазылуы. Ондық бөлшектерді жазу үшін санның негізі ретінде дәреженің теріс мәні қолданылады. Мысалы, кеңейтілген пішіндегі 777,77 саны келесідей жазылады:

$$777,77 = 7 \cdot 10^2 + 7 \cdot 10^1 + 7 \cdot 10^0 + 7 \cdot 10^{-1} + 7 \cdot 10^{-2}.$$

Екілік санау жүйесінің негізі 2, ал жазылуы екі саннан (0 және 1) тұрады.

Кеңейтілген түрдегі екілік санау жүйесіндегі сандар 0 немесе 1 коэффициенттері бар негізі 2 болатын дәрежесінің қосындысы ретінде жазылады.

Мысалы, көптаңбалы екілік санау жүйесінде берілген 101101_2 .

Санының жазылуындағы екі (төменгі индекс) санның қай жүйеге жататынын көрсетеді. Егер сан қолданылып жүрген ондық санау жүйесінде жазылса, онда төменгі 10 индексі жазылмайды.

Берілген санның кеңейтілген түрі былай жазылады:

$$101101_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 45_{10}.$$

Демек, екілік санау жүйесі нөлдер мен бірліктердің тізбегі болып табылады. Сонымен қатар ол разрядтардың жеткілікті санын қамтиды.

Екілік санау жүйесінің кемшілігі санды жазудың үлкендігі болып табылады. Мысалы, 227_{10} саны екілік жүйеде былай жазылады: 11100011_2 . Оған қарағанда сегіздік немесе он алтылық санау жүйелерін қолданған ыңғайлы.

БҰЛ ҚЫЗЫҚ!

Ежелгі дәуірде он екілік санау жүйесі кең таралды, оның шығу тарихы ондық санау жүйесі сияқты, саусақпен санауға негізделген. Осы санау жүйесінің кейбір бөліктері бүгінгі күні де ауызша және әдет-ғұрыптарда да сақталған. Мысалы, екінші разрядты бірліктің атауы – 12 саны – «дюжина». Көптеген бұйымдарды ондықпен емес, он екі етіп жинақтайды, мысалы, жиһаз жиындарындағы орындықтан немесе асхана құралдарының сандарында сақталған.

Кейбір Африка тайпаларында және Ежелгі Қытайда бес таңбалы санау жүйесі қолданылған.

Орталық Америкада (ежелгі ацтектер мен майялар арасында) және Батыс Еуропадағы ежелгі кельттердің арасында жиырмалық санау жүйесін пайдаланған.

Б.з.б. 2 мың жыл бұрын Вавилон халқы он алтылық санау жүйесін қолданған. Вавилондықтардың он алтылық санау жүйесінің негізіне позициялық принцип жатады. Математика мен астрономия ғылымдарының дамуында үлкен рөл атқарды, оның іздері бүгінгі күнге дейін сақталған. Вавилондықтардың үлгісіне сәйкес шеңберді 360 бөлікке (градусқа), сағатты 60 минутқа, ал минутты 60 секундқа бөледі.



Ондық бөлшекпен берілген санның бүтін бөлігін екілік жүйеге қалай аударуға болады?

Сандарды ондық санау жүйесіне аудару

Екілік, сегіздік, он алтылық санау жүйелерінде жазылған сандарды ондық жүйеге көшірген өте жеңіл. Бұл үшін санның кеңейтілген түрін жазып, мәнін есептеу жеткілікті.

Сандарды екілік жүйеден ондыққа аудару

Кез келген екілік жүйедегі санды, мысалы, $11,01_2$ алайық. Оның кеңейтілген түрін жазып, есептеулер жүргізейік:

$$11,01_2 = 1 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 2 + 1 + 0 + 0,25 = 3,25_{10}.$$

Санды сегіздік жүйеден ондық жүйеге аудару. Кез келген сегіздік санды алайық, мысалы, $17,4_8$. Оны кеңейтілген түрде жазып, есептеу жүргізейік:

$$17,4_8 = 1 \cdot 8^1 + 7 \cdot 8^0 + 4 \cdot 8^{-1} = 8 + 7 + 0,5 = 15,5_{10}.$$

Санды он алтылық жүйеден ондық жүйеге аудару. Кез келген он алтылық санды алайық, мысалы, $51C_{16}$. Оны кеңейтілген түрде жазып, есептеу жүргіземіз:

$$51C_{16} = 5 \cdot 16^2 + 1 \cdot 16^1 + 12 \cdot 16^0 = 1280 + 16 + 12 = 1308_{10}.$$

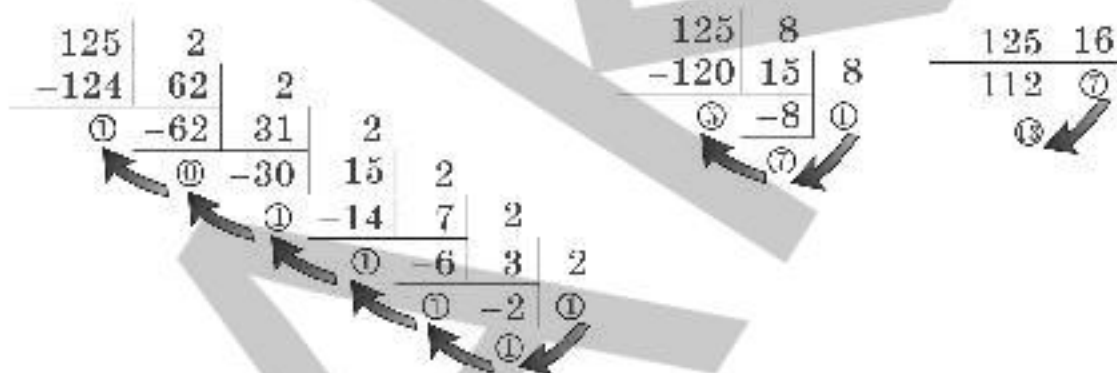
Сандарды ондық жүйеден екілік, сегіздік, он алтылық жүйеге көшіру күрделілеу және өртүрлі тәсілдермен орындалады. Осы сандарды көшіру алгоритмдерінің бірін қарастырайық.

Бүтін санды аудару алгоритмі

1. Ондық санды санау жүйесінің негізіне бөлуіміз керек. Бұл процесс бөлінді белгіштен кіші болғанша жалғасады.

2. Алынған қалдықтарды кері тізбекпен жазайық.

Ондық жүйедегі 125 санын екілік, сегіздік және он алтылық жүйелерге көшіруге мысал қарастырайық (8.1-сурет):



8.1-сурет

$$125_{10} = 1111101_2; \quad 125_{10} = 175_8; \quad 125_{10} = 7D_{16}.$$

Бұдан шығатыны: $125_{10} = 1111101_2 = 175_8 = 7D_{16}$. Демек, $13_{10} = D_{16}$.

Енді ондық бөлшекті басқа санау жүйелеріне көшіру алгоритмін қарастырайық.

Ондық бөлшекті аудару алгоритмі

1. Ондық бөлшек жүйе негізіне тізбектей көбейтіледі, ал алынған бөлшек бөлігі жүйе негізіне тағы да көбейтіледі. Бұл процесс нөлдік бөлшек бөлік алынғанша жалғасады.

2. Алынған бүтін бөліктерді тік бағанға жазасындар.

Ондық жүйедегі $0,125$ санын екілік, сегіздік және он алтылық жүйелерге көшіруге мысал қарастырайық (8.2-сурет).

$$\begin{array}{r}
 0 \mid 125 \\
 \times 2 \\
 \hline
 \textcircled{0} \mid 250 \\
 \times 2 \\
 \hline
 \textcircled{0} \mid 500 \\
 \times 2 \\
 \hline
 \textcircled{1} \mid 000
 \end{array}
 \qquad
 \begin{array}{r}
 0 \mid 125 \\
 \times 8 \\
 \hline
 \textcircled{1} \mid 000
 \end{array}
 \qquad
 \begin{array}{r}
 0 \mid 125 \\
 \times 16 \\
 \hline
 \textcircled{2} \mid 000
 \end{array}$$

8.2-сурет

Тік сызық бүтін бөліктерді бөлшек бөлігінен бөліп тұрады.

$$0,125_{10} = 0,001_2, \quad 0,125_{10} = 0,1_8, \quad 0,125_{10} = 0,2_{16}.$$

$$\text{Осыдан: } 0,125_{10} = 0,001_2 = 0,1_8 = 0,2_{16}.$$

Бүтін сандарды екілік есептеу жүйесінен сегіздік және он алтылық санау жүйесіне аудару. Бүтін екілік санды сегіздік санға аудару үшін оны үш саннан, оңнан солға бөлу, содан кейін әр топты сегіздік санға түрлендіру керек. Егер соңғы сол топта үш саннан аз болса, онда оны сол жағынан нөлдермен толықтыру қажет.

101111₂ екілік санын сегіздікке аудару мысалын қарастырайық:

$$101111_2 \rightarrow \underbrace{1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0}_5 \underbrace{1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0}_7 \rightarrow 57_8.$$

Тез аудару үшін екілік жүйе мен сегіздік жүйедегі сандардың ақиқат кестесін қолдануға болады (8.2-кесте).

8.2-кесте

Екілік жүйе	Сегіздік жүйе
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Бүтін екілік санды он алтылық санға аудару үшін оны төрт саннан, оңнан солға бөлу, содан кейін әрбір топты он алтылық санға түрлендіру керек. Егер соңғы сол топта үш саннан аз болса, онда оны сол жағынан нөлдермен толықтыру қажет.

101111_2 екілік санын он алтылық санға аудару мысалын қарастырайық:

$$00101111_2 \rightarrow \underbrace{0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0}_2 \underbrace{1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0}_F \rightarrow 2F_{16}$$

Тез аудару үшін екілік, ондық жүйемен он алтылық жүйедегі сандардың ақиқат кестесін қолдануға болады (8.3-кесте).

8.3-кесте

Ондық жүйе	Екілік жүйе	Он алтылық жүйе
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Бүтін сандарды сегіздік және он алтылық санау жүйелерінен екілік жүйеге аудару. Бүтін санды сегіздік және он алтылық санау жүйелерінен екілік жүйеге аудару үшін санның цифрларын екілік сан топтарына түрлендіру қажет. Сегіздік жүйеден екілік жүйеге аудару үшін санның әрбір цифрын үш екілік саннан тұратын топқа, ал он алтылық саннан төрт саннан тұратын топқа айналдыру керек.

127 санын сегіз және он алтылық жүйеден екілік жүйеге аудару үлгісін қарастырайық:

$$127_8 = \underbrace{001}_1 \underbrace{010}_2 \underbrace{111}_7 = 1010111_2$$

$$127_{16} = \underbrace{0001}_1 \underbrace{0010}_2 \underbrace{0111}_7 = 100100111_2$$

БАҚЫЛАУ СҰРАҚТАРЫ



1. Екілік санау жүйесінің алфавиттік белгілері қандай?
2. Неге есептеу құралдары екілік санау жүйесін негіз ретінде алады?
3. 1000000 санау жүйесін пайдалануға бола ма? Мұндай жүйеде қандай қиындық туындауы мүмкін?

ПРАКТИКАЛЫҚ ЖҰМЫС

Ондық және аралас сандарды басқа санау жүйелеріне көшіру.

А ДЕҢГЕЙІ

Берілген бүтін сандарды ондық санау жүйесінен екілік санау жүйесіне аударыңдар: 12; 523; 76; 121.

В ДЕҢГЕЙІ

Келесі сандарды ондық санау жүйесіне аударыңдар: $0,025 \rightarrow A_2$; $0,0625 \rightarrow A_{16}$; $0,0625 \rightarrow A_8$.

С ДЕҢГЕЙІ

Төмендегі сандарды ондық санау жүйесіне аударыңдар: $1,25 \rightarrow A_2$; $2,25 \rightarrow A_8$; $3,75 \rightarrow A_{16}$.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§9

Практикум. Сандарды бір санау жүйесінен екіншісіне аудару

Бүгінгі сабақта:

- ▶ бүтін сандарды ондық санау жүйесінен екілік санау жүйесіне аударуды үйренесіңдер.

Тірек ұғымдар:

- ▶ кеңейтілген және жинақталған түрде;
- ▶ негізгі санау жүйелері;
- ▶ алгоритм.

«Санау жүйесі» практикумы

1-нұсқа

1. Неге және не үшін электронды-есептеу техникасын ондық санау жүйесінен екілік жүйеге ауыстырды?

2. Сегіздік санау жүйесінде қандай сандар қолданылады?

3. 9_{16} санынан кейін он алтылық санау жүйесінде қандай сан болады?

Сандарды бір санау жүйесінен екіншісіне аударыңдар.

4. Позициялар бойынша бөлу:

$$2154_{10} = \dots$$

$$190_{10} = \dots_2$$

$$101110_2 = \dots_{10}$$

$$217_{10} = \dots_8$$

$$563_8 = \dots_{10}$$

$$436546_8 = \dots_2$$

$$11100101010010100_2 = \dots_8$$

$$432_{10} = \dots_{16}$$

$$F12B_{16} = \dots_{10}$$

$$7FD56E_{16} = \dots_2$$

$$1010101010100010101110_2 = \dots_{16}$$

2-нұсқа

1. Ақпарат — бұл...

2. Он алтылық санау жүйесі — бұл...

3. Екілік санау жүйесіндегі ең үлкен сан қандай?

Сандарды бір санау жүйесінен екіншісіне аударыңдар.

4. Позициялар бойынша бөлу:

$$1256_{10} = \dots$$

$$309_{10} = \dots_2$$

$$10110_2 = \dots_{10}$$

$$156_{10} = \dots_8$$

$$363_8 = \dots_{10}$$

$$1666_8 = \dots_2$$

$$1010011011010100_2 = \dots_8$$

$$436_{10} = \dots_{16}$$

$$12CA_{16} = \dots_{10}$$

$$E15C6E_{16} = \dots_2$$

$$10010011001011010_2 = \dots_{16}$$

БАҚЫЛАУ СҰРАҚТАРЫ



1. Позициялық санау жүйесіндегі негіз деген не?

2. Он алтылық санау жүйесінің басқалардан артықшылығы неде?

3. Сандарды екілік жүйеден сегіздік (он алтылық) жүйеге қалай аударуға болады?

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§10-11

Логикалық операциялар. Ақиқат кестесін құру

Бүгінгі сабақта:

- ▶ логикалық операцияларды (дизъюнкция, конъюнкция, инверсия) қолдануды;
- ▶ берілген логикалық элементтер үшін ақиқат кестесін құруды үйренесіндер.

Тірек ұғымдар:

- ▶ дизъюнкция;
- ▶ конъюнкция;
- ▶ инверсия.



Логика — пікірлер және олардың байланыстары туралы ҒЫЛЫМ.

Ойлау, пайымдау әдістері туралы алғашқы ілімдер Ежелгі Шығыс елдерінде пайда болды, бірақ оның негізіне грек ойшылы Аристотель құрған ілім жатады. Неміс ғалымы Г.В. Лейбниц (1666 ж.) алғаш рет ойлау, пайымдау заңдарын табиғи тілден пікірлер арасындағы байланысты математикалық қатынастармен анықтайтын математика тіліне көшіруге тырысты.

100 жылдан соң ағылшын математигі Джордж Буль математикалық заңдылықтарға бағынатын логикалық өмбебап тілді құру туралы *Лейбниц идеясын* дамытты. Ол дербес алгебраны барлық объектілерге, атап айтқанда, сандар, өріптерден бастап сөйлемдерге дейін қолдануға болатын белгілер мен ережелердің жүйесін ойлап тапты. Джордж Буль логикалық алгебраның атасы болып табылады.

XX ғасырдың басына дейін логика іс жүзінде дамыған жоқ, ол тек ЭЕМ теориясының пайда болуынан бастап қана өзінің қарқынды дамуын бастады.

Барлық ғылым сияқты логика да бірнеше пәнді қамтиды. Мысалы, формалды логика, математикалық логика, ықтималдық логика, диалектикалық логика және т.б. пәндерді атауға болады.

Формалды логика сөз арқылы берілген пікірлерді талдаумен байланысты.



Готфрид Вильгельм
Лейбниц
(1646—1716)



Джордж Буль
(1815—1864)

Ықтималдық логика параметрлері кездейсоқ болып келетін тәжірибелерді бірнеше мәрте жасауға негізделген. Алынған нәтиженің дәлдігі жасалынған тәжірибелер санына тәуелді. Мысалы, тиынды лақтырғанда «елтаңба» немесе «цифр» жағымен түсу ықтималдығы S -ке тең, яғни тиынды лақтыру саны артқан сайын осы ықтималдық нақтылана түседі.

Математикалық логика дәлелдеулер техникасын зерттейді. Бағдарлама жасау математика секілді нақтылықты және қатаң реттілікті талап етеді. Пікірлер алгебрасы математикалық логика аймағы болып табылады.

Логика алгебрасы пікірдің ақиқат немесе жалған екеніне қатысты тұжырымдауға болатын идеал пікірлерді зерттейді. Логика алгебрасы пікірдің мағынасына назар аударып, терең қарастырмайды. Сондықтан логика алгебрасы тек екі мәнге ие болып, пікірлердің кез келгені «ақиқат» немесе «жалған» пікірлерінің біреуін ғана сипаттайды.

Логика алгебрасында қарапайым пікірлер латынның бас әріптерімен таңбаланады:

$A = \{\text{Абай — қазақ халқының ұлы ақыны}\}.$

$B = \{\text{А. С. Пушкин — ұлы математик}\}.$

Ақиқат пікірге «1», ал жалғанға «0» белгісі сәйкес қойылады. Сондықтан $A = 1, B = 0.$

Құрмалас пікірлер табиғи тілде пікірлер алгебрасында логикалық амалдармен алмастырылатын «және», «немесе» жалғаулықтары мен «теріске шығару» сөздері арқылы құрылады. Логикалық амалдар ақиқаттық кестесімен беріледі және графикалық түрде Эйлер-Венн диаграммасы көмегімен бейнеленеді.

Логикалық көбейту (конъюнкция). Екі немесе одан көп пікірлерді «және» жалғаулығы көмегімен біріктіру амалы *логикалық көбейту* немесе *конъюнкция* деп аталады.

Конъюнкция:

— табиғи тілде «және» жалғаулығына сәйкес келеді;

— пікірлер алгебрасында \wedge немесе $\&$ белгілерінің бірімен таңбаланады.



Конъюнкция — ол әрбір қарапайым екі пікірге олардың екеуі де ақиқат болғанда ғана ақиқат мән қабылдайтын күрделі пікірді сәйкес қоятын логикалық амал.

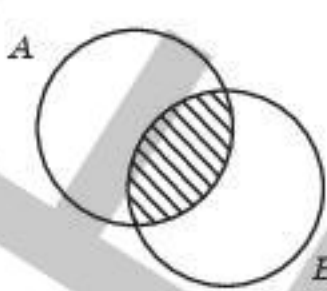
Жиындар алгебрасында конъюнкцияға жиындардың қиылысу амалы сәйкес келеді, яғни A мен B логикалық жиындарын көбейту

нәтижесінде алынған және екі жиынға да бір мезгілде жататын элементтерден тұратын жиынға сәйкес жиын алынады.

Логикалық көбейту функциясының ақиқаттық кестесі мен графикалық бейнеленуі 10.1-кестеде көрсетілген.

10.1-кесте

Логикалық көбейту функциясының ақиқат кестесі

Ақиқат кестесі			Эйлер-Венн диаграммасы
A	B	$A \wedge B$	
0	0	0	
0	1	0	
1	0	0	
1	1	1	

Мысал қарастырайық. Бізге құрмалас пікірлер берілген:

- 1) « $6 : 2 = 2$ және $6 : 3 = 3$ »;
- 2) « $6 : 2 = 2$ және $6 : 3 = 2$ »;
- 3) « $6 : 2 = 3$ және $6 : 3 = 3$ »;
- 4) « $6 : 2 = 3$ және $6 : 3 = 2$ ».

Логикалық көбейту амалының көмегімен құрылған бұл пікірлерде тек төртінші пікір ақиқат, себебі алғашқы үш құрмалас пікірдің ерқайсысында ең болмағанда бір қарапайым пікір жалған.

Логикалық қосу (дизъюнкция). Екі немесе одан да көп пікірлерді «немесе» жалғаулығы көмегімен біріктіру амалы *логикалық қосу* немесе *дизъюнкция* деп аталады.

Дизъюнкция:

- табиғи тілде «немесе» жалғаулығына сәйкес келеді;
- пікірлер алгебрасында \vee белгісімен таңбаланады.



Дизъюнкция — ол әрбір қарапайым екі пікірге олардың екеуі де жалған болғанда ғана жалған мән қабылдайтын, пікірлердің ең болмағанда біреуі ақиқат болғанда ақиқат мән қабылдайтын күрделі пікірді сәйкес қоятын логикалық амал.

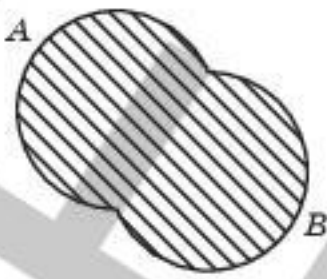
Жиындар алгебрасында дизъюнкцияға *жиындардың бірігу* амалы, яғни A және B жиындарын қосу нәтижесінде алынған жиынға не A ,

не B жиындарының бірінде жататын элементтерден тұратын жиын сәйкес келеді.

Логикалық қосу функциясының ақиқаттық кестесі мен графиктік бейнелеуі 10.2-кестеде көрсетілген.

10.2-кесте

Логикалық қосу функциясының ақиқат кестесі

Ақиқат кестесі			Эйлер-Венн диаграммасы
A	B	$A \vee B$	
0	0	0	
0	1	0	
1	0	0	
1	1	1	

Мысал қарастырайық. Бізге құрмалас пікірлер берілген:

- 1) « $6 : 2 = 2$ немесе $6 : 3 = 3$ »;
- 2) « $6 : 2 = 2$ немесе $6 : 3 = 2$ »;
- 3) « $6 : 2 = 3$ немесе $6 : 3 = 3$ »;
- 4) « $6 : 2 = 3$ немесе $6 : 3 = 2$ ».

Логикалық қосу амалының көмегімен құрылған бұл пікірлерде тек бірінші пікір жалған, себебі соңғы үш құрмалас пікірдің әрқайсысында ең болмағанда бір қарапайым пікір ақиқат.

Логикалық терістеу (инверсия). «Емес» жалғаулығының көмегімен пікірді теріске шығару амалы логикалық *теріске шығару* немесе *инверсия* деп аталады.

Инверсия:

- табиғи тілдегі «*дұрыс емес*», «*теріске шығару*» сөздеріне және «*емес*» жалғаулығына сәйкес келеді;
- пікірлер алгебрасында \bar{A} белгісімен таңбаланады.



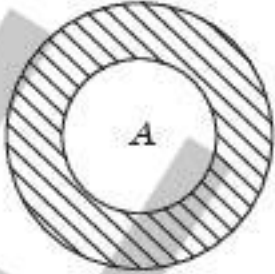
Теріске шығару — ол әрбір қарапайым пікірге негізгі пікірдің жалған екенін пайымдайтын күрделі пікірді сәйкес қоятын логикалық амал.

Жиындар алгебрасында логикалық теріске шығару амалына *эмбебап жиынға дейін толықтауыш* амалы сәйкес келеді, яғни A жиынын теріске шығару нәтижесінде оған A жиынына жатпайтын элементтерден құралған \bar{A} жиыны сәйкес қойылады.

Логикалық теріске шығару функциясының ақиқат кестесі мен графиктік бейнелеуі 10.3-кестеде көрсетілген.

10.3-кесте

Логикалық терістеу функциясының ақиқат кестесі

Ақиқат кестесі	Эйлер-Венн диаграммасы						
<table border="1"> <tr> <td>A</td> <td>\overline{A}</td> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	A	\overline{A}	0	1	1	0	
A	\overline{A}						
0	1						
1	0						

Біз қарастырған $A =$ «Алтыны екіге бөлсең, үшке тең» мысалы ақиқат пікір болсын, онда логикалық теріске шығару арқылы құрылған $\overline{A} =$ «Алтыны екіге бөлсең, үшке тең емес» пікірі — жалған.



Ақиқат кестесі — кіріс айнымалылардың барлық мүмкін болатын мәндерінде күрделі пікірдің ақиқаттығын көрсететін кесте.

Логикалық өрнекті есептеу тәртібі:

1. Жақшадағы өрнек есептеледі.
2. Инверсия (кері сан) есептеледі.
3. Конъюнкция (көбейту) есептеледі.
4. Дизъюнкция (қосу) есептеледі.

$F = (K \vee C) \wedge \neg C \rightarrow K$ мысалын қарастырайық және осы құрамдас пікір үшін ақиқат кестесін құрайық.

Ақиқат кестесін құру кезінде белгілі бір өрекеттер тізбегі бар:

— ақиқат кестесіндегі жолдар санын анықтау қажет. Жолдар саны $= 2n$, мұндағы n — логикалық айнымалылардың саны;

— ақиқат кестесіндегі бағандардың санын анықтау қажет. Бағандар саны $=$ логикалық айнымалылардың саны $+$ логикалық операциялардың саны;

— жолдар мен бағандардың көрсетілген санымен ақиқат кестесін құру, жақшалар мен басымдықтарды (\neg , $\&$, \vee) ескеріп, логикалық операцияларды орындау бірлігіне сәйкес кесте бағандарының атын енгізу қажет;

— кіріс айнымалы бағандарын мән жиынтығымен толтырыңдар;

— белгіленген бірлікке сәйкес логикалық операцияларды орындау арқылы бағандар бойынша ақиқат 10.4-кестесін толтырыңдар.

K	C	$\neg C$	$K \vee C$	$(K \vee C) \wedge \neg C$	$(K \vee C) \wedge \neg C \rightarrow K$
0	0	1	0	0	1
0	1	0	1	0	1
1	0	1	1	1	1
1	1	0	1	0	1

БАҚЫЛАУ СҰРАҚТАРЫ



1. Конъюнкция дегеніміз не?
2. Дизъюнкция дегеніміз не?
3. Логикалық көбейту дегенді қалай түсінесіңдер?
4. Логикалық қосу дегеніміз не?
5. Логикалық терістеу дегеніміз не?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Логикалық «және», «немесе» жалғаулықтарын қолданып, қарапайым пікірлерден күрделі пікір құрындар және оның ақиқаттығын анықтаңдар.

Барлық оқушылар информатиканы оқиды. } Барлық оқушылар информатика мен ағылшын тілін оқиды.
 Барлық оқушылар ағылшын тілін үйренеді. }

1. Ербол Мадинадан үлкен. Салима Мадинадан үлкен.
2. Қызыл доп жасыл доптан үлкен. Қызыл доп сары доптан үлкен.
3. Ертең қар жауады. Ертең суық болады.
4. Қайрат үй тапсырмасын орындайды. Қайрат DVD-дегі футболды көреді.
5. Айгүл түскі асқа барады. Айгүл пианинода ойнауды үйренеді.
6. Арман — Олжастың досы. Арман — Аманның досы.

В ДЕҢГЕЙІ

1. Төменде берілгендерді логикалық өрнек ретінде жазыңдар:
 - а) $0 < X \leq 3$ және $Y < 5$;
 - ә) X, Y, Z сандарының әрқайсысы 0-ге тең.
2. Ақиқат кестесін келесі функциялар үшін қарастырыңдар:
 - а) $\neg(A \vee \neg B \& C)$;
 - ә) $A \& \neg(B \vee C)$.

С ДЕҢГЕЙІ

- Математикалық өрнекті логикалық өрнек түрінде жазыңдар:
 - $X > 0 \ \& \ X < 1 \ \vee \ X < 10 \ \& \ X < 5;$
 - $X \diamond Y \ \& \ Y \diamond Z.$
- Логикалық өрнектің мәнін анықтаңдар $\neg X \ \& \ Y \ \vee \ X \ \& \ Z$, егер логикалық айнымалы келесі мәндерге ие болса: $X = \text{ЖАЛҒАН}$, $Y = \text{АҚИҚАТ}$, $Z = \text{АҚИҚАТ}$.

Рефлексия:

- Қандай ақпарат қызықтырды?
- Қандай қиындық туындады және оны кіммен талқылағын келеді?
- Қазір қандай дағдыны қолдана аласыңдар?

§12 Практикум. Логикалық операциялар. Ақиқат кестесін құру

Бүгінгі сабақта:

Тірек ұғымдар:

- логикалық операцияларды (дизъюнкция, конъюнкция, инверсия) қолдануды;
- берілген логикалық өрнекке ақиқат кестесін құруды үйренесіңдер.

- дизъюнкция;
- конъюнкция;
- инверсия.

1-нұсқа

- Қандай логикалық өрнек $\neg(A \vee \neg B) \vee \neg C$ өрнегіне тең екенін көрсетіңдер:
 - $A \wedge B \vee \neg C;$
 - $\neg A \wedge \neg B \vee \neg C;$
 - $A \wedge \neg B \wedge \neg C;$
 - $A \wedge (\neg B \vee \neg C).$
- Қандай логикалық өрнек $\neg(A \vee \neg B) \vee \neg C$ өрнегіне тең екенін көрсетіңдер:

X	Y	Z	F
0	1	1	0
1	0	0	1
0	0	1	1

F қандай өрнекке сәйкес келеді?

- $X \wedge \neg Y \wedge Z;$
- $(X \wedge \neg Y) \vee \neg Z;$
- $X \wedge \neg Y \vee Z;$
- $X \wedge \neg Y \vee \neg Z.$

3. $A \wedge \neg B \wedge \neg C$ өрнегі үшін ақиқат кестесін жасаңдар.
4. Ақиқат кестесімен берілген логикалық функциялар үшін өрнектерді құрыңдар. Өртүрлі әдістерді қолданыңдар және оларды салыстырыңдар.

а)

A	B	X
0	0	0
0	1	1
1	0	0
1	1	1

ә)

A	B	X
0	0	10
0	1	0
1	0	1
1	1	1

2-нұсқа

1. Қандай логикалық өрнек $A \wedge (\bar{B} \vee C)$ өрнегіне тең екенін көрсетіңдер:
- а) $A \wedge \bar{B} \wedge C$;
 ә) $A \wedge \bar{B} \vee C \wedge A$;
 б) $A \wedge \bar{B} \vee C$;
 в) $A \wedge B \vee A \wedge \bar{C}$.
2. F символымен X, Y, Z үш аргументтен төменде көрсетілген логикалық өрнектердің бірі белгіленген. F өрнегінің ақиқат кестесінің үзіндісі берілген:

X	Y	Z	F
0	0	1	1
0	1	0	0
1	0	0	1

F -ке қандай өрнекке сәйкес келеді?

- а) $X \vee Y \vee Z$;
 ә) $X \wedge \bar{Y} \wedge Z$;
 б) $X \vee \bar{Y} \vee Z$;
 в) $\bar{X} \wedge Y \vee \bar{Z}$.
3. $A \wedge \bar{B} \wedge C$ өрнегі үшін ақиқат кестесін құрыңдар.
4. Ақиқат кестелерімен берілген логикалық функциялар үшін өрнектерді құрыңдар. Өртүрлі әдістерді қолданыңдар және оларды салыстырыңдар.

а)

A	B	X
0	0	0
0	1	1
1	0	0
1	1	0

ә)

A	B	X
0	0	0
0	1	0
1	0	1
1	1	0

БАҚЫЛАУ СҰРАҚТАРЫ



1. Ақиқат кестесі дегеніміз не?
2. Ақиқат кестесіндегі айнымалының мәндері, әдетте, қандай ретпен жазылады? Ол не үшін қажет?

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

§13 Компьютердің логикалық элементтері

Бүгінгі сабақта:

- ▶ маңызды логикалық элементтердің мақсатын түсіндіру: конъюнктор, дизъюнктор, инвертор.

Тірек ұғымдар:

- ▶ компьютердің логикалық элементтері;
- ▶ триггер;
- ▶ сумматор.

Компьютердің аппараттық құралдарының жұмыс істеуін сипаттауға логика алгебрасының математикалық аппараты өте ыңғайлы, себебі компьютерде негізгі есептеу жүйесі — екілік жүйе, ол 1 және 0 цифрларын қолданады, ал логикалық айнымалылардың мәндері де екеу: 1 (true) және 0 (false).

Осыдан екі қорытынды шығады:

— компьютердің бір құрылғылары екілік санау жүйесінде берілген сандық ақпаратты, сонымен қатар логикалық айнымалыларды өңдеу мен сақтау үшін қолданылуы мүмкін;

— аппараттық құралдарды құрастыру кезеңінде логика алгебрасы компьютер сұлбаларының жұмыс істеуін сипаттайтын логикалық функцияларды анағұрлым ықшамдауға мүмкіндік береді, олай болса, компьютердің негізгі түйіндерінен тұратын элементар логикалық элементтердің санын азайтуға мүмкіндік береді.



Компьютердің логикалық элементі — элементар логикалық функцияны жүзеге асыратын электронды логикалық сұлбаның бөлігі.

Компьютердің логикалық элементтері **ЖӘНЕ**, **НЕМЕСЕ**, **ЕМЕС**, **ЖӘНЕ—ЕМЕС**, **НЕМЕСЕ—ЕМЕС** және басқа (оларды тағы *вентиль*

деп атайды), сонымен қатар *триггер* болады. Бұл сұлбалардың көмегімен компьютер құрылғыларының жұмысын сипаттайтын кез келген логикалық функцияны жүзеге асыруға болады. Әдетте, вентильдің екіден сегізге дейін кірісі, бір немесе екі шығысы болады. Вентильден екі логикалық күйді — «1» және «0» көрсету үшін оларға сәйкес кіріс және шығыс сигналдары кернеудің екі белгіленген деңгейіне ие болады. Мысалы: 5 және 0 В (13.1-сурет).



13.1-сурет. Компьютердің логикалық элементтері

Жоғары деңгей, әдетте, «ақиқат», ал төменгі деңгей «жалған» мәндеріне сәйкес болады.

Әрбір логикалық элементтің шартты белгісі болады, ол оның логикалық функциясын көрсетеді, бірақ қандай электронды сұлба жүзеге асырылғаны көрсетілмейді. Ол күрделі логикалық сұлбаларды жазуды және түсінуді жеңілдетеді.

Логикалық элементтердің жұмысын ақиқаттық кестесі арқылы сипаттауға болады. Компьютердің логикалық элементтерінің құрылымдық сұлбасын және олардың ақиқат кестесін қарастырайық (13.1-кесте).

13.1-кесте

Компьютердің логикалық элементтерінің құрылымдық сұлбасы және олардың ақиқат кестесі

Шартты белгіленуі	Құрылымдық сұлба	Ақиқаттық кестесі															
1	2	3															
ЖӘНЕ		<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>$x \& y$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	y	$x \& y$	0	0	0	0	1	0	1	0	0	1	1	1
x	y	$x \& y$															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
НЕМЕСЕ		<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>$x \vee y$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	y	$x \vee y$	0	0	0	0	1	1	1	0	1	1	1	1
x	y	$x \vee y$															
0	0	0															
0	1	1															
1	0	1															
1	1	1															

1	2	3															
ЕМЕС		<table border="1"> <thead> <tr> <th>x</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </tbody> </table>	x	y	0	0	0	1									
x	y																
0	0																
0	1																
ЖӘНЕ-ЕМЕС		<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>$x \& y$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	x	y	$x \& y$	0	0	1	0	1	1	1	0	1	1	1	0
x	y	$x \& y$															
0	0	1															
0	1	1															
1	0	1															
1	1	0															
НЕМЕСЕ-ЕМЕС		<table border="1"> <thead> <tr> <th>x</th> <th>y</th> <th>$x \& y$</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	x	y	$x \& y$	0	0	1	0	1	0	1	0	0	1	1	1
x	y	$x \& y$															
0	0	1															
0	1	0															
1	0	0															
1	1	1															

ЖӘНЕ сұлбасы екі немесе одан да көп логикалық мәндердің конъюнкциясын жүзеге асырады. Барлық кірісінде бір болғанда ғана **ЖӘНЕ** сұлбасының шығысында 1 болады. Егер бір кірісінде 0 болса, онда шығысында 0 болады.

НЕМЕСЕ сұлбасы екі немесе одан да көп логикалық мәндердің дизъюнкциясын орындайды. **НЕМЕСЕ** сұлбасының ең болмағанда бір кірісінде бір болса, онда шығысында да бір болады.

ЕМЕС сұлбасы (инвертор) терістеу операциясын орындайды. Бұл сұлбаның x кірісі мен z шығысы арасындағы байланысты $z = \bar{x}$, қатынасымен жазуға болады, мұндағы \bar{x} « x емес» немесе «инверсия x » дегенді білдіреді.

ЖӘНЕ—ЕМЕС сұлбасы **ЖӘНЕ** элементі және инвертордан тұрады және **ЖӘНЕ** сұлбасы нәтижесін терістеуді орындайды. Сұлбаның z шығысы мен x және y кірістері арасындағы байланыс келесі түрде жазылады: $z = \overline{x \cdot y}$, мұндағы $\overline{x \cdot y}$ «инверсия x және y » деп оқылады.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Логика алгебрасының пайда болуы мен дербес компьютерді әзірлеудің байланысы бар ма?
2. Негізгі логикалық операцияларды атаңдар.
3. Логикалық пікір болмайтын сөйлемдерге мысал келтіріңдер.
4. Логика алгебрасы мен ақпаратты екілік кодтау арасында байланысты көрсетіңдер.
5. Логикалық операциялардың басымдылығын атаңдар.

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

x, y, z — логикалық шамалар мынадай мәндерге ие: $x =$ ақиқат, $y =$ жалған, $z =$ ақиқат.

а) $\neg x \vee y$; ө) $x \vee y \& z$ өрнектерінің мәнін есептеп логикалық кесте сызыңдар.

В ДЕҢГЕЙІ

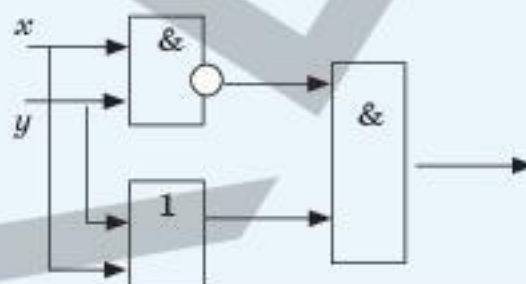
Келесі формулаларды қолданып логикалық сұлба құрыңдар:

а) $x \& (\neg y \vee z)$;

ө) $x \& y \vee \neg x \& z$.

С ДЕҢГЕЙІ

Логикалық өрнектің көмегімен логикалық сұлбаның ақиқат кестесін құрыңдар:



Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§14 Компьютердің логикалық негіздері

Бүгінгі сабақта:

- ▶ процессордың бір бөлігі ретінде басқару құрылғысының (БҚ) арифметика-логикалық құрылғының (АЛҚ) және жад регистрінің функциясын сипаттауды үйренесіңдер.

Тірек ұғымдар:

- ▶ басқару құрылғысы (БҚ);
- ▶ есте сақтау құрылғысы (ЕСҚ);
- ▶ арифметика-логикалық құрылғы (АЛҚ).

Қазіргі таңда өмірімізді дербес компьютерсіз елестету мүмкін емес. Компьютерді барлық жерде кездестіре аласыңдар: мектепте, ауруханада, банкте, кез келген кәсіпорында, әуежайда т.б. Адамзаттың басқа өнертабыстарымен салыстырғанда компьютердің ашылу тарихы

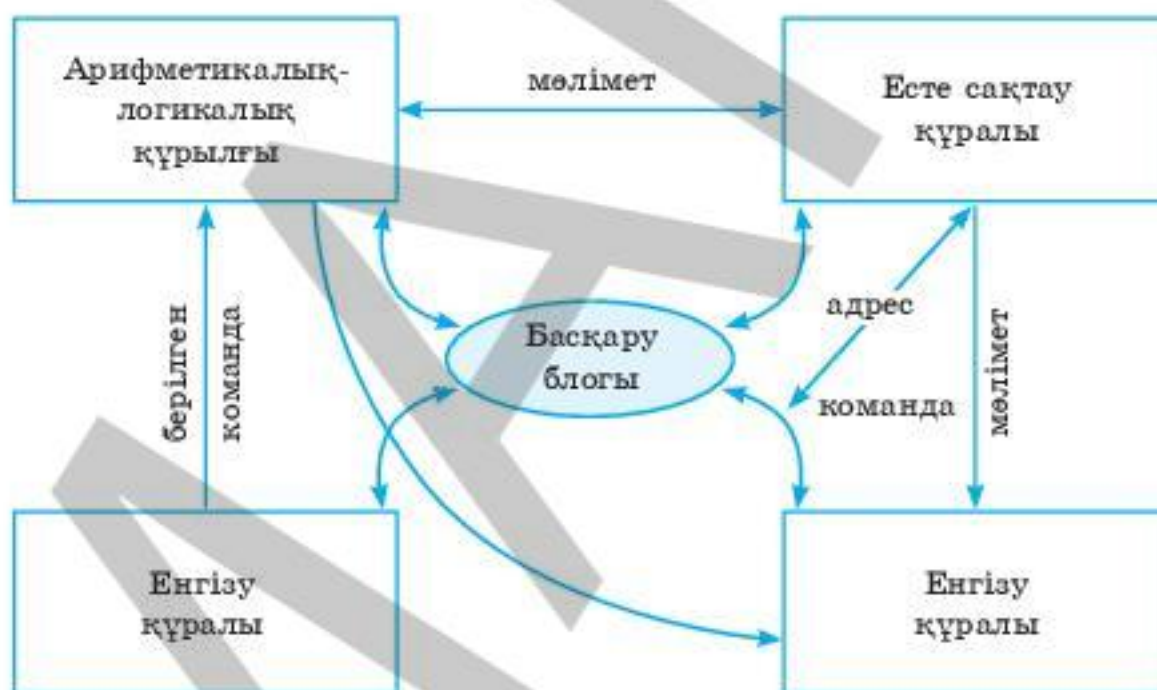
соншалықты ұзақ емес. 1940 жылдардың басында қазіргі заманғы сәулет есептеуіш машинасын фон Нейманның принциптерінің негізінде жасай бастады. Сол уақыттан бері компьютер жылдам дами бастады. Әсіресе электрониканың, интегралды схемалардың жаппай өндірілуінің арқасында компьютер 50-жылдарда бүкіл залды алып жататын үлкен машинадан үстелге ыңғайлы орналастырылған немесе тіпті тасымалдануға қолайлы өте ықшамды құрылғыға айналды.

Компьютерлердің әр жаңа буыны (ал олардың алты түрі ерекшеленді) жұмыс жылдамдығы мен сенімділігі, енгізу және басып шығару құрылғыларын жетілдірді.

Әдетте, компьютердің сәулетін сипаттағанда, ғалымдар оның негізгі принциптері машиналардың көпшілігіне байланысты принциптеріне ерекше назар аударады, сондай-ақ бағдарламалау мүмкіндіктеріне әсер етеді.

Компьютердің сәулеті оған бағдарламалау мүмкіндігіне байланысты, сондықтан ДК сәулетін сипаттағанда командалар мен жады сипаттамаларына назар аударылады.

Фон Нейман машинасының сәулетін қарастырайық. Фон Нейман машинасы есте сақтау құрылғыдан — жады, арифметикалық логикалық құрылғыдан — АЛҚ, басқару құрылғысы — БҚ, сондай-ақ басып шығару және енгізу құрылғыларынан тұрады (14.1-сурет).



14.1-сурет. Фон Нейманның есептеу машинасының сұлбасы

Бағдарламалар мен деректер кіріс құрылғысынан арифметикалық-логикалық құрылғы арқылы жадыға енгізіледі. Барлық бағдарлама нұсқаулары көрші жады ұяшықтарына жазылады және өңдеуге арналған деректер ерікті ұяшықтарда болуы мүмкін. Кез келген бағдарлама үшін соңғы команда өшіру пәрмені болуы керек.

Команда қандай әрекетті орындау керектігін және көрсетілген операцияны орындау үшін деректер сақталатын жады модулінің мекенжайларын, сондай-ақ нәтиже жазылуы қажет ұяшықтың мекенжайын көрсетеді.

Арифметикалық-логикалық құрылғы көрсетілген деректер бойынша пәрмендермен көрсетілген әрекеттерді орындайды. Арифметикалық-логикалық блоктан нәтижелер жадыға немесе шығыс құрылғысына шығарылады. Жад құрылғысы мен шығыс құрылғысы арасындағы негізгі айырмашылық мынада: бұл деректер жадта компьютерді өңдеуге ыңғайлы пішінде сақталады және шығыс құрылғылары (принтер, монитор және т.б.) қолданушыға ыңғайлы болатындай жасалады.

Басқару құрылғысы (БҚ) компьютердің барлық бөліктерін басқарады. Басқару құрылғысынан басқа құрылғыларға «Не істеу керек?» сигналдары кіреді және басқа құрылғылардан БҚ олардың жағдайы туралы ақпарат алады.

Басқару құрылғысында «командалық есептегіш» деп аталатын арнайы тіркеуі (ұяшық) бар. Бағдарлама мен деректерді жадыға жүктегеннен кейін бағдарламаның бірінші командасының мекенжайы бағдарлама санауышына жазылады. Жад картасы еске сақтау орнының мазмұнын еске түсіреді, оның мекенжайы нұсқаулық есептегіші болып табылады және оны арнайы құрылғыға — «Командалық тіркелімге» орналастырады. БҚ пәрменнің жұмысын анықтайды, мекенжайлары пәрменде көрсетілетін деректерге жадында «ескертулер» береді және пәрменнің орындалуын басқарады. Операция АЛҚ немесе компьютерлік құрылғылар арқылы орындалады.

Нұсқаушы кез келген команданы орындау нәтижесінде біреуіне өзгертіледі, демек, бағдарламаның келесі әрекетін көрсетеді.

Заманауи компьютерлерде БҚ және АЛҚ функциялары орталық процессор деп аталатын бір құрылғыны орындайды. *Орталық процессордың* принциптері келесі бөлімде қарастырылады.

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

1. Paint графикалық редакторында фон Нейман есептеуіш машинасының сұлбасын құрастырындар.
2. «Джон фон Нейманның ғылымға қосқан үлесі» жайлы эссе жазындар.

В ДЕҢГЕЙІ

1. MS Word бағдарламасында фон Нейман есептеуіш машинасының сұлбасын құрастырыңдар.
2. «Гарвард архитектурасы» туралы мәліметтер дайындаңдар.

С ДЕҢГЕЙІ

1. Кез келген бағдарламада фон Нейман есептеуіш машинасының сұлбасын құрастырыңдар.
2. Джон фон Нейман компьютердің «фон Нейман» сәулетінің жалғыз авторы емес екенін растайтын фактілерді көрсететін хабар дайындаңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§15 Практикум. Логикалық есептерді шешу*Бүгінгі сабақта:*

- ▶ логикалық операцияларды (дизъюнкция, конъюнкция, инверсия) қолдануды;
- ▶ берілген логикалық өрнекке ақиқаттық кестесін құруды үйренесіңдер.

Тірек ұғымдар:

- ▶ дизъюнкция;
- ▶ конъюнкция;
- ▶ инверсия.

Негізгі берілген есептер мен пайымдауларды сұлбалар мен кестелер түрінде ұсынуға бағытталған міндеттерді шешу маңызды, олар ақпараттың көрнекі графикалық ұстанымы болып табылады, есептерді шешу процесін жылдамдатады және жеңілдетеді.

Сұлбалардың көмегімен тапсырмаларды шешудің мұндай реті ұсынылады: шартты, тапсырма сұрағын қысқаша жазып, оны шешуге кірісу. Тапсырма шартты элементтері символдық айнымалылармен көрсетіледі. Егер екі элемент арасындағы шарт бойынша сәйкестік болса, онда олар тұтас сызықпен қосылады. Егер сәйкестік элементтері арасында болмаса, олар нүктелік сызықпен қосылады.

Кестелердің көмегімен төрт, бес және одан да көп элементтер жұбы бар есептер шешіледі.

1-тапсырма. Мектеп мұғалімдері

Жоғарғы сыныптарда үш мұғалім — Өубәкіров, Ғалиев және Семенов жұмыс істейді. Олардың әрбіреуі екі пән оқытады, яғни олардың кестелерінде алты пән жазылған: математика, физика, химия, тарих, әдебиет және ағылшын тілі. Семенов — оқытушылардың ең жасы. Химия пәні мұғалімінің тарих пәні мұғалімінен жасы үлкен. Олардың үшеуі — химия пәнінің мұғалімі, физика пәнінің мұғалімі және Ғалиев спортпен шұғылданады. Әдебиет және ағылшын тілі мұғалімдерінің арасында дау туындаған кезде Семенов та оған қатысады. Ғалиев ағылшын тілін де, математиканы да оқытпайды.

Кім қандай пәндерді оқытады?

Берілгені:

Өубәкіров
Ғалиев
Семенов

Талқылау:

Есептің шарты бойынша химия, физика пәндерінің мұғалімі және Ғалиев спортпен шұғылданады. Демек, Ғалиев химия және физика мұғалімі емес. «Химия, Ғалиев» және «Физика, Ғалиев» ұяшықтарына минус қоямыз.

Пәндер:

математика
физика
химия
тарих
әдебиет
ағылшын тілі

Есептің шарты бойынша әдебиет және ағылшын тілі мұғалімдерінің арасында дау туындаған кезде Семенов те оған қатысады. Демек Семенов әдебиет және ағылшын тілі мұғалімі емес. «Әдебиет, Семенов» және «ағылшын тілі, Семенов» ұяшықтарына минус қоямыз.

Анықтау:

Кім қандай пәндерді оқытады?

Есептің шарты бойынша Ғалиев ағылшын тілін де, математиканы да оқытпайды. «Ағылшын тілі, Ғалиев» және «математика, Ғалиев» ұяшықтарына минус қоямыз. Есептің шарты бойынша Семенов — оқытушылардың ең жасы. Химия пәнінің мұғалімі тарих пәнінің мұғалімінен жасы үлкен. Семенов химия мұғалімі емес, «Химия, Семенов» ұяшығына минус қоямыз.

Сонымен есептің шарты бойынша келесі 15.1-кестені аламыз.

15.1-кесте

	Математика	Физика	Химия	Тарих	Әдебиет	Ағылшын тілі
Өубәкіров						
Ғалиев	—	—	—			—
Семенов			—		—	—

Есеп шартынан әрбір мұғалім екі пәнді оқытатыны белгілі. Кестеден Ғалиев тарих пен әдебиетті, химия мен ағылшын тілін Өубәкіров оқытатынын көруге болады. «Тарих, Ғалиев», «әдебиет, Ғалиев», «химия, Өубәкіров», «ағылшын тілі, Өубәкіров» ұяшықтарына «+» қоямыз (15.2-кесте).

15.2-кесте

	Математика	Физика	Химия	Тарих	Әдебиет	Ағылшын тілі
Әубәкіров			+			+
Ғалиев	—	—	—	+	+	—
Семенов			—		—	—

«Тарих» және «әдебиет» бағандарындағы және «Әубәкіров» жолындағы бос ұяшықтарды минустармен толтырамыз, яғни Ғалиев тарих пен әдебиетті (дәлелдеме бойынша) оқытады, демек, Әубәкіров пен Семенов тарих пен әдебиетті оқытпайды, себебі Әубәкіров химия мен ағылшын тілін (дәлелдеме бойынша) оқытады, демек, ол басқа пәндерді оқытпайды (15.3-кесте).

15.3-кесте

	Математика	Физика	Химия	Тарих	Әдебиет	Ағылшын тілі
Әубәкіров	—	—	+	—	—	+
Ғалиев	—	—	—	+	+	—
Семенов			—	—	—	—

Кестеден Семенов химия, тарих, әдебиет, ағылшын тілін оқытпайтынын көреміз, яғни Семенов — физика және математика мұғалімі. «Математика, Семенов» және «физика, Семенов» ұяшықтарына «+» қоямыз (15.4-кесте).

Соңында:

15.4-кесте

	Математика	Физика	Химия	Тарих	Әдебиет	Ағылшын тілі
Әубәкіров	—	—	+	—	—	+
Ғалиев	—	—	—	+	+	—
Семенов	+	+	—	—	—	—

Жауабы: Семенов физика және математика, Ғалиев — тарих және әдебиет, Әубәкіров — химия және ағылшын тілін оқытады.

2-тапсырма. Ит шаруашылығы мектебі.

Достар ит өсіру мектебінде өз иттерін — Джек, Лесси және Грифті жаттықтырып, көп ұзамай жарысқа қатысты.

Төрешілердің бірі жарыс нәтижелері туралы достардың сұрағына жауап берді:

«Джек екінші орын алды.

Лесси, менің ойымша, екінші емес.

Гриф бірінші болған жоқ».

Нәтижелерді жариялағаннан кейін төреші екі рет қателесіп, бір рет дұрыс айтты.

Егер барлық қатысушылар әртүрлі орындарға ие болса, жүлделі орындар қалай бөлінді?

Шешуі.

Берілгені:

Иттер:

Джек (Д)

Лесси (Л)

Гриф (Г)

Орындар:

1, 2, 3

Анықтау:

Кім 1-орынды иеленді,

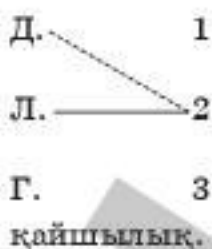
2-орын, 3-орын?

Талқылау:

Джек 2-орынға ие болғаны туралы шынайы пікір болса, онда Лесси 2-орынға ие емес, жалған, яғни Лесси 2-орынға ие болды. Джек пен Лесси 2-орынды бөлісе алмайтындықтан, есеп шартымен қарама-қайшылық алды. Демек, біздің бірінші пікір ақиқат деген болжамымыз, дұрыс емес.

Графикалық көрсету.

Джек 2-орын алғаны туралы ақиқат пікір болсын.



2-нұсқа

«Лесси 2-орын алған жоқ» жауабы ақиқат болсын, онда «Гриф бірінші болған жоқ» — жалған, демек, Гриф 1-орын алды.

Себебі (дәлелдеме бойынша) Гриф 1-орынға ие болды, яғни Лесси мен Джек 1-орынға ие болған жоқ.

Лесси 2-орынды да (болжам бойынша) және 1-орынды да алған жоқ (дәлел бойынша), Лесси 3-орынға ие болды.

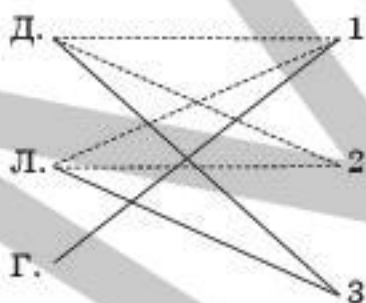
Джек 2-орынға ие деген жалған (болжам бойынша), яғни Джек 2-орынға ие емес, бірақ ол 1-орынға ие емес (дәлел бойынша), яғни Джек 3-орынға ие болды.

Ал Джек пен Лесси 3-орынды бөлісе алмады.

Демек, екінші пікір шынайы деген дұрыс емес.

Графикалық көрсету.

Лесси 2-орынға ие болмағаны туралы ақиқат пікір болсын.



қайшылық

3-нұсқа

Грифтің бірінші болмағаны шынайы сөз болсын, онда Джек 2-орынға ие болды деген сөз жалған, яғни Джек 2-орынға ие болмады.

Өйткені Лесси 2-орынды алмады деген жалған (болжам бойынша), яғни Лесси 2-орынға ие болды.

Лесси 2-орынға ие болғандықтан (дәлелдеме бойынша), Гриф 2-орынға ие болған жоқ.

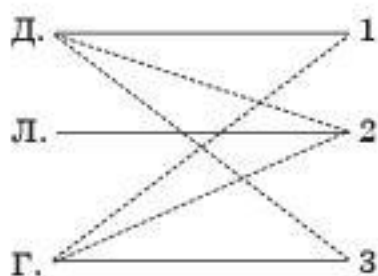
Себебі Гриф 1-орын емес (болжам бойынша) және 2-орын емес (дәлел бойынша), демек, Гриф 3-орын алды.

Гриф 3-орынға ие болғандықтан, Джек 3-орынға ие болмады.

Джек 2- және 3-орындарды (дәлелдеме бойынша) алмайтындықтан, Джек 1-орынды иеленді.

Графикалық иллюстрация.

Грифтің бірінші болмағаны туралы шынайы пікір болсын.



дұрыс болжам.

Белгілі бір өрнекке вентиль кестесін құру. Вентиль арифметикалық амалдарды орындауға және ақпаратты сақтауға мүмкіндік беретін күрделі схемалардан тұрады. Сонымен қатар белгілі бір функцияларды жүзеге асыратын схема өртүрлі комбинациялар мен вентильдер санын құруға болады. Логикалық тізбектер элементтердің барынша аз санынан құрылуы керек, бұл өз кезегінде жұмыс жылдамдығының жоғары болуын қамтамасыз етеді және құрылғының сенімділігін арттырады.

Логикалық кесте құру ережесі:

1. Логикалық айнымалылардың санын анықтау.
2. Негізгі логикалық операциялардың санын және олардың тәртібін анықтаңдар.

3. Әр логикалық операция үшін тиісті вентиль (клапан) салыңдар және вентильді логикалық операция тәртібіне қосыңдар.

3-есеп. Екі қабат үйде баспалдақ бір X шамымен жарықталған. Бірінші қабатта бір A қосқышы орнатылған, ал екінші қабатта B қосқышы бар. Егер A қосылса, шам жанып тұрады. Екінші қабатқа көтерілгенде және қосылғанда B шамы өшеді. Егер біреу сыртқа шығып B түймесін басса, онда шам қосылады және бірінші қабатқа түсіп A батырмасын басқанда шам өшіп қалуы керек.

Есепті шешу алгоритмі:

- ақиқат кестесін құрыңдар;
- логикалық функцияны анықтаңдар;
- логикалық кесте құрыңдар.

A	B	X
0	0	0
1	0	1
1	1	0
0	1	1
0	0	0

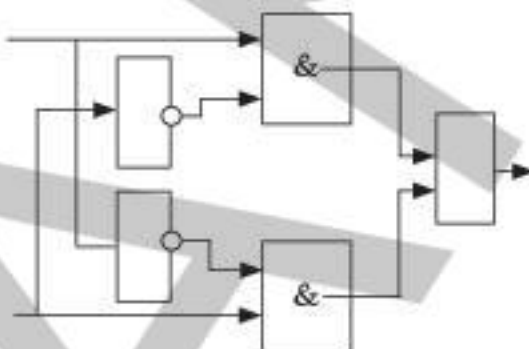
Ақиқат кестесіне сәйкес логикалық функция жасау үшін шығыс айнымалы мәндерін жазу қажет.

Әрбір айнымалы жиынтығы үшін функция логикалық 1-ді қабылдайтын болса, дизъюнкциямен біріктірілген конъюнкциялар жазылады.

Әрбір 0 логикалық мәні бар әрбір жолдың айнымалысы конъюнкцияда теріс болып шығады, ал 1 логикалық мәні бар айнымалы теріс емес болады.

$$(A \& \neg B) \vee (\neg A \& B).$$

Функцияға сәйкес тізбек құрайық:



БАҚЫЛАУ СҰРАҚТАРЫ



1. Формалды логика «әдеттегі», «тұрмыстық» логикадан немен ерекшеленеді?
2. Табиғи әрекет тәртібін өзгерту үшін не істеуге болады?

Рефлексия:

- Қандай ақпарат қызықтырды?
- Қандай қиындық туындады және оны кіммен талқылағын келеді?
- Қазір қандай дағдыны қолдана аласыңдар?

§16

Мәтіндік ақпаратты кодтау принциптері

Бүгінгі сабақта:

- Unicode және ASCII символдарын кодтау кестесін салыстыруды үйренесіңдер.

Тірек ұғымдар:

- кодтау;
- код;
- бит.

Бір ақпарат өртүрлі тәсілдермен ұсынылуы мүмкін. Өр халықтың өзінің тілі бар: олар символдар (өріптер) жиынтығы: қазақ, орыс, ағылшын және т.б. Мысалы, ежелгі адамдар барабан дыбыстарымен (қатты, ақырын) алысқа кодталған хабарлар таратқан. Кейін ақпарат телеграфта Морзе кодының көмегімен, яғни нүктелер мен сызықшалар түрінде таралды (16.1-сурет).

А	·-	П	···	Б	···
Б	···	Р	·-	Ы	·-
В	·-	С	··	Й	···
Г	··	Т	-		
Д	··	У	···	1	···
Е	·	Ф	···	2	···
Ж	···	Х	···	3	···
З	··	Ц	·-	4	···
И	··	Ч	··	5	···
К	···	Ш	···	6	···
Л	···	Щ	···	7	···
М	··	Э	···	8	···
Н	·-	Ю	···	9	···
О	—	Я	···	0	···

16.1-сурет. Морзе коды

Ақпаратты қандай да бір тілдердің көмегімен ұсыну *кодтау* деп аталады. Өр халықтың ақпаратты ұсыну, кодтау түрлері де өртүрлі, сондықтан өр халық өртүрлі тілде сөйлеседі. Ақпаратты сурет, таңба,

өріп түрінде ақпарат тасушыларда (қағаз, тастарда, компьютерде т.б.) беру — *жазбаша кодтау* деп аталады.



Кодтау — ақпаратты код түрінде беру процесі.

Код — ақпарат ұсыну үшін берілетін таңбалар жиынтығы (таңбалар).



Күнделікті өмірде ақпаратты кодтаудың сан алуан түрін кездестіресіңдер: жүргізушілер мен жаяу жүргіншілер үшін жол қозғалысының ережесі бағдаршам арқылы және жол белгілерімен, мұғалімнің айтқан ақпаратын дәптерге өріптер жинағы ретінде кодталады.

XVII ғасырда неміс ғалымы Готфрид Вильгельм Лейбниц ұсынған кодтаудың арқасында компьютерде кез келген ақпаратты өңдеуге болады. Сандық, мәтіндік, графикалық, аудио, бейне және т.б. ақпараттардың барлығы өңделгеннен кейін тізбектелген электр импульстары түрінде беріледі, яғни импульстің (сигналдың) болуы — бір, ал болмауы — нөлмен кодталады (16.2-сурет).



16.2-сурет. Ақпаратты кодтау

Бүгінде 0 және 1 таңбаларынан тұратын ақпаратты кодтау жолы қолданылады. 0 және 1 таңбалары *бит* деп аталады (ағылшынша binary digit — екілік таңба).



Бит ақпаратты өлшейтін ең кіші бірлік болып табылады және екілік санымен көрсетіледі. Ақпараттың көлемі үлкен болса, оны байтпен өлшейді, яғни құрайды. 1 байт = 8 бит.

БҰЛ ҚЫЗЫҚ!

1946 жылы алғаш рет Принстон университетінің математигі Джон Таки өзінің мақалаларының бірінде «бит» терминін қолданды.





Мәтіндік ақпаратты кодтау. Компьютерге енгізілген мәтінді кодтау үшін әр таңбаға екілік сан беріліп, қарапайым кодтау әдісі қолданылады. Демек, пернетақтадағы пернені басқанда сигнал компьютерге екілік сан ретінде жіберіледі. Әрбір таңбаны кодтау үшін талап етілетін ақпарат көлемі 8 бит болады, яғни таңбаның екілік кодының ұзындығы сегіз екілік сандар. Мысалы, «1» пернесін басқан кезде 00110001 екілік кодын және «2» пернесін басқан кезде 00110010 кодын аласыңдар. Әрбір екілік кодтау кезінде таңбаға 00000000-ден 11111111 дейінгі аралықта код беріледі (ондық кодтауда 255-ге дейін) (16.1-кесте).

16.1-кесте

Таңбаны кодтау

Екілік код	Ондық код	КОИ-8	Windows	Mac	ISO
00000000	0				
...			
00001000	8	Backspace пернесі			
...			
00001101	13	Enter пернесі			
...			
00100000	32	Бос орын пернесі			
...			
01011010	90	Z			
...			
10000000	128	-	Ъ	А	к
...
11001100	204	л	М		Ь
...
11111111	255	ь	я	бөлінбей- тін бос орын	п

Адам таңбаларды пішіні арқылы, ал компьютер екілік кодтар бойынша ажыратады. Демек, компьютер экранында — символдар, ал компьютердің жадында — екілік кодтар.

Компьютер экранында символды көрсету процесінде керісінше кодтау орындалады, яғни символдың екілік кодын кескінге түрлендіру.

Қай пернені басылғанына байланысты сәйкес екілік код алынады. Сәйкестік ережелері немесе кодтау ережелері *код* деп аталатын кестеге жазылады.



Код кестелері — алфавит таңбалары мен екілік сандар арасында сәйкестікті орнататын кесте. Бұл сандар таңбалар коды деп аталады және компьютердегі таңбалардың ішкі көрінісіне сәйкес келеді.

Әртүрлі таңбаларды кодтау. Код кестесіндегі алғашқы 33 код (ондық кодтар 0-ден 32-ге дейін) операцияларға (сызық арна, символды алып тастау және т.б.) сәйкес келеді. 33-тен 127-ге дейінгі ондық кодтар латын алфавитінің таңбаларына, сандарына, арифметикалық амалдар белгілеріне және тыныс белгілеріне сәйкес келеді.

128-ден 255-ке дейінгі ондық кодтар ұлттық, яғни түрлі ұлттық код жүйесінде бір кодқа әртүрлі белгілер сәйкес келеді. Бүгінгі күні орыс әріптеріне арналған 5 түрлі код кестелері (KOI-8, MS-DOS, Windows, Mac, ISO) бар.

Соңғы жылдары *Unicode* мәтіндік символдарды кодтаудың жаңа халықаралық стандарты кеңінен таралған.

Осылайша қазіргі уақытта бірдей символдардың алты түрлі кодтау белгісі бар (16.2-кесте).

16.2-кесте

Әртүрлі кодтаудағы кейбір таңбалардың ондық кодтары

Таңба	KOI-8	Windows	Mac	ISO	MS-DOS	Unicode
А	225	192	128	176	128	1040
В	247	194	130	178	130	1042
Э	252	221	157	205	157	1069
я	241	255	223	239	239	1103

Операциялық жүйеге және қосымшаларға орнатылған арнайы конверттік бағдарламаларда мәтіндік құжаттарды қайта кодтауға болады.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Код деген не, кодтаудың ақпараты қандай?
2. Өмірден шифрлеуге қатысты бірнеше мысалдар келтіріңдер. Әр мысалдан қандай ережелер шығарылады?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Мәтіндік режимде компьютер мониторының экраны, әдетте, жолда 80 таңбадан 25 жолға бөлінеді. Unicode кодында монитор экранындағы мәтін ақпаратының көлемін анықтаңдар.

В ДЕҢГЕЙІ

Unicode 16 биттік кодтау арқылы кодталған келесі сөздің ақпараттық көлемі қандай екенін анықтаңдар.

С ДЕҢГЕЙІ

Орыс тіліндегі хабарламаны 16 биттік Unicode кодынан KOI8-R 8 биттікке кодтау кезінде ол 480 битке азайды. Таңбалардағы хабарламаның ұзындығы қандай?

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

Өзіңді тексер!

1. Бір компания құлыпқа жабылатын бөлмеде деректер қорының мұрағаттық және басқа серверлік ақпараттың көшірмесін сақтайды. Есіктегі құлып — ол:
 - а) ақпаратты сақтаудың әкімшілік құралы;
 - ә) ақпаратты қорғаудың техникалық құралы;
 - б) ақпаратты сақтау құралы болмайды;
 - в) ақпаратты сақтаудың кең тараған құралы;
 - г) ақпаратты сақтаудың патенттелмеген құралы.
2. Word бағдарламасында «Ақпаратты қорғау» функциясы қандай қызмет атқарады:
 - а) құжатты құпия сөзсіз қарау мен түзетуден қорғайды, бірақ вирустан қорғамайды;
 - ә) құжаттарды құпия сөзсіз өзгерту мүмкіндігін шектейді;

б) $3 + 5 = 2 * 4$;

в) $2 + 6 > 10$;

г) сканер — компьютердің экранында көрсетілген ақпаратты қағазға басып шығаратын құрылғы;

ғ) $II + VI > VIII$;

д) 2 және 6 сандарының қосындысы 8 санынан үлкен;

е) тінтуір — ақпаратты енгізу құрылғысы.

11. Ақиқат пен жалған пікірге екі мысалдан келтіріңдер:

а) биология;

ә) география;

б) информатика;

в) тарих;

г) өдебиет;

ғ) математика;

д) қазақ тілі.

12. Келесі пікірлерге қарсы пікір құрастырыңдар:

а) 1 саны құрама сан болады;

ә) 0-мен аяқталатын натурал сандар жай сандар болады;

б) 3 саны 198 санының бөлгіші болмайтыны дұрыс емес;

в) 4 цифрымен аяқталатын кез келген сан 4 санына бөлінетіні дұрыс емес;

г) кейбір сүтқоректілер құрлықта өмір сүрмейді.

III ТАРАУ

АЛГОРИТМДЕУ ЖӘНЕ БАҒДАРЛАМАЛАУ

Бұл тараудан білесіңдер:

- ▶ функция мен процедураларды;
- ▶ жолдармен, файлдармен жұмыс істеуді;
- ▶ сұрыптау түрлерін: көпіршік әдісі, сызықты іздеу, екілік іздеуді.

Мынаны үйренесіңдер:

- ▶ функциялар мен процедураларды пайдалана отырып, бағдарламау тілінде кодты жазуды;
- ▶ жолдарды өңдеу үшін процедуралар мен функцияларды пайдалануды;
- ▶ ақпаратты оқу және жазу үшін файлдарды пайдалануды;
- ▶ практикалық есептерді шешу үшін графаларда іздеу алгоритмдерін жүзеге асыруды.

§17-18

Пайдаланушы функциялары мен процедуралары

Бүгінгі сабақта:

- ▶ функциялар мен процедураларды пайдалана отырып, бағдарламалау тілінде кодты жазуды үйренесіңдер.

Тірек ұғымдар:

- ▶ процедура ұғымы;
- ▶ функция ұғымы;
- ▶ функция аты;
- ▶ функция денесі;
- ▶ функция түрлері.

«Функциялар» ұғымы сендерге таныс, себебі оны бағдарламалау тілінің стандартты функцияларын бағдарлама барысында қолдандыңдар (мысалы: input, int, randint).



Бағдарламалауда функциялар қажет пе?

Бағдарламалаудағы функциялар бағдарлама барысында бірнеше рет жұмыс істеуге болатын оқшауланған код блогы болып табылады.



Мұндай нұсқаулар блогы не үшін қажет?

Ең алдымен бастапқы код көлемін қысқарту үшін: қайталанатын өрнектерді жеке блокқа шығару әлдеқайда ыңғайлы, содан кейін қажет болған жағдайда оған жүгіну қажет.

Функция — бұл деректермен күрделі әрекеттер жасайтын және кейбір мәнді қайтаратын ішкі бағдарлама. Функцияның нәтижесі сан, жол немесе басқа нысан болуы мүмкін.

Сан цифрларының қосындысын есептеу функциясының мысалын қарастырайық. Бұл жағдайда келесі алгоритм қолданылады (17.1-сурет):

```
sum=0
while n!=0:
    sum+=n%10
    n=n//10
```

17.1-сурет. Сан цифрларының ұзындығын есептеу функциясының мысалы

Санның соңғы цифрын алу үшін (қосындыға қосылатын) санды 10-ға бөлгендегі қалдықты аламыз. Содан кейін соңғы санды алып тастап, келесі цифрға өтеміз. Цикл n мәні нөлге тең болғанға дейін жалғасады.



Функцияның мәні неге тең екені бағдарламада қалай көрсетіледі?

Ол үшін RETURN операторын қолданасындар, содан кейін мән-нәтиже жазылады:



Ағылшын тілінен аударғанда return «қайтару» дегенді білдіреді.

```
def sumDigits( n ):
    sum = 0
    while n != 0:
        sum += n % 10
        n = n // 10
    return sum
# негізгі бағдарлама
print ( sumDigits(12345) )
```

17.2-сурет. Бағдарлама кодының түрі

Ұсынылған бағдарламалық кодты түсіндірейік (17.2-сурет). def — бұл функцияны жасауға мүмкіндік беретін Python бағдарламалау тілінің командасы. sumDigits — бұл функция аты, ол (айнымалылардың аттары сияқты) кез келген сөз болуы мүмкін. Қос нүкте (:) функция тақырыбының аяқталуын білдіреді. Тақырыптан кейін функцияның денесі жазылады. Функция денесінің соңында return командасы бар (ол болмауы да мүмкін). Осы нұсқаулық бағдарламаның негізгі тармағына қандай да бір мәнді береді. Егер return командасы болмаса, онда негізгі бағдарламаға ештеңе қайтарылмайды және n айнымалысына ештеңе меншіктелмейді.



Функцияның түрін өрнектеу бағдарламаның негізгі тармағын шақырған кезде ғана орындалады. Мысалы, егер функция бастапқы кодта болса, бірақ онда еш жерде шақырылмаған болса, онда ондағы нұсқаулар бір рет те орындалмайды.

Бұдан әрі процедураны қарастырайық. Бағдарламаның бірнеше жерінде «бағдарлама қатесі» деген қате туралы хабарды экранға шығару қажет екенін елестетіп көрейік. Мұны, мысалы, осылай жасауға болады:

```
cout << «бағдарлама қатесі»;
```


Бұл шығару операторын қате туралы хабарды шығару қажет болған жерге қоюға болады. Мұнда келесі қиындықтар кездесуі мүмкін. *Біріншіден*, жол — хабарлама жадыда бірнеше рет сақталады. *Екіншіден*, егер хабар мәтінін өзгерту қажет болса, шығару операторларын барлық бағдарлама бойынша іздеу керек.

Осы үшін кейбір әрекеттерді орындайтын көмекші алгоритмдер — процедуралар арналған.

Процедураны пайдалана отырып, бағдарлама фрагментін қарастырайық (17.3-сурет):

```
def Error():
    print ( "бағдарлама қатесі" )
n = int ( input() )
if n < 0:
    Error()
```

17.3-сурет

Процедура (жартылай қалың қаріппен белгіленген) `def` кілт сөзінен басталады. `Error` — процедураның аты, содан кейін бос жақшалар (бірақ олар бос емес болуы мүмкін) және қос нүкте.

Процедура оны шақыру кезінде анықталуы тиіс, яғни жадта нысан процедураны жасайтын `def` нұсқаулығы орындалуы тиіс. Егер процедура негізгі бағдарламадан шақырылатын болса, онда оның анықтамасын шақыру нүктесінен бұрын қою керек.



Процедура тақырыбында және оны шығару кезінде процедура атынан кейін дөңгелек жақшаларды қою керек (біздің мысалымызда олар бос тұрады).

Осылайша егер қандай да бір операциялар бірнеше рет орындалса, процедуралар бағдарлама кодын қысқартады.

Функция мен процедура — бұл аргументтерді қабылдайтын көмекші алгоритмдер. Бірақ, процедураға қарағанда, функция әрқашан мән-нәтиже қайтарады. Сан, таңба немесе басқа түрдегі нысан нәтиже болуы мүмкін.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Функция дегеніміз не? Ол процедурадан немен ерекшеленеді?
2. Бағдарлама мәтінінде функциялар қалай жазылады?
3. Бағдарлама мәтіні бойынша функцияның қандай мәнін қайтарады?
4. Процедураларды пайдаланудың мәні неде?
5. Бағдарлама мәтініне процедураның «жұмыс істеуі» үшін енгізу жеткілікті ме?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

1. Функцияның денесінде үш айнымалының мәндерімен есептеулер жүргізілетін бағдарламаны жазыңдар.
2. Сан цифрларының санын есептейтін функцияны пайдаланып бағдарламаны жазыңдар.

В ДЕҢГЕЙІ

Пайдаланушыдан сан қабылдайтын және бірден енгізілген барлық сандардың орташа арифметикалық санын шығаратын бағдарламаны жазыңдар. Бағдарлама пайдаланушы 0 енгізгенге дейін санды сұрауы тиіс.

С ДЕҢГЕЙІ

1. Жарыста спортшының өнерін 5 сарапшы бағалайды, олардың әрқайсысы бағаны балмен (бүтін сан) қояды. Қорытынды баға алу үшін сарапшылардың бағаларынан ең жақсы және нашар баға алынып тасталады, ал қалған үшеуі үшін орташа арифметикалық баға шығарылады. Сарапшылардың 5 бағасын енгізетін және спортшының көрсетуінің қорытынды бағасын қайтаратын функцияны жазыңдар.
2. Екі функциясы бар бағдарламаны ойлап жазыңдар, бағдарламаның негізгі тармағы кем дегенде 10 жолдан тұрады және әрбір функция кем дегенде бір рет шақырылады.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§19

Пайдаланушы функциялар мен процедуралардың мысалдары

Бүгінгі сабақта:

- ▶ функциялар мен процедураларды пайдалана отырып, бағдарламалау тілінде кодты жазуды үйренесіңдер.

Тірек ұғымдар:

- ▶ «процедура» ұғымы;
- ▶ «функция» ұғымы;
- ▶ функция аты;
- ▶ функция денесі;
- ▶ функция түрлері.

Бағдарламаны функцияларға бөлудің бірнеше себептері бар:

— жаңа функцияны жасау нұсқаулық тобына атау беру мүмкіндігін береді. Бұл бағдарламаны оқуды, түсінуді және баптауды жеңілдетуге мүмкіндік береді;

— функциялар қайталанатын код бөліктерін жою арқылы бағдарлама кодын қысқартуға болады. Кейінірек сендер тек бір жерде түзетулер енгізе аласыңдар;

— ұзын бағдарламаны функцияға бөлу бір уақытта жеке бөліктерді реттеуге, содан кейін оларды біртұтас жинауға мүмкіндік береді;

— жақсы жобаланған функция көптеген бағдарламаларда қолданылуы мүмкін. Функцияны бір рет жазып, оны бірнеше рет пайдалана аласыңдар.

Python ең танымал математикалық функцияларды ұсынатын математикалық модульді (module) қамтиды. Пайдалану алдында оны импорттауымыз керек:

```
> > import math
```

Бұл нұсқаулық `math` деп аталатын модульдік нысанды (module object) жасайды. Егер модульдік нысанды экранға шығарсаңдар, онда ол туралы кейбір ақпаратты алуға болады:

```
> > > print math
<module 'math' (built-in)>
```

Модульдік нысаннан анықталған функциялар мен айнымалыларды қамтиды. Осы функциялардың біріне кіру үшін модуль атауын және нүктемен бөлінген функцияның атауын (period) көрсету қажет. Бұл пішім нүктелік *нотация* деп аталады (dot notation).

```
> > ratio = signal_power / noise_power
> > decibels = 10 * math.log10(ratio)
> > > radians = 0.7
> > > height = math.sin (radians)
```

Бірінші мысалда «сигнал-шу» қатынасының 10 негізі бойынша логарифм есептеледі. `Math` модулі, сондай-ақ `no` негізі бойынша логарифмдерді есептейтін `log` функциясын ұсынады.

Екінші мысалда radians синусы есептеледі. Айнымалының аты sin және басқа да тригонометриялық функциялар (cos, tg және т.б.) радиандарды, аргументтерді қабылдайды. Градусты радианға ауыстыру үшін 360-қа бөліп, 2π -ге көбейтіндер:

```
> > degrees = 45
> > radians = degrees / 360.0 * 2 * math.pi
>>> math.sin (radians)
0.707106781187
```

Math.Pi өрнегі math модулінен Pi айнымалы мәнін алады. Бұл айнымалының мәні 15 таңбаға дейінгі дәлдікке жақын.

Жаңа функцияларды қосу. Бұрын Python ішіне енгізілген функцияларды пайдаландық. Енді жаңа функцияларды қалай қосуға болатынын қарастырайық. Функцияны анықтау (function definition) функция шақырылған кезде орындалатын жаңа функцияның атауын және нұсқаулардың ретін анықтайды. Функцияны анықтағаннан кейін оны бірнеше рет пайдалана аласыңдар. Мысалы,

```
def print_lyrics():
    print "I ' m a lumberjack, and I ' m okay."
    print 'I sleep all night and I work all day.'
```

Def — бұл негізгі сөз, ол одан әрі функцияны анықтау керектігін көрсетеді. Функция атауы — print_lyrics. Функция атауларының ережелері айнымалыларға ұқсас: әріптер, сандар және кейбір пунктуация белгілері болуы мүмкін, бірақ басында сан болмауы керек. Негізгі (сақталған) сөздерді функциялардың атаулары ретінде пайдалануға болмайды. Функциялар мен айнымалылардың атаулары сәйкес келмеуі тиіс.

Бос жақшалар атауынан кейін функция аргументтерді қабылдамайтынын көрсетеді. Кейінірек біз кіріс аргументтерін қабылдайтын функцияларды қарастырамыз.

Функцияны анықтаудың бірінші жолы тақырып (header) деп аталады, қалған бөлігі функцияның түрі (body) деп аталады. Тақырып қос нүктемен аяқталады, функция денесі кері шегінеді. Шарт бойынша шегіну әрдайым төрт бос орын болады. Функцияның түрі нұсқаулардың кез келген санын қамтуы мүмкін.

Экранға шығатын жолдар қос тырнақшаға алынған. Бір және екі тырнақшалар өзара алмастырылады, адамдардың көпшілігі бір тырнақшаларды пайдаланады, кейде жалғыз тырнақша жол ішінде кездесетін жағдайлар да болады.

Егер интерактивті режімде функцияны таңдасандар, онда денеге арналған Python функциясы кері шегінеді және соңында бос жолды енгізу қажет болады.

Функцияны анықтау кезінде атауы дәл сондай айнымалы құрылады.

```
> > > print print_lyrics
<function print_lyrics at 0xb7e99e9c>
> > > print type(print_lyrics)
<type 'function'>
```

Print_lyrics мәні 'function' түрі бар функционалдық нысан (function object) болып табылады.

Жаңа функцияны шақыру синтаксисі кірістірілген функцияға ұқсас:

```
> > > print_lyrics()
I 'm a lumberjack, and I' m okay.
I sleep all night and I work all day.
```

Бір рет функцияны анықтағанда оны басқа функциялардың ішінде пайдалануға болады. Мысал үшін жаңа функцияны пайдаланып, жолды қайталайық:

```
def repeat_lyrics():
    print_lyrics()
    print_lyrics()
```

Содан кейін repeat_lyrics шақырыңыз:

```
> > > repeat_lyrics()
I 'm a lumberjack, and I' m okay.
I sleep all night and I work all day.
I 'm a lumberjack, and I' m okay.
I sleep all night and I work all day.
```

Егер жоғарыда сипатталғанның барлығын бірге жинасақ, мынаны аламыз:

```
def print_lyrics():
    print "I' m a lumberjack, and I ' m okay."
    print 'I sleep all night and I work all day.'
def repeat_lyrics():
    print_lyrics()
    print_lyrics()
repeat_lyrics()
```

Бұл бағдарлама екі функцияны қамтиды: print_lyrics және repeat_lyrics. Функциялардың анықтамалары басқа нұсқаулықтарға ұқсас басқару алады, нәтижесі функционалдық нысанды құру болып табылады. Функцияның ішіндегі функция шақырылмайынша басқара алмайды.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Функция дегеніміз не? Ол процедурадан немен ерекшеленеді?
2. Бағдарлама мәтіні бойынша функция қандай мәнді қайтаратынын қалай анықтауға болады? Мысал келтіріңдер.
3. Бағдарлама мәтініне процедураны «жұмыс істеу» үшін енгізу жеткілікті ме?



Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

§20 Жолдармен жұмыс жасау

Бүгінгі сабақта:

- ▶ жолдарды өңдеу үшін процедуралар мен функцияларды қолдануды үйренесіндер.

Тірек ұғымдар:

- ▶ символдық жол;
- ▶ жол ұзындығы;
- ▶ конкатенация;
- ▶ кесінді алу;
- ▶ жолдың реверсі.



Символдық жол дегеніміз не?

Символдық жол дегеніміз таңбалардың тізбегі екенін білесіңдер. Компьютерде әртүрлі 256 символ қолданылады, арнайы кесте бойынша әрбір символдың (0-ден 255-ке дейін) коды бар. Басқа айнымалылар сияқты жол жадыға жазылады және қандай деректер жазылғаны компьютерде тұрады, яғни аппаратта ол байттар тізбегі болады.

Жол қай жерде аяқталатынын қалай анықтауға болады? Оның екі шешімі бар:

1. Жолдың ұзындығын жеке ұяшықта сақтау.
2. Жолдың соңын білдіретін бір ерекше символ таңдау және жолдың ортасында бұл символ кездеспеуі керек.



Символдық жол — жадыда қатар орналасқан (көрші ұяшықтарда) символдардың тізбегі. Символдық шамалармен жұмыс істеу үшін негізгі деректер типі — ол символдық жолдар (типі string).

Жолға мән жазу үшін меншіктеу операторы қолданылады:

```
S = «Қайрат футбол ойнайды».
```



Жол тырнақшаға немесе жалғыз апострофқа алынады. Егер жол тырнақшалармен шектелген болса, оның ішінде апостроф болуы немесе болмауы мүмкін.

Пернетақтадан жолды енгізу үшін `input` функциясын қолданады:

```
s = input («Атын енгізіндер: »)
```

Жолдың ұзындығын анықтау үшін `len` функциясын қолданыңдар.

```
n = len(s)
```

Бұл мысалда `n` айнымалысына `s` жолының ұзындығы жазылады.

```
prin (s[5])
```

Келесі мысалда жолдың жеке белгісі бар. Экранға 5 индексі бар `s` жолының символы шығарылады.

Пернетақтадан жолды енгізетін бағдарламаны қарастырайық, онда барлық «*a*» әріптерін «*б*» әрпіне ауыстырады және алынған жолды экранға шығарады (20.1-сурет).

```
s = input( "Жолды енгізіндер:" )
s1 = ""
for c in s:
    if c == "a": c = "б"
    s1 = s1 + c
print ( s1 )
```

20.1-сурет. Бағдарламаның бөлігі

Түсініктеме берейік. `for c in s` циклінде `s` жолына кіретін барлық символдар іріктеледі. Олардың әрқайсысы кезекті қадамда `c` айнымалысына жазылады. Содан кейін осы айнымалының мәнін тексерейік: егер ол «*a*» әрпімен сәйкес келсе, онда оны «*б*» әрпімен ауыстырамыз және қосу операторының көмегімен `s1` жаңа жолының соңына тіркейміз.



Бағдарламада қандай кемшілікті байқадыңдар?

Бұл мысалдағы жолдарды «көп рет» қосу өте баяу жұмыс істейтін болады. Сондықтан символдарды ауыстыру қажет болатын есептерде стандартты `replace` функциясын қолданған дұрыс.

Жолдармен операциялар орындау. Жолдарды өңдеу кітапханасы жол деректерімен жұмыс істеу, жолдарды салыстыру, символдар жолында іздеу, жолдарды белгілеу (жолдарды логикалық кесектерге бөлу) және жолдар ұзындығын анықтау үшін көптеген пайдалы функцияларды қамтамасыз етеді.

Жолдармен жұмыс істеудің кейбір типтік функцияларын қарастырайық. Жолдармен орындайтын операциялар 20.1-кестеге жинақталған.

20.1-кесте

Жолдармен орындалатын операциялар

Жолдармен операциялар	Мысал	Нәтиже
'+' — жолдарды біріктіру (қосу) үшін қолданылады. Конкатенация (басқа атауы)	<pre>s1 = "Hello" s2 = "Aidin" s = s1 + ", " + s2 + "!"</pre>	s жолда «Салем, Айдын!» жазылады
Кесінді алу операциясы жолдың бөлігін (қосымша жол) ерекшелеу үшін қолданылады	<pre>s = "0123456789" s1 = s[3:8] 3-тен 7-ге дейінгі s жолының символдарын білдіреді (яғни, оны қоспағанда 8-ге дейін)</pre>	s жолға «34567» жазылады
Жолдың бір бөлігін жою үшін жойылатын бөлікке дейін және одан кейін бастапқы жолдың бір бөлігін біріктіре отырып, жаңа жол жасау қажет	<pre>s = "0123456789" s1 = s[:3] + s[9:] S[:3] тілігі "жолдың басынан бастап s[3] символына дейін, оны қоспағанда", ал S[9:] жазбасы - "s[9] бастап жолдың соңына дейін барлық таңбалар" дегенді білдіреді.</pre>	s1 айнымалысында «0129» қалады
Жол реверсі (керісінше жазу)	<pre>s1 = s[::-1]</pre>	Жол элементтерінің бастапқы және соңғы индекстері көрсетілмегендіктен барлық жол іске қосылды. «-1» саны индекстің өзгеру қадамын білдіреді және барлық символдар кері ретпен жазылатынын білдіреді
upper және lower әдістері жолды тиісінше жоғары және төменгі регистрге аударуға мүмкіндік береді	<pre>s = "aAbBcC" s1=s.upper()# "AABBCC" s2=s.lower()# "aabbcc"</pre>	
find жолдарда іздеу функциясы	<pre>s="Алматы менің қалам." n=s.find("c")#n=2 if n >= 0: print("символ нөмірі", n) else: print("Символ табылмады.")</pre>	Find әдісі s жолында үлгі басталатын символдың («c» әрпі) бүтін санын қайтарады. Егер жолда бірнеше үлгі болса, функция олардың біріншісін табады. Егер үлгі табылмаса, функция «-1» қайтарады. Қаралған мысалда N айнымалысына 2 саны жазылады

Осы оқулықта көрсетілмеген басқа да кіріктірме функциялар туралы Интернет ресурстарынан қарап, таныса аласыңдар.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Символдық жол дегеніміз не?
2. Символдық жолға мән қалай беріледі?
3. Жолдың ұзындығы қалай есептеледі?
4. Берілген нөмірмен жол элементіне қалай қол жеткізуге болады?
5. Жолдармен орындалатын негізгі операцияларды атаңдар және мысал келтіріңдер.

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Пернетақтадан арасына бос орын тастап, бір жолға өздеріңнің аты-жөндеріңді жазыңдар.

В ДЕҢГЕЙІ

Енгізілген символдық жолда бас және кіші «a» әрпінің барлығын «b» әрпіне және керісінше ауыстыратын бағдарламаны жазыңдар. «abcABC» жолын енгізгенде «bacBAC» нәтижесі шығуы тиіс.

С ДЕҢГЕЙІ

Натурал сандардың қосындысы жазылған жолды енгізіңдер, мысалы, «1 + 25 + 3». Бұл өрнекті есептеңдер.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§21

Жолдармен жұмыс жасаудың мысалдары

Бүгінгі сабақта:

- ▶ жолдарды өңдеу үшін процедуралар мен функцияларды қолдануды үйренесіңдер.

Тірек ұғымдар:

- ▶ символдық жол;
- ▶ жол ұзындығы;
- ▶ конкатенация;
- ▶ кесінді алу;
- ▶ жолдың реверсі.

Жол — Python нысанының мысалы. Нысанда деректер (шын мәнінде жолдың өзі) және нысанға кіріктірілген және нысанның кез келген данасы (instance) үшін қолжетімді функциялар болып табылатын әдістер бар.

Python-да нысан үшін қолжетімді әдістердің тізімін шығаратын `dir` функциясы бар. `Type` функциясы нысанның түрін көрсетеді:

```
> > > stuff = 'Hello world'
> > > type(stuff)
< type 'str' >
> > > dir(stuff)
['capitalize', 'center', 'count', 'decode', 'encode',
'endswith', 'expandtabs', 'find', 'format', 'index',
'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace',
'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip',
'partition', 'replace', 'rfind', 'rindex', 'rjust',
'rstrip', 'rsplit', 'rstrip', 'split', 'splitlines',
'startswith', 'strip', 'swapcase', 'title', 'translate',
'upper', 'zfill']
> > > help (str.capitalize)
Help on method_descriptor:
capitalize(...)
S.capitalize () -> string
Return a copy of the string S with only its first character
capitalized.
>>>
```

Әдіс функцияға ұқсас болады — ол дәлелдерді қабылдап, өртүрлі синтаксисте мәнді енгізуді қайтарады, бірақ синтаксис өртүрлі болады. Айнымалы атау әдісінің атын жазып, оны дәлелді белгіш ретінде енгізу арқылы қолданады.

Мысалы, `upper` әдісі кіріс жолды алады және жоғары регистрдегі барлық әріптермен жаңа жолды қайтарады:

```
> > word = 'banana'
> > > new_word = word.upper()
```

```
> > > print new_word
```

```
BANANA
```

Бос дөңгелек жақшалар әдісте аргументтер жоқ дегенді білдіреді. Бұл жағдайда `word` нысанының `upper` әдісін шақырдық деп айта аламыз.

Find жол әдісін қарастырайық:

```
> > word = 'banana'
```

```
> > > index = word.find('a')
```

```
> > > print index
```

```
1
```

```
>>>
```

Бұл мысалда `word` нысанының `find` әдісін шақырдық және кіріс параметрі ретінде ізделінді әріпті жібердік. `find` әдісі арқылы тек жеке таңбалар ғана емес, қосымша таңбаларды іздеуге болады:

```
> > > word.find('na')
```

```
2
```

Екінші аргумент ретінде `find` әдісі іздеу басталатын индексті қабылдайды:

```
> > > word.find('na', 3)
```

```
4
```

Кең таралған есептердің бірі — `strip` әдісі арқылы жолдың басынан және соңына дейін бос орындарды (бос орындарды, табуляцияны немесе жолды аудару таңбаларын) жою:

```
> > line = ' Here we go '
```

```
> > line.strip()
```

```
'Here we go'
```

`startswith` сияқты кейбір әдістер логикалық мәндерді қайтарады:

```
> > line = 'Please have a nice day'
```

```
> > line.startswith('Please')
```

```
True
```

```
> > line.startswith('p')
```

```
False
```

Жолды төменгі регистрге ауыстыратын `lower` әдісін алдын ала қолданамыз:

```
> > line = 'Please have a nice day'
```

```
> > line.startswith('p')
```

```
False
```

```
> > line.lower()
```

```
'please have a nice day'
```

```
> > line.lower().startswith('p')
```

```
True
```

Соңғы мысалда біз екі әдісті қатарынан бір өрнекте шақырамыз.

Жолдарды талдау (parsing). Көбіне жолда ішкі жолды табу қажет болады. Мысалы бізде келесідей пішімделген бірнеше жол бар болсын:

```
From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008.
```

Және әрбір жолдан мекенжайдың тек екінші бөлігін шығарғымыз келеді (мысалы, `uct.ac.za`) делік. Мұны `find` әдісін және жол кесіндісін пайдалана отырып жасауға болады.

Біріншіден жолда `at` — таңба (`@`) орнын табамыз. Содан кейін `at` — символдан кейін бірінші бос орын позициясын табамыз. Осыдан кейін ғана жол кесегін пайдалану арқылы іздеген жолдың бөлігін алып тастаймыз:

```
> > data = ' From stephen.marquard@uct.ac.za Sat Jan 5
09:14:16 2008 '
> > > atpos = data.find ('@')
> > > print atpos
21
> > > sppos = data.find (',', atpos)
> > > print sppos
31
> > host = data[atpos+1:sppos]
> > > print host
uct.ac.za
```

Біз іздеуді жүзеге асырғымыз келетін жолда позицияны қоюға мүмкіндік беретін `find` әдісінің нұсқасын пайдаланамыз.

Пішімдеу операторы. `%` пішімдеу операторының таңбасы айнымалыларда сақталған деректер жолының бөлігін ауыстыра отырып, жолдарды жасайды. Бүтін сандарға қолданғанда модуль бойынша алынады.

Екінші оператордың пішімдеуін анықтайтын (`format sequences`) бір немесе одан да көп тізбектелген пішімдеулерден тұратын пішімдеу жолы (`format string`) бірінші операторды береді.

Нәтиже жол болып табылады.

Мысалы, «`%d`» пішімдеу тізбегі екінші оператор бүтін сан ретінде пішімдеуі тиіс дегенді білдіреді (`d` ол `decimal` дегеннен):

```
> > > camels = 42
```

```
> > '%d' % camels
'42'
```

Нәтижесі «42» жолы болып табылады, оны 42 бүтін санымен шатастырмау керек.

Пішімдеу реті жолдың кез келген бөлігінде пайда болуы мүмкін, сондықтан сөйлемде бірізділікті келесі түрде енгізуге болады:

```
> > > camels = 42
> > > 'I have spotted %d camels.' % camels
'I have spotted 42 camels.'
```

Егер жолда пішімделетін тізбек 1-ден көп болса, екінші аргумент кортеж (tuple) болуы керек. Өрбір пішімделетін тізбек реті бойынша кортеж элементімен сәйкес біріктіріледі. Келесі мысалда «%D» — бүтін сандар үшін «%g» — қалқыма нүктелі сандар үшін және «%s» — жолды пішімдеу үшін қолданылады:

```
> > > 'In %d years I have spotted %g %s.' % (3, 0.1, 'camels')
'In 3 years I have spotted 0.1 camels.'
```

Кортеждегі элементтер саны жолдағы форматталатын тізбектер санына сәйкес келуі тиіс. Сонымен қатар элементтердің түрлері форматталатын тізбектерге сәйкес болуы тиіс:

```
> > '%d % d % d' % (1, 2)
TypeError: not enough arguments for for format string
> > '%d' % 'dollars'
TypeError: illegal argument type for built-in operatio
```

БАҚЫЛАУ СҰРАҚТАРЫ



1. Жолдың ұзындығы қалай есептеледі?
2. Жолдармен негізгі операцияларды және оларға сәйкес стандартты функцияларды атаңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 22

Файлдармен жұмыс жасау

Бүгінгі сабақта:

- ▶ ақпаратты оқу және жазу үшін файлдарды пайдалануды үйренесіңдер.

Тірек ұғымдар:

- ▶ мәтіндік файлдар;
- ▶ екілік файлдар.

Өртүрлі қолданбалы бағдарламалармен жұмыс істеу кезінде сендер пернетақтадан көптеген деректерді енгізу өте қиын екенін көрдіңдер, өсіресе бірнеше рет қайталау қажет болса. Мұндай жағдайларда қажетті деректерді жазатын дискіде файлдар жасалады, ал бағдарлама файлдан мәліметтерді өзі оқиды.

Файл — әмбебап құрал. Мәтіндік файлдар (тек әріптерді, сандарды, жақшаларды және т.б. жазуға болады) және *екілік* (онда кестедегі кез келген символдар сақталуы мүмкін) болады. Біз тек мәтіндік файлдарды қарастырамыз.

Python-да әрбір файл байттардың тізбекті ағыны ретінде қарастырылады. Бағдарлама арқылы жіберілген ағын *кіріс* деп аталады, ал бағдарлама арқылы берілетін ағын *шығыс* деп аталады.

Бағдарламалау тілдерінде файлмен жұмыс істеу үшін файл *көрсеткіш* деп аталатын арнайы айнымалы қолданылады. Бұл ашық файл туралы барлық ақпарат сақталған жадыдағы деректер блогының мекенжайы. Python-да файлды ашатын және файлдық көрсеткіш — айнымалыны қайтаратын *open* функциясы қолданылады.

Open функциясы екі параметрді қабылдайды: *файл атауы* (немесе файл жолы) және *файлды ашу режимі*:

- «r» — оқу;
- «w» — жаңа файлға жазу;
- «a» — файлдың соңына қосу.

Кейде бағдарлама файлды аша алмайды. Егер файл оқу үшін ашылмаса, бұл келесі жағдайларда болуы мүмкін:

- дискіде файл немесе файл атауы дұрыс жазылмаған;
- файл басқа бағдарламамен пайдаланылуда және бұғатталған (заблокирован).

Егер файл жазу үшін ашылса, операция сәтсіз аяқталуы мүмкін, себебі:

- дискіде орын жоқ;
- файл жазудан қорғалған;
- файл атауы дұрыс берілмеген (мысалы, онда екі нүкте, сұрақ белгісі және т.б. бар.) болуы мүмкін.

Close әдісі файлды жабады (22.1-сурет):

```
Fin = open ( "input.txt" )
Fout = open ( "output.txt", "w" )
# мұнда файлдармен жұмыс орындалады
Fout.close()
Fin.close()
```

22.1-сурет

Егер файл ашылмаса, табылмаса, қате пайда болады. Егер файл жазу үшін ашылса, оның құрамы жойылып кетуі мүмкін.

Файлды оқу/жазу үшін негізгі функцияларды қарастырайық (22.1-кесте).

22.1-кесте

Оқу немесе жазу үшін негізгі функциялар

Әдістің сипаттамасы	Мысалы
readline — мәтін файлынан бір жолды оқу	s = Fin.readline()
split — жолды бос орындар бойынша бөліп, тиісті сөздерден тізімді құрастырады	s = Fin.readline().split()
int — символдық жолдар бар есептеулерді орындау қажет болғанда қолданылады	a, b = int(s[0]), int(s[1])
write — жолды файлға шығару үшін	Fout.write ("{:d} + {:d} = {:d}\n".format(a, b, a+b))

БАҚЫЛАУ СҰРАҚТАРЫ



1. Ішкі мазмұны бойынша мәтіндік және екілік файлдардың айырмашылығы неде?
2. Файлмен жұмыс істеу үшін қандай айнымалы қолданылады?
3. Файлдан оқу немесе жазу үшін негізгі функцияларды атаңдар.

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Файл бағанында жазылған барлық сандардың орташа арифметикалық мәнін табатын және нәтижені басқа файлға шығаратын бағдарламаны жазыңдар.

B ДЕҢГЕЙІ

Файлда жазылған жұп оң сандар арасында ең кіші және ең үлкен санды табатын және нәтижені басқа файлға шығаратын бағдарламаны жазыңдар. Мұндай сандардың мүлдем болмауы мүмкін екенін ескеріңдер.

C ДЕҢГЕЙІ

Файлда бағанға бүтін сандар жазылған. Бірдей сандардың қатарынан жүретін ең ұзын тізбектің ұзындығын анықтайтын және нәтижені басқа файлға шығаратын бағдарламаны жазыңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§23**Файлдармен жұмыс жасаудың мысалдары**

Бүгінгі сабақта:

- ▶ ақпаратты оқу және жазу үшін файлдарды пайдалануды үйренесіңдер.

Тірек ұғымдар:

- ▶ мәтіндік файлдар;
- ▶ екілік файлдар.

Файлдарды оқу. Файлдың дескрипторында деректер жоқ болғанша, циклде оқып және файлдағы әрбір жол үшін санауышты көбейте отырып, `for` операторымен конструкцияны құрайық.

```
> > fhand = open ('mbox.txt')
> > > count = 0
> > > for line in fhand:
count = count + 1
> > print ('Line Count:', count)
Line Count: 132045
```

Біз файл дескрипторын `for` циклінде бірізділік ретінде пайдалана аламыз. Біздің цикл файлдағы жолдар санын есептейді және соңғы мәнді экранға шығарады. `For` циклінің аудармасы: «файлдық дескриптор ұсынған файлдағы әрбір жол үшін `count` айнымалысын бірлікке көбейту».

`Open` функциясы бүкіл файлды оқымайды, себебі файл бірнеше гигабайт деректерді қамтуы мүмкін. `Open` функциясы файл өлшеміне қарамастан бірдей уақыт алады.

Файл `for` циклі арқылы оқылған жағдайда, `Python` жаңа жолдың таңбасын пайдалана отырып, файлдағы деректерді жеке жолдарға бөлуді жүзеге асырады. `Python` жаңа жолды пайдалана отырып, әрбір жолды оқиды және `for` циклінің әрбір итерациясы үшін айнымалы `line` соңғы символы ретінде жаңа жолды қамтиды.

Сондықтан `for` циклі бір жол бойынша деректерді бір рет оқиды, бұл деректерді сақтау үшін бүкіл жадыны жүктеусіз үлкен файлдағы жолдарды тиімді оқу және есептеу үшін тиімді. Жоғарыда жазылған бағдарлама әр жолды оқу, санау және оларды тастау үшін аз жадын пайдалана отырып, файлдағы кез келген көлемдегі жолдарды санай алады.

Егер файлдың өлшемі жедел жады өлшемімен салыстырғанда салыстырмалы түрде кіші болса, файлдың дескрипторы үшін `read` әдісін пайдалана отырып, бір жолда барлық файлды оқуға болады.

```
> > fhand = open ('mbox-short.txt')
> > > inp = fhand.read()
> > > print(len (inp))
94626
> > print (inp [: 20])
From stephen.marquar
```

Бұл мысалда `mbox-short.txt` файлының барлық мазмұны (94626 таңбалар) `inp` айнымалысында тікелей оқылады. `inp`-да сақталған деректер жолының 20 таңбасын басып шығару үшін жол бөлігін пайдаландық.

Егер деректер файлы компьютердің жедел жадысында жүктелсе ғана `open` функциясын пайдаланудың мұндай тәсілі мүмкін екенін есте сақтаңдар, әйтпесе `for` немесе `while` циклдерін пайдаланыңдар.

Файл арқылы іздеу. Файл арқылы деректерді іздеген кезде бұл көптеген жолдарды елемей және белгілі бір критерийге сәйкес келетін жолдарды өңдей отырып, файл арқылы оқудың жалпы үлгісі болады. Қарапайым іздеу механизмін құру арқылы жолдық әдістермен файлды оқу үшін үлгіні біріктіре аламыз.

Мысалы, егер біз файлды оқып, «`From:`» префиксінен басталатын жолдарды экранға шығарғымыз келсе, біз қалаған префиксі бар жолдарды таңдап, `startswith` жол әдісін пайдалана аламыз.

```
> > fhand = open ('mbox-short.txt')
> > > for line in fhand:
if line.startswith ('From:'):
print (line)
```

Бағдарлама орындалады, келесі нәтижені аламыз:

```
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
...
```

Қалаған нәтиже алынды. Бірақ неге бос жолдар пайда болды? Бұл — жаңа жол символының көрінбейтін нәтижесі. Өр жол жаңа жолмен аяқталады, өйткені `print` функциясы жаңа жолды қамтитын сызық айнымалысынан жолды басып шығарады және одан кейін `print` функциясы басқа жаңа жолды қосады, нәтижесінде біз қос бос орын пайда болғанын байқаймыз.

Біз соңғы таңбадан басқа барлық жолды басып шығару үшін жолдың бір бөлігін пайдалана алар едік, бірақ жолдың оң жағындағы бос орындарды жойып жіберетін `rstrip` әдісін қолданған қолайлы болып табылады:

```
> > > for line in fhand:
        line = line.rstrip()
        if line.startswith ('From:'):
            print (line)
```

Келесі нәтиже аламыз:

```
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
From: zqian@umich.edu
...
```

`Continue` нұсқаулығын пайдаланып, іздеу құрылымын күрделендіре аласындар. Циклде іздеудің негізгі идеясы — «қажет емес» жолды алып тастап, «қажетті» жолды қалдыру.

Келесі үлгі бойынша цикл жасап, қажетті жолдарды қалдырайық:

```
> > > for line in fhand:
        line = line.rstrip()
        if not line.startswith ('From:'):
            continue
        print (line)
```

Бағдарлама қорытындысы алдыңғы нәтижеге сәйкес келеді.

Мәтіндік редакторда іздеу имитациясы үшін `find` жол әдісін пайдалана аласындар. `Find` басқа жолда жол кірісін іздейтіндіктен немесе жол табылмаған болса, 1 жол позициясын қайтаратындықтан, келесі

циклді жазуға болады "@uct.ac.za" (яғни, хаттар Оңтүстік Африкадағы Кейптаун университетінен келеді):

```
> > fhand = open ('mbox-short.txt')
> > > for line in fhand:
    line = line.rstrip()
    if line.find('@uct.ac.za') == -1 :
        continue
    print (line)
```

Келесі нәтижені аламыз:

```
From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008
X-Authentication-Warning: nakamura.uits.iupui.edu: apache
set sender to
stephen.marquard@uct.ac.za using-f
From: stephen.marquard@uct.ac.za
Author: stephen.marquard@uct.ac.za
From david.horwitz@uct.ac.za Fri Jan 4 07:02:32 2008
X-Authentication-Warning: nakamura.uits.iupui.edu: apache
set sender to
david.horwitz@uct.ac.za using-f
From: david.horwitz@uct.ac.za
Author: david.horwitz@uct.ac.za
r39753 | david.horwitz@uct.ac.za | 2008-01-04 13:05:51
+0200 (Fri, 04 Jan
2008) / 1 line
From david.horwitz@uct.ac.za Fri Jan 4 06:08:27 2008
X-Authentication-Warning: nakamura.uits.iupui.edu: apache
set sender to
david.horwitz@uct.ac.za using-f
From: david.horwitz@uct.ac.za
```

БАҚЫЛАУ СҰРАҚТАРЫ



1. Файлдық айнымалы дегеніміз не? Неге файлмен жұмыс істеу үшін файл аты емес, файл айнымалысы қолданылады?
2. Файлдан деректерді оқуды басынан бастап қалай бастауға болады?
3. Файлдағы деректер аяқталғанын қалай анықтауға болады?

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

§24–25 Сұрыптау әдістері

Бүгінгі сабақта:

- ▶ практикалық есептерді шығару үшін сұрыптау алгоритмдерін іске асыруды үйренесіңдер.

Тірек ұғымдар:

- ▶ «сұрыптау» ұғымы;
- ▶ көпіршік әдісі;
- ▶ сызықтық іздеу;
- ▶ екілік іздеу.



Сұрыптау — белгілі бір тізім элементтерін белгіленген тәртіппен орналастыру.

Сұрыптаудың бірнеше түрлері болады (алфавит бойынша, күндер және т.б. бойынша), олар тек элементтерді салыстыру процедурасымен ерекшеленеді. Біз сұрыптаудың қарапайым нұсқасын қарастырайық — өсу тәртібімен массив элементтерін орналастыру. Бағдарламашылар сұрыптаудың көптеген әдістерін ойлап тапты. Олар екі топқа бөлінеді:

- түсінікті, бірақ тиімді емес;
- тиімді, бірақ түсініксіз (жылдам сұрыптау және т.б.).

Өзірге қарапайым міндеттер үшін жеткілікті бірінші топтағы әдістерді ғана қарастырайық.

Көпіршік әдісі. Бұл әдістің атауы белгілі физикалық құбылысқа сәйкес берілген. Әдістің идеясы — стақандағы су көпіршігі түбінен жоғары қарай көтеріледі.

Массивтер үшін — ең кішкентай («жеңіл») элемент жоғары жылжиды («қалқып шығады»).

Бұл былай жүзеге асады:

- төменнен бастап қатар тұрған екі элементті салыстырайық; егер олар «дұрыс емес» тұрса, олардың орындарын ауыстырамыз;
- массив бойынша 1 рет салыстыру үшін бір элемент (ең кішкентай) өз орнына барады (24.1-сурет).

5	5	5	1
2	2	1	5
1	1	2	2
3	3	3	3

24.1-сурет. 1-өту

Содан кейін келесі элементтер жұбын және т.б. қарастырамыз (24.2, 24.3-суреттер).

1	1	1
5	5	2
2	2	5
3	3	3

24.2-сурет. 2-өту

1	1
2	2
5	3
3	5

24.3-сурет. 3-өту

N элементтерден тұратын массивті сұрыптау үшін $N-1$ өту қажет ($N-1$ элементтерді өз орнына қою жеткілікті).

Көпіршікті сұрыптау әдісі кодта берілген (24.4-сурет).

```
#тізім құрамыз
li = [0, 5, 8, 4, 9, 3]
#тізім ұзындығын есептейміз
n = len(li)
#«өту» тізіміне сәйкес саны сыртқы цикл есептейміз.
for j in range(0,n-1):
    #енгізілген цикл i және i+1 элементімен салыстырылып, қажет
    болса, орындарын ауыстыруға болады
    #j өлшеміне сәйкес салыстыру азая береді
    for i in range(0,n-j-1):
        if li[i] < li[i+1]:
            li[i],li[i + 1] = li[i + 1], li[i]
    print(j+1, "- өту циклі - ",end=" ")
    print(li)
print(li)
```

24.4-сурет

Көпіршік әдісі үлкен массивтерде өте баяу жұмыс істейді. Массив көлемі 10 есе ұлғайған кезде бағдарламаны орындау уақыты 100 есе артатыны дәлелденген (әдіс N^2 ретімен).

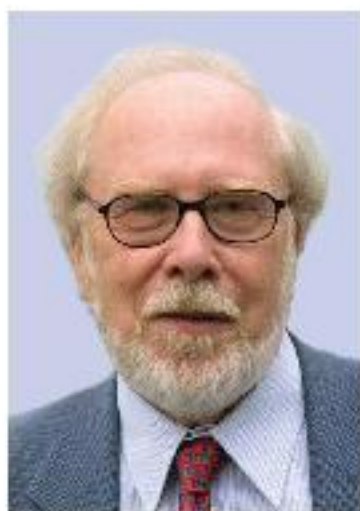
Сызықтық іздеу. Сызықтық іздеу әрбір массив элементін іздеу кiлтiмен салыстырады. Массив реттелмеген болғандықтан, іздеу мәнi массивтiң бiрiншi элементi болуы мүмкiн. Бiрақ бағдарлама массив элементтерiнiң жартысын іздеу кiлтiмен салыстыруы тиiс.

Бұл алгоритмде бiзге қажеттi элемент болатын массив бөлiгiнiң сол және оң жақ шегi қарастырылады. Іздеу бiрiншi бөлiк элементiнен басталады. Егер iзделетiн мән осы нүктедегi функцияның мәнiне тең болмаса, онда келесi нүктеге ауысу жүзеге асырылады. Осылайша әрбiр тексеру нәтижесiнде іздеу аймағы бiр элементке кемидi.

Сызықтық іздеу әдісі шағын немесе сұрыпталмаған массивтер (24.5-сурет) үшін жақсы жұмыс істейді. Алайда үлкен массивтер үшін сызықтық іздеу әдісі тиімсіз.

```
def linearSearch(li, x):
    i = 0
    length = len(li)
    while i < length and x <> li[i]:
        i += 1
    return i if i < length else None
```

24.5-сурет. Сызықтық іздеу әдісінің бір бөлігі



Никлаус Вирт — швейцариялық ғалым, информатика саласындағы маман, бағдарламалау тілдерін өзірлеу саласындағы танымал теоретиктердің бірі

Никлаус Вирт циклдің әрбір орамында бір салыстырудан, атап айтқанда, жолдың аяқталуын тексеруден, құтылуға мүмкіндік беретін осы әдісті әрі қарай зерттеді. Жолдың аяқталуын тексеруден құтылу үшін ізделінетін элементті ең соңына қосамыз, бұл бір шартты тоқтатуға кепілдік береді, ал табылған элементтің позициясына соңғы тексеру оны жалған немесе жалған еместігін көрсетеді (24.6-сурет).

```
def linearSearchVirt(li, x):
    i = 0
    length = len(li) - 1
    while x <> li[i]:
        i += 1
    return i if i < length else None
```

24.6-сурет. Қажетті элементін қосу

Бұдан әрі тиімді әдіс — екілік іздеу әдісін қарастырайық.

Массивте екілік іздеу. Екілік іздеу алгоритмі әр салыстырудан кейін тағы тексерілмеген массив элементтерінің жартысын жоққа шығарады. Алгоритм массивтің орташа элементінің орнын анықтайды және оны іздеу кілтімен салыстырады. Егер олар тең болса, іздеу кілті табылады және осы элементтің индексі беріледі. Өйтпесе есеп массив элементтерінің жартысына қысқарады. Егер іздеу кілті массивтің орташа элементіне қарағанда аз болса, одан әрі іздеу массивтің бірінші жартысында, ал егер көп болса, екінші жартысында жүзеге асырылады. Егер іздеу кілті таңдалған ішкі массивтің (бастапқы массивтің бөлігі) орташа элементімен сәйкес келмесе, алгоритм бастапқы массивтің ширегіне дейін іздеу аймағын қайта қолданады және қысқартады.

Іздеу кілті орта элементке тең болғанға дейін немесе қалған кіші массив құрамында ең болмағанда іздеу кілтіне тең емес бір элемент бар болғанға (яғни, өзiрге іздеу кілт табылмағанға дейін) дейін іздеу жалғасады. Келесі кодты қарастырайық (24.7-сурет):

```
li = [0, 3, 5, 7, 10, 20, 28, 30, 45, 56]
x = 45
i = 0
j = len(li)-1
m = int(j/2)
while li[m] != x and i < j:
    if x > li[m]:
        i = m+1
    else:
        j = m-1
    m = int((i+j)/2)

if i > j:
    print('Элемент табылмады')
else:
    print('Элемент индексі: ', m)
```

24.7-сурет. Соңғы элементін қосу

БҰЛ ҚЫЗЫҚ!

Екілік іздеу – өте тамаша әдіс. Егер, мысалы, массив ұзындығы 1023-ке тең болса, бірінші салыстырудан кейін аймақ 511 элементке дейін, ал екіншісінен кейін 255-ке дейін азаяды. 1023 элементтен тұратын массивте іздеу үшін 10 салыстыру жеткілікті.



БАҚЫЛАУ СҰРАҚТАРЫ

1. Сұрыптау дегеніміз не?
2. Массивтің элементтері қалай сұрыпталады?



ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

1. Массивтегі элементті іздеу алгоритмінің блок-сызбасын құрыңдар.
2. Параграфта берілген бағдарламалық кодты енгізіңдер, бағдарламаны құрастырыңдар және іске қосыңдар.

В ДЕҢГЕЙІ

1. Екі өлшемді массивті қалыптастыру алгоритмінің блок-сызбасын құрастырыңдар, оның элементтері пернетақтадан енгізіледі.
2. Массив элементтерінің қосындысын табуға мүмкіндік беретін бағдарламаны жазыңдар.

С ДЕҢГЕЙІ

1. Натурал сандарды қамтитын бір өлшемді массивті реттеу үшін сұрыптау бағдарламасын қолданыңдар.
2. Деректер массивіндегі ең кіші және ең үлкен мәндерді есептеуге мүмкіндік беретін бағдарламаны жазыңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§26–28 Сұрыптау әдістерінің мысалдары

Бүгінгі сабақта:

- ▶ практикалық есептерді шығару үшін сұрыптау алгоритмдерін іске асыруды үйренесіңдер.

Тірек ұғымдар:

- ▶ «сұрыптау» ұғымы;
- ▶ көпіршік әдісі;
- ▶ сызықтық іздеу;
- ▶ екілік іздеу.

Күрделі құрылымдарды кілтті пайдаланып сұрыптау. Сұрыптау әдісі — табиғатта сандар немесе жолдар сияқты белгілі бір тәртіппен орналасқан заттармен жұмыс істеуге қолайлы, бірақ күрделі құрылымдармен не істеу керек? Мұны `sorted()` функциясы арқылы жүзеге асыруға болады.

`Sorted ()` функциясы кілтті аталған параметр ретінде қабылдайды. Бұл кілт сұрыптау мақсатында мәнді анықтау үшін `sorted ()` функциясымен пайдаланылатын бір параметрді қабылдайтын функция болуы керек. Мысал қарастырайық. Айталық, бізде аты мен жасын жазу атрибуттары бар `Person` класы берілсін:

```
class Person(object):
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __repr__(self):
        return "" % (self.name, self.age)
```

мұндағы `__repr__` функциясы — Python интерпретаторында нысан қалай ұсынылатынын қайта анықтау үшін пайдаланылатын арнайы функция.

Функцияны анықтаудың себебі — *сұрыптау тәртібін таңдау*. Пайдаланушы анықтаған нысандарды келесі түрде көрсетеді: «`<__main__.Person object at 0xb7083ccc>`». Егер осы күйінде қалдырса, онда төмендегі мысалдарда өртүрлі даналарды ажырату үшін біршама қиындықтар туындауы мүмкін.

Адамдар тізімін жасайық:

```
jack = Person('Jack', 19)
adam = Person('Adam', 43)
becky = Person('Becky', 11)
people = [jack, adam, becky]
```

`Sorted ()` функциясы өзі адамдар тізімімен не істеу керекін білмейді:

```
a = sorted(people)
print (a) # [<name: Jack, age: 19>, <name: Adam, age: 43>, <name: Becky, age: 11>]
```

Бірақ та біз `sorted ()` функциясына қолданылатын кілтті көрсету арқылы сұрыптау атрибутын көрсете аламыз. Мұны келесі мысалда анықтайық:

```
def byName_key(person):
    return person.name
```

Кілт функциясы бір аргументті қабылдап, сұрыптау негізделген мәнді беруі керек. `Sorted ()` функциясы қолданылатын итерацияланатын әрбір элементте `key` функциясын тудыруы және тізімді сұрыптау кезінде беру мәнін пайдалануы тиіс.

```
a = sorted(people, key = byName_key)
print (a) # [<name: Adam, age: 43>, <name: Becky, age: 11>, <name: Jack, age: 19>]
```

Назар аударындар, функцияның өзіне сілтеме беріп, оны шақырмай, оның қайтарылатын мәніне сілтеме береміз. Бұл — өте маңызды сөт. Естеріңде болсын, `sorted()` `key` функциясын әрбір итерацияланатын элементке шақыру арқылы қолданады.

Тағы бір кодты қарастырайық, бұл жолы адамның жасын сұрыптауды қарастырамыз:

```
def byage_key(person):
    return person.age
a = sorted(people, key = byAge_key)
print (a) # [<name: Becky, age: 11>, <name: Jack, age: 19>, <name: Adam, age: 43>]
```

Кері сұрыптау. `Sorted()` функциясы кері сұрыптауды жеңілдетеді. Функция қатаң логикада әрекет ететін `reverse` деп аталатын опциялық параметрді қабылдайды.

```
data = [3, 2, 5, 4, 7, 1]
a = sorted(data, reverse = True)
print (a) # [7, 5, 4, 3, 2, 1]
data = ('Zane', 'Bob', 'Janet')
b = sorted(data, reverse = True)
print (b) # ['Zane', 'Janet', 'Bob']
```

Attrgetter функциясын қолданып сұрыптау. Бұл жолы кері сұрыптау адамның жасы бойынша жүргізілген. Шын мәнінде, белгілі бір нысан атрибуты бойынша сұрыптау — сендер үшін кілттер функциясын жасай алатын функцияның арқасында орындай алатын Python функциясының қарапайым міндеті:

```
from operator import attrgetter
```

Attrgetter() шақыру нәтижесі — бұл қарастырған екі функцияға ұқсас функция. Таңдау үшін атрибут атауын анықтайық, содан кейін `attrgetter` нысанды қабылдап, осы нысаннан белгілі бір атрибутты қайтаратын функциясын жасайды.

```
getName = attrgetter ('name')
result = getName (jack)
print(result) # 'jack'
```

Осылайша `attrgetter(name)` біздің `byName_key()` функциясымен ерте анықталған функцияны қайтарады:

```
result = sorted(people, key = attrgetter('name'))
print(result) # [<name: Adam, age: 43>, <name: Becky, age: 11>, <name: Jack, age: 19>]
```

`attrgetter(age)` функциясы `byAge_key()` функциясы ретінде белгілі бір функцияны қайтарады:

```
result = sorted(people, key = attrgetter('age'))
```

```
print(result) # [<name: Becky, age: 11>, <name: Jack, age: 19>, <name: Adam, age: 43>]
```

Сұрыптау функциясында кеу-ді алдын ала қолдану. Осы күнге дейін негізгі функцияларымыз қарапайым атрибуттарды санаушылар болды, бірақ олар сұрыптау үшін мәндерді де есептей алады. Тағы бір мысалды қарастырайық. Бұл жолы Snake класын анықтаймыз:

```
class Snake(object):
    def __init__(self, name, toxicity, aggression):
        self.name = name
        self.toxicity = toxicity
        self.aggression = aggression
    def __repr__(self):
        return "<%s>" % self.name
```

Жыланның аты белгілі, toxicity (уландыруының мөлшері қаншалықты екенін өлшейді) және aggression (0-ден 1-ге дейінгі сан түрінде ұсынылған, ол жыланның шабуыл жасау ықтималдығын көрсетеді) бар.

```
gardenSnake = Snake('gardenSnake', 10, 0.1)
rattleSnake = Snake('rattleSnake', 100, 0.25)
kingCobra = Snake('kingCobra', 50, 1.0)
snakes = [rattleSnake, kingCobra, gardenSnake]
```

Енді уыттылық пен агрессивтілік көрсеткіштерін негізге ала отырып, жылан қаншалықты қауіпті екенін және қауіптілік дәрежесі бойынша жыландар тізімін сұрыптай аласыңдар:

```
def byDangerous_key(snake):
    return snake.toxicity * snake.aggression
result = sorted(snakes, key = byDangerous_key)
print(result) # [<gardenSnake>, <rattleSnake>, <kingCobra>]
```

Жыландар күтілген тәртіппен сұрыпталған (жылан (rattlesnake) кобраға қарағанда kingCobra улы жылан, кобраның агрессивтілік деңгейі оны аса қауіпті етеді).

Кездейсоқ сұрыптау. Кілттер сұрыпталатын элементтермен байланысты болуы міндетті емес (алайда бұл сұрыптаудың ең өнімді тәсілі емес). Келесі кілтпен кез келген ретті жасай аласыңдар:

```
from random import random
def randomOrder_key(element):
    return random()
```

random() функциясы — стандартты кітапхананың бір бөлігі, ол кездейсоқ түрде 0-ден 1-ге дейін сандар береді. Осы кілтті пайдалана отырып, сұрыптау кездейсоқ тәртіпті береді:

```
a = sorted(snakes, key = randomOrder_key)
```

```
print(a) # [<kingCobra>, <gardenSnake>, <rattleSnake>]
b = sorted(snakes, key = randomOrder_key)
print(b) # [<rattleSnake>, <kingCobra>, <gardenSnake>]
```

БАҚЫЛАУ СҰРАҚТАРЫ



1. Көпіршіктің әдісі қандай идеяға негізделген? Таңдау әдісі?
2. Көпіршік әдісі мен таңдау әдісін мысал арқылы салыстырыңдар. Олардың қайсысы аз ауыстыруды қажет етеді?

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§29–30

Графтар және оларды алгоритмдеуде қолдану

Бүгінгі сабақта:

- ▶ практикалық есептерді шешу үшін графтарда іздеу алгоритмдерін жүзеге асыруды үйренесіңдер.

Тірек ұғымдар:

- ▶ граф;
- ▶ екілік граф;
- ▶ иерархия;
- ▶ ені бойынша іздеу;
- ▶ тереңдікке іздеу.

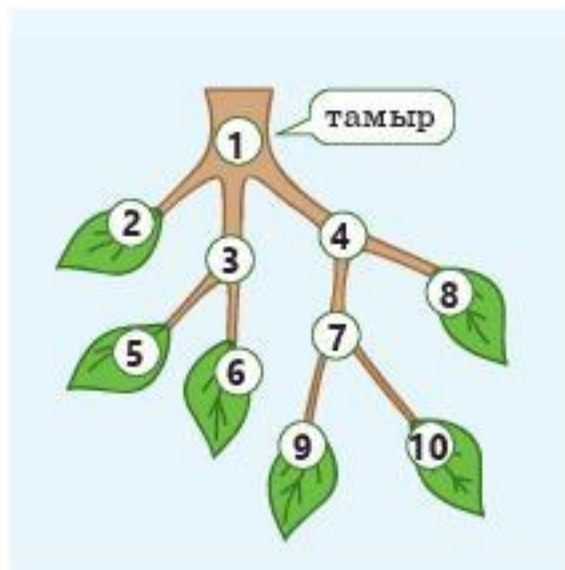
Граф — информатикадағы ең көп таралған құрылымдардың бірі (29.1-сурет). Графты ақпаратты сақтау үшін және жылдам іздеуді қамтамасыз ететін иерархиялық байланысты деректерді ұсыну тәсілі ретінде пайдалануға болады.

Граф түйіндерден және олардың қабырғаларынан тұрады. Python-да графқа сәйкес келетін арнайы кірістірілген деректер құрылымы жоқ. Python-да графты сипаттау үшін салынған тізімдерді пайдаланады.

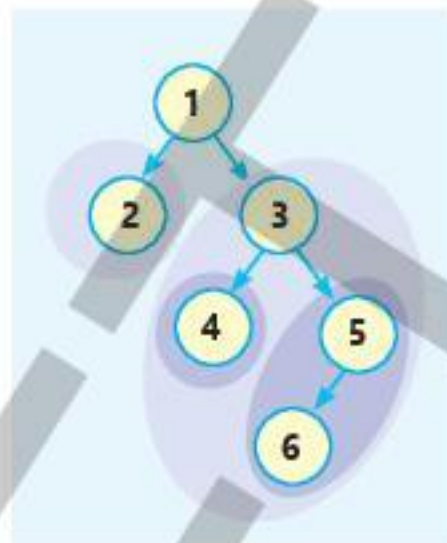
Бұл параграфта екілік графты қарастырайық (29.2-сурет).



Екілік (бинарлы) граф — әрбір түйіні екі түйіннен аспайтын граф.



29.1-сурет. Граф құрылымы



29.2-сурет. Екілік граф мысалы

Екілік граф туралы деректерді тізім түрінде сақтау үшін рекурсивті анықтама қолданылады:

— бос граф — екілік граф;

— екілік граф — бұл түйін және онымен байланысқан екілік ішкі граф.

Осылайша Python екілік граф үш элементтен тұрады. Бірінші элемент — бұл түйін деректері (белгі), екіншісі — «сол жақ түйін», үшінші элемент — «оң жақ түйін», әрбір кіші граф бос болуы мүмкін.

Келесі мысалдарды қарастырайық:

а)	ө)	б)	в)

29.3-сурет. Екілік граф сұлбасы

29.3, а-суретте бір «а» белгісі бар граф ұсынылған және [«а», [], []] түрінде жазуға болады. Егер екеуі де бос болса, олар жазылмайды, жазба [«а»] түрінде болады.

29.3, в-суретте үш түйіндері бар графты [«а», [«б», «с»]] деп жазамыз. 29.3, б және в-суреттерде графтарды сәйкесінше [«а», [«б»]] және [«а», [[], «с»]] түрінде жазуға болады.



Назар аударыңдар, 29.3, *в*-суреттегі мысалында сол түйіннің орнында бос тізімнің болуы міндетті, өйтпесе «с» белгісі бар торап оң емес, сол түйін ретінде қарастырылады (көптеген тапсырмалар үшін бұл маңызды).

Екілік граф түйіндерін айналып өту есебінің мысалын қарастырайық. Қарапайым нұсқа — Бұл ТСО («түбір — сол жақ — оң жақ»).

Екілік граф «Т» тізімі түрінде берілген болсын. Бұл тізімнің бірінші элементі $T[0]$, сол жағы $T[1]$, оң жағы $T[2]$. Өрбір түйін үшін келесі өрекеттерді орындау қажет:

— түйінді өңдеу (бұл үшін тек түйін белгісін экранға шығарамыз);

— алдымен сол, содан кейін оңға бұрылып шығарамыз.

Айналып өту алгоритмі рекурсивті болып табылады (29.4-сурет).

```
def DFS ( T ):
    if not T: return
    print(T[0], end=" ")
    for sub in T[1:]:
        DFS ( sub )
```

29.4-сурет. Айналып өту алгоритмі

Егер функцияға бос граф берілсе, шығыс (рекурсияның аяқталуы) орындалады. Егер граф бос болмаса, экранға торап белгісі шығады, содан кейін циклде рекурсивті шақырулар арқылы барлық кіші графтарды айналып өту орындалады.



DFS функциясы (depth-first search) — тереңдікке іздеу. Демек, мұндай айналып өту кезінде алдымен графтың тереңдігіне оның сол жақтағы соңғы элементіне (жапырағына) дейін бұтақтан өтіп, содан кейін артқа ораламыз.

«Ені бойынша іздеу» (breadth-first search) деп аталатын графты (29.5-сурет) айналып өту үлгісін қарастырайық.

Оның мәні: алдымен түбірі, содан кейін «түйіндер», өрі қарай «түйіндердің түйіндері» өңделеді т.б. Мұндай айналып өтуді кезектің көмегімен ұйымдастыру ыңғайлы:

```
def BFS ( T ):
    if not T: return
    Q = [T]
    while Q:
        node = Q.pop(0)
        print(node[0], end=" ")
        for sub in node[1:]:
            if sub: Q.append ( sub )
```

29.5-сурет. Айналып өту мысалы

Алдымен кезекке графтың түбірі жазылады. Содан кейін кезек бос болмайынша, кезектің басынан бір элементті аламыз, осы торапқа барамыз (оның белгісін экранға шығарамыз) және оның барлық бос емес кіші графтарына сілтемелерді жазамыз. Бұл ретте, бірінші кезекте сол жағы, содан кейін оң жағы алынады.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Екілік граф дегеніміз не?
2. Ені бойынша іздеу алгоритмінің мәнін түсіндіріңдер.
3. Тереңдікке іздеу алгоритмінің мәнін түсіндіріңдер.

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Ойын графын жасаңдар.

Екі ойыншының да алдына үйілген тастар қойылды. Бірінші үймеде — 3, екіншісінде 2 тас бар (29.6-сурет). Өрбір ойыншының бұдан басқа шексіз көп тастары бар.

Ойыншылар кезекпен жүреді. Ойыншы қандай да бір үймедегі тастардың санын 3 есе арттырады немесе бір үймеге 1 тасты қосады.

Ойын жүрісінен кейін тастардың жалпы саны екі үйілмеде кем дегенде 16 болғанда, ойыншы жеңеді.

Ойында кім жеңеді? Бірінші қадам жасайтын ойыншы ма, әлде екінші қадам жасайтын ойыншы ма? Ойыншы қалай жүруі керек?



29.6-сурет

В ДЕҢГЕЙІ

$+$, $-$, $*$, $/$ операциялардың тек бір мәнді сандары мен белгілерін қамтитын дұрыс арифметикалық өрнектерді есептеуге арналған бағдарламаны құрастырыңдар.

С ДЕҢГЕЙІ

$+$, $-$, $*$, $/$ операциялардың тек бір мәнді сандары мен белгілерін және көп мәнді сандар мен жақшаларды қамтитын дұрыс арифметикалық өрнектерді есептеуге арналған бағдарламаны құрастырыңдар.

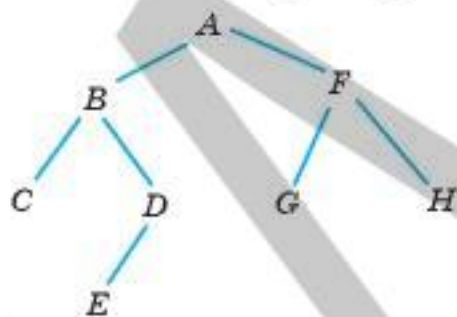
Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

Өзіңді тексер!

1. Мәнді қайтармайтын ішкі бағдарлама қалай аталады?
 - а) процедура;
 - ә) функция;
 - б) құрылым;
 - в) итерация;
 - г) рекурсия.
2. Ішкі бағдарламалар — функциялардың ерекшелігі неде?
 - а) нақты өрекеттерді орындаудан басқа, функция, меншіктеу операторының оң жағында қолданылатын мәнді қайтарады;
 - ә) міндетті түрде оларда параметрлер болу қажет;
 - б) параметрлер болмайды;
 - в) параметрлердің тұрақты саны болады;
 - г) параметрлердің саны айнымалы болады.
3. Бағдарламада келесі s символдық жол қолданылады: $s = \text{«Көне штиблеттен сәлем!»}$ Бұл символды жолда дұрыс операторларды белгілеңдер:
 - а) $s = \text{"12345"};$
 - ә) $\text{"12345"} = s;$
 - б) $s[1] = \text{"0"};$
 - в) $s1 = s[:];$
 - г) $s[:] = s1.$

4. Бағдарлама орындалғаннан кейін экранда не шығады?
- `s = "123";`
 - `s = s + "0" + s;`
 - `s = s + s;`
 - `print (s).`
5. Бағдарлама орындалғаннан кейін экранда не шығады?
- `s = "0123456";`
 - `q = "abc";`
 - `s = s[:2] + s[5:];`
 - `q = q[0] + s + q[1:];`
 - `print (q).`
6. Программа орындалғаннан кейін экранда не шығады?
- `s = "0123456789";`
 - `n = s.find("456");`
 - `q = s[:n+2] + s[n+5:];`
 - `print (q).`
7. Граф дегеніміз не?
- циклсіз граф, оның бір төбесі графтың түбірі болады;
 - қабырғасы жоқ граф, оның бір төбесі графтың түбірі болады;
 - төбесі жоқ граф, тек қырлары болады;
 - тұзағы жоқ граф;
 - барлық төбелерінің арасында қабырғалары бар граф.
8. Екілік (бинарлық) граф дегеніміз не?
- әрбір түйіннің (төбесі) екіден көп емес желісі болатын граф;
 - әрбір түйін (төбесі) екі желісі бар граф;
 - екіден көп емес бұтағы бар граф;
 - екі бұтағы бар граф;
 - екі түбірі бар граф.
9. Келесі екілік (бинарлық) графты тура айналып өтуді орындау:



- A, B, C, D, E, F, G, H;
- C, B, E, D, A, G, F, H;
- C, E, D, B, G, H, F, A;
- H, G, F, E, D, C, D, A;
- A, B, F, C, D, G, H, E.

10. 100 элементтен тұратын массивте тізбектей іздеу барысында неше рет салыстыру қажет?
- а) максимум 100;
 - ә) максимум 20;
 - б) минимум 5;
 - в) минимум 25;
 - г) орташа 20.
11. 1, 2, 3, 4, 5, 6, 7 сандар массивінде 4 санын тізбектеп іздеу кезінде сандарды неше рет салыстыру орындалады?
- а) 4;
 - ә) 5;
 - б) 6;
 - в) 1;
 - г) 2.
12. Екілік іздеуді тағы да қалай атайды?
- а) бинарлы іздеу, жартылай бөлу әдісі, дихотомия;
 - ә) жұп;
 - б) екіжақты;
 - в) жұпты;
 - г) қосарланған.
13. 100 элементтен тұратын массивте екілік іздеу кезінде неше рет салыстыру қажет?
- а) максимум 7;
 - ә) максимум 5;
 - б) максимум 2;
 - в) минимум 5;
 - г) минимум 2.
14. S жолында Q жолының символдарын кері ретпен жазу үшін көпнүктенің орнына қандай оператор жазу қажет?
- а) $Q = \langle 0123456789 \rangle$;
 - ә) $S = \langle 0 \rangle$;
 - б) `for k in range(1,10).`



АҚПАРАТТЫҚ ЖҮЙЕЛЕР

Бұл тараудан білесіңдер:

- ▶ Bigdata-ны;
- ▶ деректер қорының негізгі ұғымдарын;
- ▶ деректер қорымен жұмыс жасауды;
- ▶ деректер қорының бастапқы кілтін;
- ▶ деректер қорындағы формалар, есептер, сұраныстарды;
- ▶ құрылымдалған сұраныстарды.

Мынаны үйренесіңдер:

- ▶ Bigdata-ны пайдалануда оң және теріс әсерлерін бағалауды;
- ▶ «реляциондық деректер қоры» ұғымын түсіндіруді;
- ▶ деректер қорында бастапқы кілтті анықтауды;
- ▶ деректер қоры жолдарының мәліметтер типін анықтауды;
- ▶ біркестелі және көпкестелі деректер қорын құруды;
- ▶ деректерді енгізуге арналған форма жасауды;
- ▶ алынған мәліметтерді қолдана отырып есептер жасауды;
- ▶ конструкторды пайдаланып, іріктеу сұранысын құруды;
- ▶ құрылымдалған сұраныстар тілін пайдалануды.

§31 Bigdata

Бүгінгі сабақта:

- ▶ Bigdata қолданудың он және теріс жақтарын бағалауды үйренесіңдер.

Тірек ұғымдар:

- ▶ Bigdata – үлкен деректер.

Bigdata термині шамамен 2011 жылдан бастап қолданыла бастады.



Bigdata дегеніміз не?

Өртүрлі салаларда (маркетинг, экономика) келесі анықтамалар қолданылады:

- Bigdata — 500 Гб, 1 Тб-дан үлкен деректер.
- Bigdata — Excel бағдарламасында өңдеуге болмайтын деректер.
- Bigdata — бір компьютерде өңдеуге болмайтын деректер.



Bigdata (үлкен деректер) — көптеген тораптар бойынша үлестіру жағдайында адам қабылдайтын нәтижелерді алу үшін үлкен көлемдегі құрылымдалған және құрылымданбаған деректерді өңдеу тәсілдерінің, құралдарының және әдістерінің тобы.

Bigdata деп деректердің қандай да бір көлемін емес, ақпаратты үлестіріп өңдеуге мүмкіндік беретін деректерді өңдеу әдістерін түсінеміз.

Бұл әдістерді үлкен деректер массивіне (мысалы, Интернеттегі барлық парақшалардың мазмұны), сондай-ақ шағын деректер массивіне (мысалы, осы тақырып бойынша жобаға) қолдануға болады.



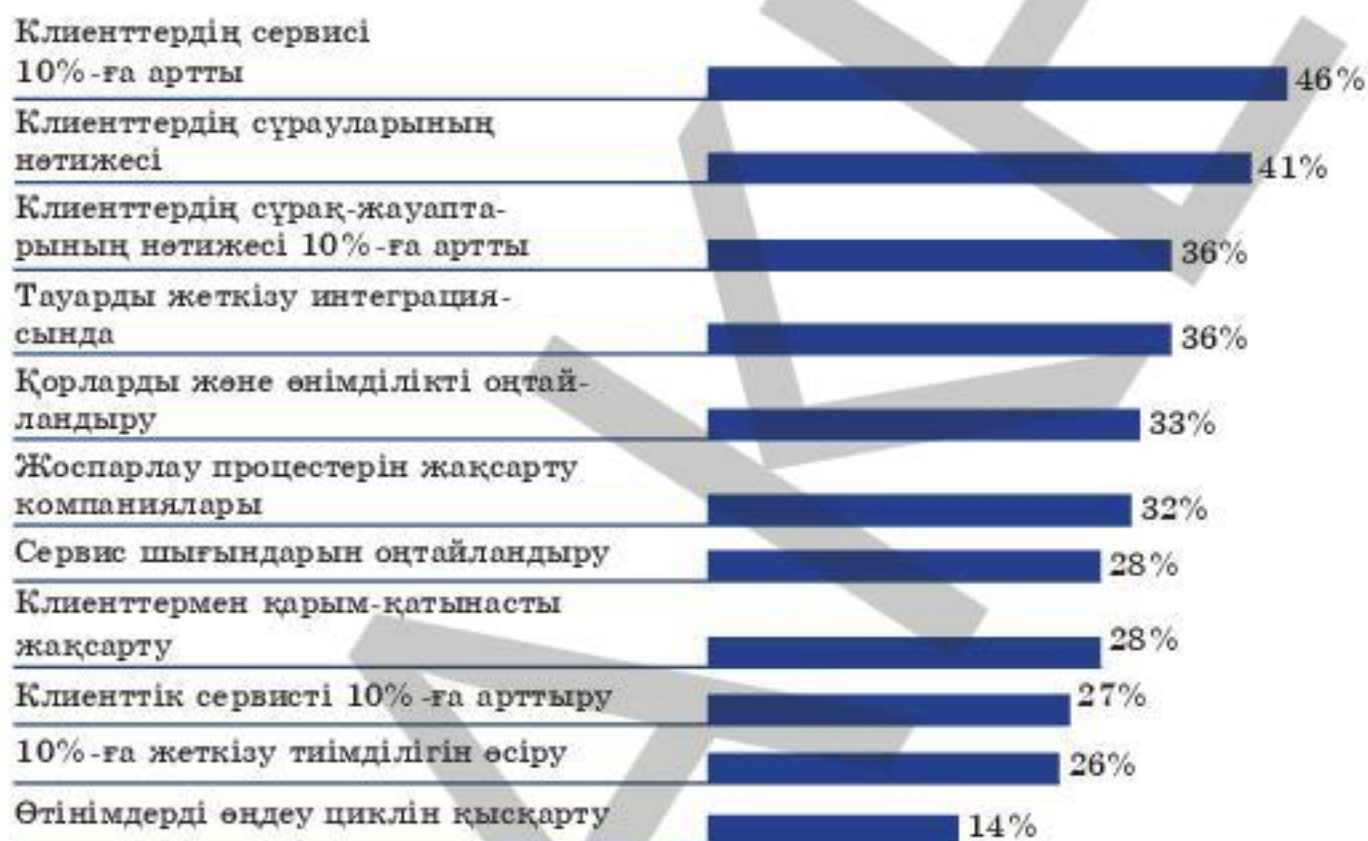
Bigdata не үшін қажет?

— *Біріншіден*, қарапайым реляциялық деректер қоры тиімді қолдануға мүмкіндік бермейтін жүздеген терабайт және петабайт деректер көлемін сақтау және басқару үшін;

— *екіншіден*, мәтіннен, суреттерден, бейнелерден және басқа деректер типінен тұратын құрылымдалған ақпаратты ұйымдастыру үшін қажет.

Қазіргі уақытта дереккөздердің саны қарқынды өсуде, сонымен қатар ақпаратты сақтау мүмкіндіктері де өсуде, демек, оларды өңдеу технологиялары соғұрлым сұранысқа ие болып келеді.

2000 жылдардан бастап сандық тасымалдауыштар кең таралды, осылайша сандық түрдегі ақпарат қолжетімді болды (31.1-сурет). Сандық дәуірдің дамуына қатқыл дискілер ерекше үлес қосқанын айту қажет. Олардың өндірісін арзандату — үлкен деректер трендінің қалыптастырудың негізгі факторы деп айтуға болады.



31.1-сурет. Bigdata-ны енгізудің нәтижелері

Үлкен деректермен жұмыс істеу әдістері қажет болатын деректерге бірнеше мысал келтірейік (31.2-сурет):

- көлік компаниялары үшін автомобильдердің GPS-сигналдары;
- ұлттық кітапханада кітаптарды цифрлеу;
- банктің барлық клиенттерінің транзакциялары туралы ақпарат;
- ірі сауда желісіндегі барлық сатылымдару туралы ақпарат және т.б.



31.2-сурет. Үлкен деректермен жұмыс істеу сұлбасы

Bigdata-мен жұмыс істеу алгоритмін келесі түрде сипаттауға болады: деректерді жинақтау, алынған ақпаратты есептер мен дашборд (dashboard) көмегімен құрылымдау, инсайт және мәнмәтін құру, сонымен қатар әрекет етуге ұсынымдар тұжырымдау.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Bigdata дегеніміз не?
2. Үлкен деректерді қолдану мысалдарын келтіріңдер.
3. Bigdata жұмыс істеу алгоритмін сипаттаңдар.

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

«Дашборд дегеніміз не?» жобасын өзірлеңдер.

В ДЕҢГЕЙІ

«Банкте алаяқтыққа жол бермеу үшін Bigdata қалай қолданылады?» жобасын өзірлеңдер.

С ДЕҢГЕЙІ

«Телекоммуникациялық провайдерлерде Bigdata қалай қолданылады?»

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

§32–33 Bigdata. Жобалау жұмысы*Бүгінгі сабақта:*

- ▶ Bigdata қолданудың оң және теріс жақтарын бағалауды үйренесіндер.

Тірек ұғымдар:

- ▶ Bigdata – үлкен деректер.

Жоба тақырыптары:

1. Медицинада үлкен деректерді қолдану. Медицина саласында Bigdata технологиясын іске асыру дәрігерлерге ауыруды мұқият зерттеуге және нақты жағдай үшін тиімді емдеу курсың таңдауға мүмкіндік береді. Ақпаратты талдаудың арқасында медицина қызметкерлері рецидивтерді болжау және алдын алу шараларын қолдану оңай болады. Нәтижесі — диагнозды дәл қою және емдеудің жетілдірілген әдістері.

Жаңа әдістеме науқастардың проблемаларын басқа жағынан қарауға мүмкіндік береді, бұл бұрын белгісіз мәселе көздерінің ашылуына әкеледі.

**32.1-сурет. Медицина саласындағы технологиялар**

Мысалы, кейбір нәсілдер басқа этностық топтардың өкілдеріне қарағанда, жүрек ауруларына генетикалық жағынан бейім. Енді науқас белгілі бір ауруға шағымданған кезде, дәрігерлер оның нәсілінің өкілдері туралы деректерді назарға алады. Деректерді жинау және талдау науқастар туралы көбірек білуге мүмкіндік береді: тамақты ұнатудан және өмір сүру стилінен бастап ДНҚ генетикалық құрылымына және жасушалардың, ұлпалардың, ағзалардың метаболиттеріне дейін.

Тағы бір мысал — Нью-Джерсидегі Лурд медициналық орталығында болған оқиға. Емделуші дәрі-дәрмектің мезгілсіз қабылдауынан туындаған құрысу ұстама салдарынан тексерілуден өткенде, дәрігерлер ер адамның денсаулығында бұдан басқа да күрделі ауру бар екенін анықтады. Бұл жүрекше фибрилляциясы еді. Диагнозды анықтау бөлімше қызметкерлері пациенттің телефонына, атап айтқанда, оның фитнес-жаттықтырушымен байланысқан қосымшаға қол жеткізуінің арқасында мүмкін болды. Қосымшадан алынған мәліметтер диагнозды анықтаудағы негізгі факторы болды, себебі тексеру кезінде ер адамның ешқандай жүрек ауытқулары анықталған жоқ.

Бұл — медицина саласында үлкен деректерді пайдалану бүгінгі күні осындай маңызды рөл атқаратынын көрсететін екі мысал ғана.

2. Bigdata — бөлшек сауда өзегі. Мысалы, Target дүкен желісі деректерді терең талдау және жеке болжау жүйесі арқылы тұтынушылардың талғамдарын жоғары дәлдікпен анықтай алады. Өрбір клиент үшін *id* бекітіледі, ол өз кезегінде кредиттік картаға, есімге немесе электрондық поштаға бекітіледі. Идентификатор бір кездері адам сатып алған заттары туралы барлық ақпарат сақталған бума болып табылады.

3. Заң және тәртіп қорғаудағы үлкен деректер. Соңғы бірнеше жылда құқық қорғау ұйымдары үлкен деректерді қалай және қашан пайдаланатынын анықтады. Ұлттық қауіпсіздік агенттігінің лаңкестік актілерді болдырмау үшін үлкен деректер технологияларын қолданатыны жалпыға белгілі факт болып табылады. Басқа ведомстволар ұсақ қылмыстарға жол бермеу үшін прогрессивті әдіснаманы іске қосады.

4. Bigdata технологиясының қалалардың дамуына көмегі. Деректерді талдау қалалар мен елдердің тіршілік әрекетінің бірқатар аспектілерін жақсарту үшін де қолданылады. Мысалы, Bigdata технологиясын қалай және қашан пайдаланғанын біле отырып, көлік ағындарын оңтайландыруға болады. Ол үшін автокөліктердің онлайн режімде қозғалуы есепке алынады, әлеуметтік медиа және метеорологиялық деректер талданады. Мысалы, автобустар кешігіп келе жатқан по-

йызды күтіп тұрады, ал бағдаршамдар жолдардағы кептелісті азайту үшін жолдардағы жүктілікті болжауға қабілеті бар «ақылды» қаланың концепциясы құрылған (32.1-кесте).

32.1-кесте

Жобаны бағалау критерийлері

Критерийлер	0 балл	1 балл	2 балл	3 балл
1	2	3	4	5
Зерттеу проблемасын қою деңгейі	Репродуктивті сипаттағы жұмыс — басқа көздерден алынған ақпарат қана бар, жалпылама жоқ, мазмұндық қорытындылар жоқ	Жалпы жұмыс репродуктивті, бірақ жақсы, дербес қорытынды жасалды	Жұмыс ішінара ізденісті талап етеді. Жұмыста жеке сипаты бар проблема-лар бар (жалпы тақырыпқа емес, тек оның қандай да бір аспектілеріне қатысты)	Зерттеу жұмысы бір ғылыми проблеманы шешуге арналған, жаһандық емес, бірақ өз бетінше тұжырымдалған
Тақырыптың өзектілігі мен бірегейлігі	Тақырыбы белгілі, толық зерттелген, әдебиетте толық қамтылған. Бұл ретте автор жеке мәселелерді немесе қызығушылықты шешуге байланысты субъективті мүддеден басқа оның таңдауының немен байланысты екенін көрсете алмады	Тақырып зерттелген, бірақ онда жаңа деректердің салдарынан «ақ дақтар» пайда болды немесе тақырып өте аз белгілі, бірақ мәселе «жасанды». Ғылым үшін шынайы қызығушылық танытпайды	Тақырыбында «ақ дақтар» жеткілікті немесе мәселе дұрыс қойылған, соның салдарынан тақырып күтпеген жағынан ашылады	Тақырып аз зерттелген, сипаттамасы жоқ, оны ашу үшін зерттеудің көршілес салаларының көзқарастарын салыстыра отырып, өз бетінше көптеген қорытындылар жасауды талап етеді
Дәлелдемелердің қисындылығы)	Жұмыс осы тақырып бойынша авторға белгілі бір жүйесіз баяндау болып табылады	Жұмыста ақпаратты құрудағы кейбір қисындылықты байқауға болады, бірақ тұтастық жоқ	Жұмыста кейбір маңызды дәлелдер қалып қойған немесе «артық» ақпарат бар. Қажетсіз толық мәтін жүктелген, бірақ жалпы логика бар	Мақсат дәйекті түрде іске асырылды, қажетті үлгілер жасалды, мәтінді қажетсіз толық жүктейтін қосымша ақпарат жоқ

1	2	3	4	5
Әдеби ақпарат көздерді пайдаланудағы өдептілік	Жұмыста бір-біріне қарама-қайшы келетін және орнықсыз пайдаланатын көзқарастар авторларына сілтеме жоқ	Барлық қажетті жағдайларда қарама-қайшылықтар жоқ, сілтемелер жоқ немесе олар сирек болады	Мәтін принциптік мазмұн туралы ақпарат (анықтамалар, жалпылама, сипаттама, пікір, бағалау және т.б.) жасалған жағдайларда авторларға неғұрлым қажетті сілтемелерді қамтиды)	Мәтін мазмұндық ақпарат берілген жағдайларда (анықтамалар, сипаттама, жалпылама, сипаттама, пікір т.б. бағасы) авторларға барлық қажетті сілтемелерден тұрады, бұл ретте автор өз көзқарасын дәлелдей отырып, беделді дерек-көзге жүгінгенде бөтен пікірді шебер пайдаланады
Дереккөздер саны	Әдебиеттер тізімі жоқ	1-2 дереккөзі	Тізім бірнеше дереккөздерді қамтиды, бірақ қарастырылып отырған мәселенің кейбір маңызды аспектілері қалып қойған	Тізім осы тақырып бойынша оқушыға қолжетімді барлық негізгі дереккөздерді қамтиды
Зерттеу тереңдігі	Жұмыс үстірт, иллюстративті, дереккөздері негізінен танымал	Жұмыс бір маңызды дереккөз негізінде құрылады, қалғандары — иллюстрация ретінде қолданылатын танымал әдебиет	Мәселені қарау мазмұнды деңгейде құрылады, бірақ қарау тереңдігі біршама	Мәселені қарау жеткілікті терең мазмұнды деңгейде құрылады
Рәсімдеу	Безендіру автордың өз логикасымен негізделген, мүлдем кездейсоқ сипатқа ие	Жұмыс құрылымы бар, бірақ қатал емес	Жұмыс жалпы келесі бағанда баяндалған талаптарға сәйкес келеді, бірақ кейбір кемшіліктері бар немесе талаптардың біреуі орындалмайды	Жұмыс тақырыптың логикасымен негізделген нақты құрылымы, дұрыс рәсімделген әдебиеттер тізімі, дұрыс жасалған сілтемелер мен мазмұны бар (мазмұны)

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

§34–35 Деректер қорының негізгі ұғымдары

Бүгінгі сабақта:

- ▶ «реляциондық деректер қоры» ұғымын түсіндіруді;
- ▶ «жол», «жазба», «индекс» терминдерінің анықтамасын құрастырып, деректер қорында бірінші кілтті анықтауды үйренесіндер.

Тірек ұғымдар:

- ▶ реляциондық деректер қоры;
- ▶ кілт өрісі.

Әрбір адам үнемі «деректер қорымен» кездеседі. оған түрлі каталогтер, энциклопедиялар және т.б. жатқызуға болады. Мысалы, «жазба кітапшасы» дерекқорында адамдар туралы ақпарат, олардың әрқайсысында өзінің аты-жөні, телефон нөмірі және т.б. бар.

Деректер қорында ақпарат тапсырыс түрінде сақталады. Сондықтан жазу кітапшасында барлық жазбалар алфавит бойынша сұрыпталады, ал кітапхана каталогінде — алфавит (алфавиттік каталог) немесе білім саласы бойынша (пәндік каталог) реттеледі.



Деректер қоры — деректерді жүйелі түрде бірдей сипаттамалары бар нысандар тобында сақтауға мүмкіндік беретін ұйымдастырылған құрылым.

Деректер қорының қалай ұйымдастырылғанын қарастырайық. Онда көптеген ұқсас нысандар туралы ақпарат бар. Сонымен қатар әрбір нысан үшін кейбір белгілердің мәні ғана маңызды болады («белгі» термині кейбір белгілер мен қасиеттерді белгілеу үшін қолданылады).

Мысалы, мектеп кітапханасында (34.1-кесте) деректер базасының нысандары: көркем және техникалық әдебиет, газеттер мен журналдарға жазылым және т.б., ал олардың белгілері: жанр, кітаптың атауы, автордың аты-жөні, жарияланған жылы, беттердің саны, кітаптың болуы немесе оқырмандардың бірінің қолында болуы және т.б.

«Мектеп кітапханасы» деректер қорының үзіндісі

№	Автордың аты-жөні	Атауы	Шыққан жылы	Бет саны
1054	Әуезов М.	Абай жолы. 1-том	1942	315
1298	Куприн А. И.	Анар білезік	1980	320
3762	Тургенев И. С.	Көктем суы	1986	560

Деректер қорының ерекшелігі — бұл қорға енген барлық нысандардың белгілерінің саны бірдей. Кейде кейбір белгілердің мәні болмауы да мүмкін (мысалы, сынып оқушыларының телефондарының деректер қорын құрастырғанда, кейбіреуінің телефон деген тұсында сызықша тұрады, яғни мәні жоқ деген сөз).

Белгілердің біразын көбінесе *кілт* деп атайды. Бұл маңызды, өйткені бұдан өрі қарай кілт бойынша сұрыптау жүргізуге болады.

Кейбір белгілер міндетті түрде болуы керек, мысалы, кітапханадағы барлық кітаптың мүлктік нөмірі мен коды.

Әдетте, деректер қоры адамдардың көбісі үшін қолжетімді болып табылады, бірақ олардың арасында бүкіл деректер қорында қол жеткізе алатын және бір уақытта оған өзгерістер енгізе алатын бір ғана адам болады.

Деректерден басқа бұл қорда ұйымның әр қызметкеріне өзінің құқығы бар деректермен ғана жұмыс істеуге мүмкіндік беретін әдістер мен құралдар да бар.

Деректер қорындағы барлық ақпарат реттелген. Қордағы деректерді басқара білу үшін деректер қорын басқару жүйесі енгізілген жеке бағдарлама қажет.



Деректер қорының басқару жүйесі (ДҚБЖ) — деректерді сақтауға, іздеуге және өңдеуге көмектесетін бағдарламалық қамтамасыз ету болып табылады.

Қазіргі уақытта сәулет, ішкі бағдарламалау тілі, жұмыс істейтін операциялық жүйе, деректер сақталатын және өңделетін және басқа да сипаттамалардан ерекшеленетін бірнеше ДҚБЖ классификациясы бар. Ең танымал реляциялық ДҚБЖ: Access, FoxPro, Paradox. Бірнеше серверлерде таратылатын үлкен деректер қорымен жұмыс істеу үшін жасалған (серверлер әртүрлі салаларда болуы мүмкін) аса күрделі болып бөлінеді. Осы типтегі ДҚБЖ — Oracle, Informix.

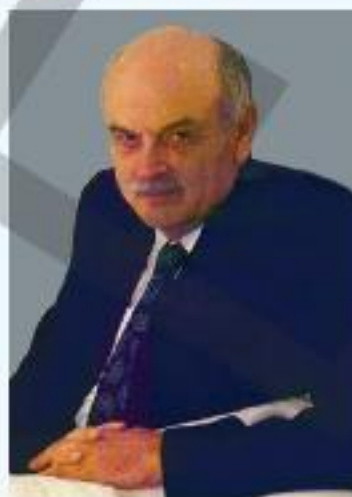
Деректер қорының түрлері. Деректер қорының бірнеше түрі бар: *реляциялық (кестелік), иерархиялық және желілік.*

Реляциялық (кестелік) деректер қоры. Қазіргі уақытта реляциялық деректер базасына артықшылық беріледі. Relational (қатынасы) екі өлшемді кестелерді (қарым-қатынастарды) қамтиды.

БҰЛ ҚЫЗЫҚ!

Реляциялық деректер базасы моделі өткен ғасырдың 60-жылдарының аяғында ұсынылған. Негізгі тұжырымдамасына сәйкес салыстырмалы деректер қоры – екі өлшемді өзара байланысты кестелер жиынтығын қамтитын деректер қоймасы.

Мичиган Университетінің (АҚШ) докторы Э.Кодд реляциялық деректер қоры мен байланыс теориясы дамуына үлкен үлес қосты.



Эдгар Кодд



Екі өлшемді кестелер дерек терминологиясында жазылған жолдардан және өрістер деп аталатын бағаннан тұрады.

Үлгі ретінде «Тауарлар» кестесін қарастырайық, оның әрбір жолында жеткізілетін тауардың біреуі туралы мәлімет бар.

34.2-кесте

Тауарлар

	1-өріс	2-өріс	3-өріс	4-өріс	5-өріс
Жазу	Код	Жабдықтаушы	Тауар түрі	Өлшем бірлігі	Бағасы, теңге
1	1	ЖК Ахметов	Вафли	Жәшік (5 кг)	360
2	2	ЖАҚ «Батыр-Баян»	Мармелад	Жәшік (10 кг)	420
3	3	ЖСШ «Азия»	Зефир	Жәшік (8 кг)	350

34.2-кестенің бағандарында деректер нысанының негізгі сипаттамалары жазылады. Әрбір өріс біртекті болып табылады, яғни ондағы деректердің түрі мен ұзындығы бірдей. Әрбір кесте өрісінің бірегей атауы бар. Мәні сәйкес жазбаны бірегей түрде анықтайтын өріс *кілт өрісі* деп аталады.

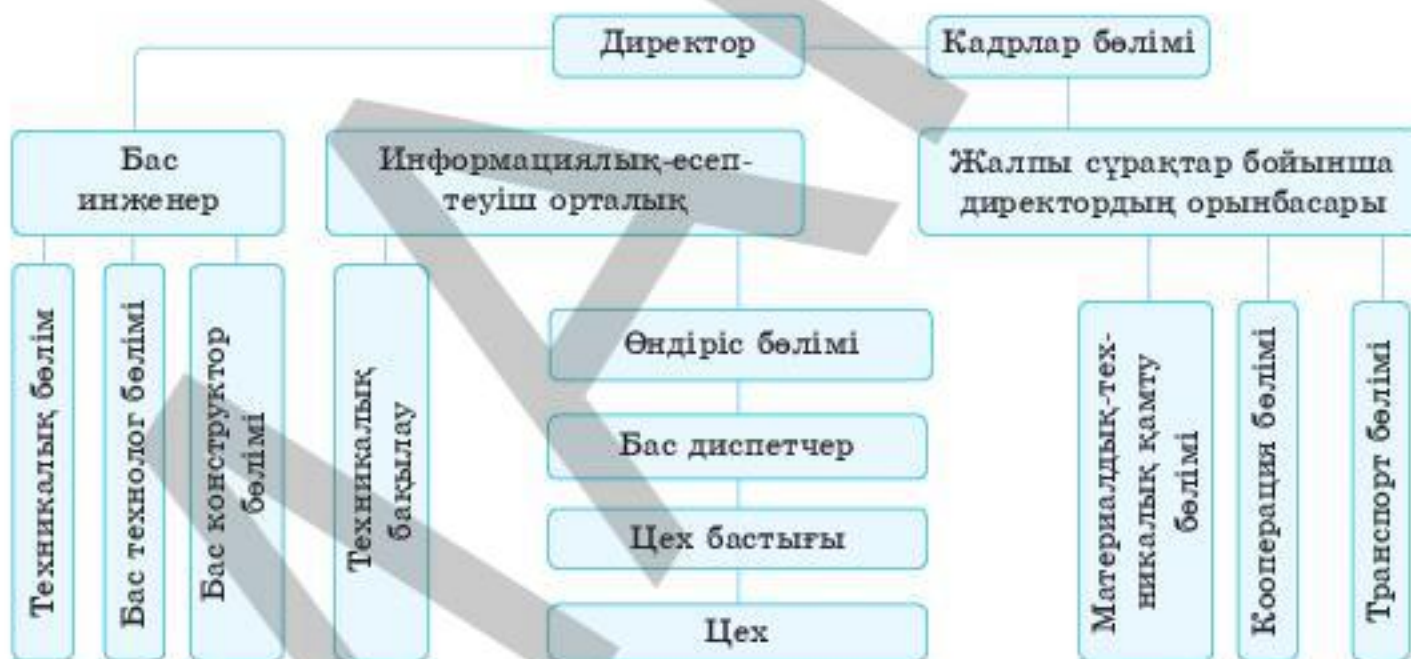
Бұл кестенің «КОД» өрісінде кез келген жазбаларда қайталанбайтын бірегей мәндер бар екенін ескеріңдер, бұл кілт өрісі мәртебесіне ие болады. «Өлшем бірліктері», «Бағасы» (сәйкестік болуы мүмкін) немесе кез келген басқа өріс негізгі өріс рөліне сәйкес келмейді.

Егер кілт өрісі біреу болса, онда бұл қарапайым кілт болып табылады, егер бірнеше негізгі өрістер болса, онда бұл — құрамды кілт.

Иерархиялық деректер қорының графикалық түрде түрлі деңгейдегі нысандардан тұратын төңкерілген ағаш ретінде ұсынылуы мүмкін. Құрылымның ең жоғары деңгейінде — жоғары түйіндері жоқ ағаштың түбірі. Қалған түйіндер (кейін пайда болған) бір-бірімен байланысты және олардың әрқашан жоғары тұратын бастапқы түйіні болады.

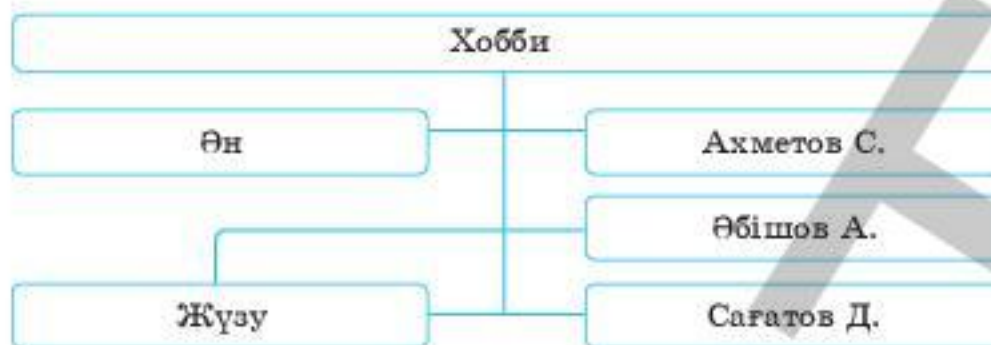
Мысал ретінде иерархиялық кәсіпорын моделін қарастырайық (34.1-сурет), ол мыналарды көрсетеді:

- өндірістегі ұйымдастыру қатынастары;
- жауапкершілікті бөлу;
- операциялардың тәртібі.



34.1-сурет. Иерархиялық деректер моделі

Желілік деректер қоры. Желілік деректер қоры иерархияны жалпылау болып табылады және әрбір элемент кез келген басқа элементпен байланыстырылуы мүмкін ағаш құрылымы түріндегі деректерді ұйымдастыруды қамтиды (34.2-сурет).



34.2-сурет. Желі деректерінің үлгісі

Иерархиялық және желілік құрылымның кемшілігі: жаңа файлдарды қосу немесе жаңа байланыстар орнату кезінде деректерді жоғалту қаупі бар.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Деректер қоры дегеніміз не?
2. ДҚБЖ дегеніміз не?
3. Реляциялық деректер қоры дегеніміз не?
4. Реляциялық деректер үлгісінің мысалын келтіріңдер.
5. Иерархиялық және желілік деректер моделі арасындағы айырмашылық қандай?
6. Иерархиялық және желілік деректер моделіне мысалдар келтіріңдер.

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

1. «Салымшылар» деректер базасының желілік құрылымы келтірілген (34.3-сурет). Бұл құрылымды кесте түріне айналдырыңдар.



34.3-сурет. «Салымшылар» деректер қоры

В ДЕҢГЕЙІ

2. «Менделеев кестесінің» деректер базасының иерархиялық құрылымы келтірілген (34.4-сурет). Бұл құрылымды кесте түріне айналдырыңдар.



Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

§ 36 Деректер қорындағы алғашқы кілт

Бүгінгі сабақта:

Тірек ұғымдар:

- ▶ деректер қорында алғашқы кілтті анықтауды үйренесіндер.

- ▶ деректер қоры;
- ▶ реляциялық деректер қоры;
- ▶ еріс;
- ▶ жазба;
- ▶ индекс;
- ▶ алғашқы кілт.

Алдыңғы тарауда қарапайым деректер қорының құрылымын бағандар мен жолдардан тұратын тіктөртбұрышты кесте ретінде ұсынуға болатынын, сондай-ақ тік бағандар — *өрістер*, ал көлденең жазбалар деп аталатынын білдіңдер.

Сақталатын ақпараттың бірлігі көлденең жол — бір нысан туралы ақпаратты сақтайтын (оқушы, кітап, тауар және т.б.) жазба болады.

Әрбір жазба *кілт* деп аталатын бір өрістің басқа мәнінен ерекшеленеді.



Деректер қорының алғашқы кілті — мәні өртүрлі жазбаларда қайталанбайтын өріс (немесе өрістер жиынтығы).

Егер алғашқы кілт бір өрістен тұрса, ол *қарапайым* деп аталады, бірнеше өрістен тұрса, *құрама кілт* деп аталады.

36.1-кестеде кілтті таңдау мүмкін емес, өйткені әрбір өрістің мәні қайталанатын. Бірақ өртүрлі жазбаларда «күн» және «уақыт» екі өрістің мәндері бір уақытта сәйкес келе алмайды. Олар кестенің құрамдас кілтін құрайды. Әрбір өрістің типі мен форматы болады.

36.1-кесте

«Ауа райы» деректер қорының мысалы

Күні	Уақыты	Температура	Ылғалдығы	Желдің жылдамдығы
21.01.2019	12:00	-28	65	6
21.01.2019	16:00	-26	00	5
...

Өрістің типтері:

1. *Символдық* — бұл типті өріс символдар жолы ретінде қарастырылатын және өріптерден, цифрлардан, тыныс белгілерінен және т.б. тұруы мүмкін ақпаратты сақтауға арналған.

2. *Сандық* — бұл түрдегі өрістер тек сандарды сақтауға арналған.

3. *Мерзімі* — осы типті өрістер қандай да бір күнді белгіленген форматта сақтауға арналған: күні, айы, жылы.

4. *Логикалық* — бұл түрдегі өрістер «иә-жоқ» немесе «ақиқат-жалған» түрінің мәндерін сақтауға арналған.

Өрістің форматтары:

Символдық өрістің форматы жазбадағы өрістің алатын символдық позицияларының санын анықтайды. Мысалы, егер символдық өріс форматы 10 болса, онда оның өртүрлі жазбалардағы мәндері 0-ден 10 символға дейін болуы мүмкін.

Сандық өрістің форматы, әдетте, екі бөліктен тұрады: ұзындығы мен дәлдігі.

Ұзындығы — бұл сан жазуға берілетін символдық позициялардың толық саны.

Дәлдік — бөлшек бөлшекке бөлінген позициялар саны. Ондық нүкте де орынды алатынын ескеру керек. Мысалы, 123.45 санын жазу түрі: ұзындығы — 6, дәлдігі — 2. Бүтін сан, яғни бөлшек бөлігі жоқ санның 3 дәлдігі бар.

Логикалық шаманың форматы стандартты — 1 символ. Көбінесе келесі белгілер қолданылады: 1 — (ақиқат), 0 — (жалған).

Мерзім форматы, әдетте, 8 таңбаның ұзындығына ие. КК/АА/ЖЖ (немесе КК.АА.ЖЖ, немесе КК-АА-ЖЖ. Мұндағы КК — санның екі таңбалы белгісі, АА — ай, ЖЖ — жыл. Кейде КК/АА/ЖЖ стандарты қолданылады. Басқа белгілері де болады.

Осылайша өрістердің мәндері — белгілі бір түрлердің кейбір шамалары. Шаманың түрі онымен қандай әрекеттер жасауға болатынына тәуелді. Мысалы, сандық шамалармен арифметикалық операцияларды орындауға болады, ал символдық және логикалық операцияларды орындауға болмайды.

БАҚЫЛАУ СҰРАҚТАРЫ



1. «Өріс», «жазба» сөздерінің мағыналарын түсіндіріңдер.
2. Алғашқы кілт деген не?
3. Қарапайым кілттің құрама кілттен қандай айырмашылығы бар? Мысал келтіріңдер.
4. Өрістердің қандай типтері мен форматтарын білесіңдер?
5. Кез келген мәселелерді шешу үшін күнделікті өмірде дерек қорын пайдалану мүмкіндігіне мысал келтіріңдер.

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Төменгі деректердің қайсысы кілт болуы мүмкін?

- тегі;
- аты;
- жеке куәлігінің нөмірі;
- үйдің нөмірі;
- машинаның тіркеу нөмірі;
- қала;
- электрондық пошта адресі;
- жұмысты орындау күні;
- кіржуғыш мәшине маркасы.

В ДЕҢГЕЙІ

Ауа райы өрісінің типін анықтаңдар.

Күні	Жауын-шашын	Температура, °С	Қысым, мм сын бағ	Ылғалдығы, %
15.03.2019	Қар ***	-3,5	746	67
16.03.2019	Жауын-шашын-сыз	0	750	62
17.03.2019	Қар ****	1,0	746	100

С ДЕҢГЕЙІ

Кез келген пәндік аймақты (әлем елдері, әлем астаналары, сыныптастар, жануарлар, үй шығындары және т.б.) таңдаңдар, деректерді құрылымдаңдар (өрістерді таңдау қажет), деректер типін анықтаңдар. Мәліметтер типін, өріс кілтін анықтап, бірнеше мысал келтіріп, кестені толтырыңдар. Деректер қорында іздеуді қандай критерийлер бойынша жүргізуге болады.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 37–38 Деректер қорын жасау

Бүгінгі сабақта:

- ▶ деректер қорында деректер типін анықтауды;
- ▶ біркестелі деректер қорын жасауды;
- ▶ көпкестелі деректер қорын жасауды үйренесіңдер.

Тірек ұғымдар:

- ▶ деректер қоры;
- ▶ реляциялық деректер қоры;
- ▶ өріс;
- ▶ жазба;
- ▶ индекс;
- ▶ алғашқы кілт.

Кесте деректер қорының басты нысаны болып табылады. Деректер қоры бір немесе бірнеше кестеден тұруы мүмкін. Кестелер негізінде деректер қорының нысандарының қалған бөлігі жасалады.

Деректер қорын жобалау үшін:

1. *Деректер қорын құрудың мақсатын анықтау.*

Құрылымды жасамас бұрын, өзіңіз жасайтын деректер қорының мақсаты (сұрақ қойылған тақырыпты сипаттаңдар), қандай функциялар және қандай ақпарат болуы керек деген сұрақтарға жауап беру керек. Демек, деректер қорының кестелерінің негізгі тақырыптарын және кесте өрістері қамтылатын ақпаратты анықтау қажет.

2. *Деректер кестелерін анықтау.*

Деректер қорын жасау процесінде ең күрделі сатылардың бірі кестелерді жасау болып табылады, өйткені деректер қоры берген нәтижелер (есептер, шығарылым пішіндері және т.б.) кестенің құрылымы туралы толық түсінікті әр кезде бере бермейді.

Кестелерді жобалау кезінде келесі принциптерді орындау ұсынылады:

— кестедегі ақпарат қайталанбауы және кестелер арасында қайталау болмауы керек;

— егер белгілі бір ақпарат бір кестеде болса, онда ол тек бір жерде өзгертілуі керек;

— кестеде бір тақырып бойынша ақпарат болуы керек.

3. *Кестеде талап етілетін өрістерді анықтау.*

Әрбір кестеде бөлек тақырып бойынша ақпарат бар және кестедегі әрбір өріс кесте тақырыбы бойынша жеке ақпаратты қамтиды.

4. *Кілттерді орнату және кестелер арасындағы қатынастарды анықтау.*

Әртүрлі кестелердің деректерін байланыстыру үшін әр кестеде кілт өрісі болуы керек. Бұл өрістің мәні кестеде қажетті жазбаны таңдауға, сондай-ақ кестелер арасындағы қатынастарды дұрыс орнатуға мүмкіндік береді.

5. *Деректер қорының нысанын құру және енгізу.*

Егер кесте құрылымдары талаптарға сәйкес келсе, барлық деректерді енгізуге болады.

Содан кейін кез келген сұрауларды, пішіндерді, есептерді, макрос-тарды және модульдерді жасай аласыңдар.

Мысал қарастырайық. Пайдаланушының жеке кітапханасынан CD және DVD дискілерін пайдалану туралы ақпаратты қамтитын «Диск кітапханасы» деректер қорын жасау керек болсын.

«Дискілік кітапхана» деректер қорын пайдалану кезінде басты міндет — барлық мүдделі клиенттерге дискілерді жеткізуді бақылау.

Ақпаратты сақтауға қажетті ақпараттарды талдағаннан кейін, алдымен оны үш топқа бөлеміз: дискілердің клиенттері туралы ақпарат, дискілердің бар болуы туралы ақпарат және дискілерді шығару туралы ақпарат. Бұл жағдайда деректер қоры кестесінің құрылымы келесі формаға ие болады (37.1-кесте).

Деректер қорының «Диск кітапханасы» кестесінің құрылымы

Өрістің атауы	Деректер типі	
Тегі	Мәтіндік	Диск клиенттері туралы ақпарат
Аты	Мәтіндік	
Мекенжайы	Мәтіндік	
Телефон нөмірі	Мәтіндік	
Электрондық поштасы	Мәтіндік	
Диск атауы	Мәтіндік	Диск туралы ақпарат
Диск түрі	Мәтіндік	
Диск құны	Қаржылы	
Берілген күні	Күні/уақыты	Дискі шығару туралы ақпарат
Төлемді қайтару	Логикалық	

Мұндай кестемен жұмыс істеу өте қолайсыз. Мысалы, бірнеше клиентке бірнеше дискілер шығарылғанда, ол туралы бірнеше мәрте қайталау керек: аты-жөні, мекенжайы т.б. Бұдан кестенің көлемі ұлғаяды және ақпарат енгізгенде қателердің ықтималдығы артады.

Осы кестенің әрбір жазбасын толтырған кезде, кейбір өрістердің нақты бір клиент үшін бір рет қана қолданылатынын байқаймыз, мысалы, таңдалған дискінің атауы; ал басқалары бірнеше реттен қолданылады: аты-жөні, клиенттің мекенжайы. Бұл нақты бір клиентті көру үшін өртүрлі бірнеше дискіні бірден таңдап алуына байланысты.

Жасалған деректер қорымен жұмыс істеу кезінде тиімділікті арттыру үшін қалыпқа келтіру процесін орындау қажет.

Ол үшін бір кестені үш кестеге 37.2, 37.3, 37.4 бөлеміз: «Клиенттер», «Дискілер», «Дискілер шығару».

Әр кестенің құрылымын сипаттайық.

Клиент

Атауы	Деректер типі
Клиент коды	Сандық
Тегі	Мәтіндік
Аты	Мәтіндік
Мекенжайы	Мәтіндік
Телефон нөмірі	Сандық
Электрондық поштасы	Мәтіндік

Дискілер

Өрістің атауы	Деректер типі
Диск нөмірі	Сандық
Жолақ атауы	Мәтін
Диск түрі	Мәтін
Дискінің құны	Ақша

Дискілерді шығару

Өрістің атауы	Деректер типі
Беру коды	Сандық
Клиент коды	Сандық
Диск коды	Сандық
Берілген күні	Күні / уақыты
Төлемді қайтару	Логикалық

Ұсынылған кестелерде алғаш рет *Есептегіш* деректері берілген.

Ол әрбір жаңа жазбаға ауысқан кездегі қадамдар санының бүтін сандық мәндерін сақтау үшін қолданылады. Есептегіш кесте жазбасының бірегей сәйкестендіру ретінде пайдаланылуы мүмкін, оған басқа ешқандай мән берілмейді. Біздің жағдайда *клиенттің коды*, *диск коды* және *беру коды* бір клиентті басқасынан, бір дискіні екіншісінен ажырататын және әрбір дискінің мәселесіне бірегей ерекше мәндерді тағайындайтын бірегей сәйкестендірулер болады.

Клиент коды, *диск коды* жолдары кейінгі параграфта қарастырылатын кестелер арасындағы байланыстарды орнату үшін пайдаланылады.

Деректер қорымен жұмыс істеу үшін Microsoft Access қолданамыз. Ұқсас тәсілдер OpenOffice.org пакетінен тегін Base ДҚБЖ қолданылады (сондықтан практикалық тапсырмаларды кез келген программаларда орындауға болады).

Microsoft Access ДҚБЖ барлық деректер қоры кеңейтілуі mdb болатын бір файл түрінде болады. Онда:

— *кесте* — деректерді сақтауға арналған;

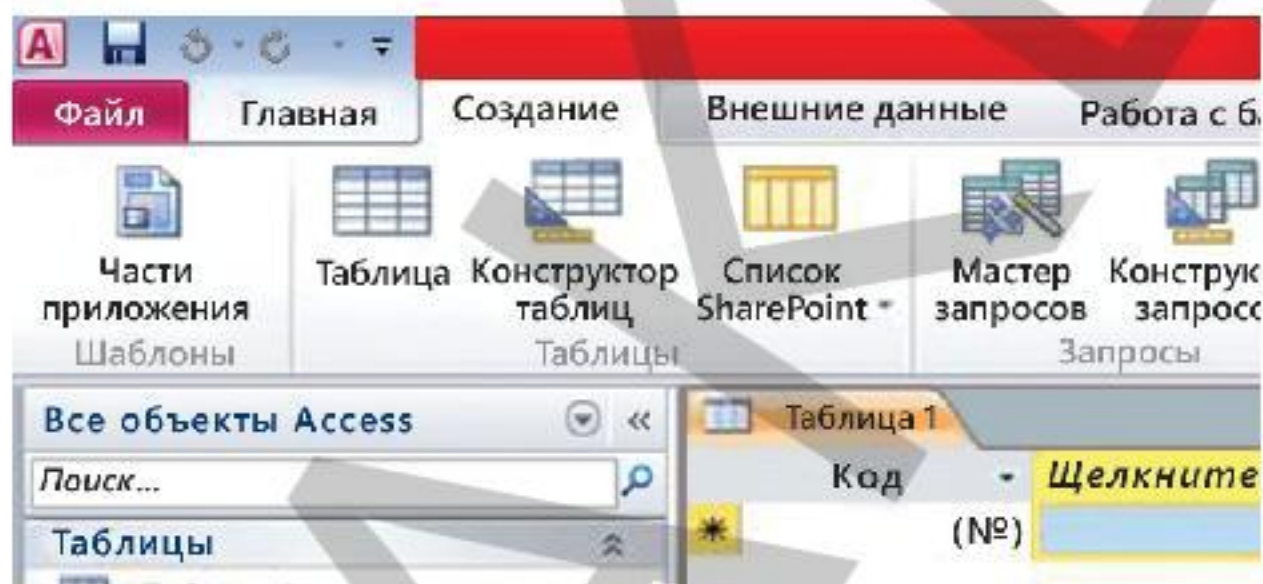
— *форма* — деректерді енгізуге және өзгертуге болатын сұхбат терезесі;

— *сұраныс* — нәтижесінде қажетті деректерді іріктеу немесе жазбаларды өзгерту немесе жою сияқты басқа әрекеттер орындалатын деректер қорына хабарлама беру;

— *есеп* — деректерді басып шығаруға арналған құжаттар үлгісі.

Кестені дайындамас бұрын Access пәрмендер тізбегінің көмегімен жаңа деректер қорын жасау керек: *Файл* → *Жаңа деректер қоры файлын құру* (*Файл* → *Создать* → *Новая база данных*). Жаңа деректер қоры терезесінде қор файлының атын жазу, мысалы, *Кітапхана* (*Библиотека*) → *Ашу* (*Открыть*).

ДҚБЖ Access бағдарламасы кесте құрылымын *Конструктор*, *Шебер* (*Мастер*) немесе *Кесте* (*Таблица*) режимінде (деректерді енгізу арқылы) үш жолмен жасауға мүмкіндік береді (37.1-сурет).



37.1-сурет. Кесте құрылымын таңдау

Шебер (*Мастер*) және *Конструктор* режимдерінің жұмысын қарастырайық. Кестелер режимінде деректерді енгізу бос кестеде орындалады.

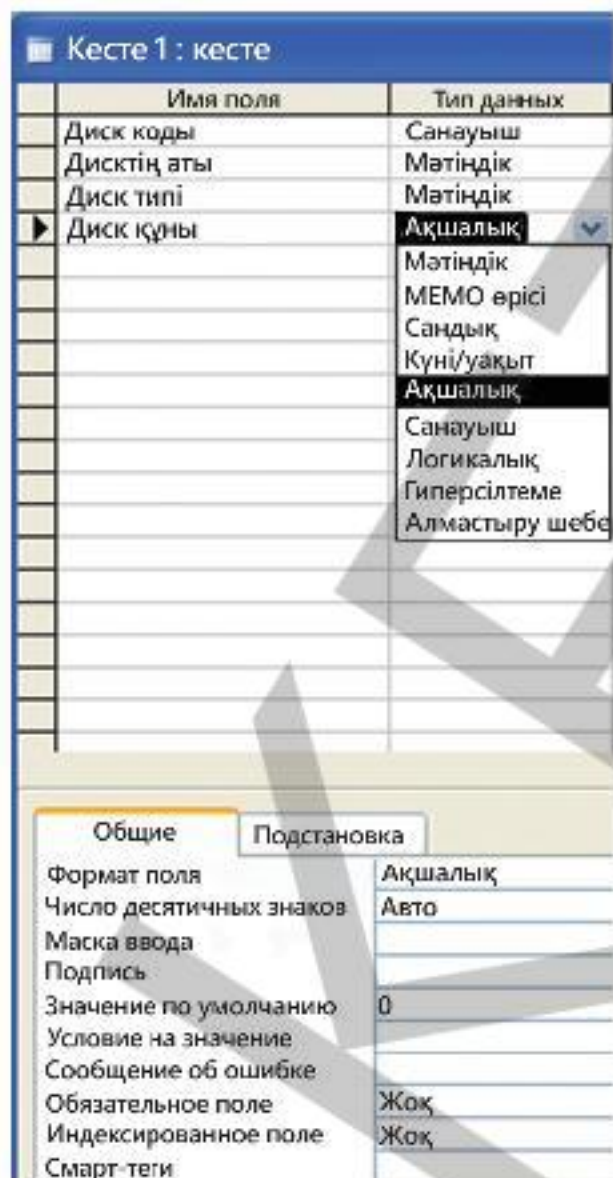
Дизайн көрінісінде кесте құрылымын жасау. Дизайн көрінісінде кесте жасау үшін:

1. *Кестелер* (*Таблица*) → *Құру* (*Создать*) → *Конструктор* режимін таңдаңдар. Содан кейін, *Конструктор* терезесі пайда болады.

2. Өріс атауларын теріндер және деректер типін, сипаттарын (өлшемі т.б.) көрсетіндер.

3. Сипаттама бөлімінде түсіндірмелерді енгізуге болады.

Біздің мысалда «Дискілер» кестесінде әрқайсысының атауының, деректер түрі және терезенің төменгі жағында енгізу қажет өрістердің төрт жолы болуы керек (37.2-сурет).



37.2-сурет. Кесте дизайнерінің терезесі

Кестенің деректер жинағынан кейін *Файл* → *Сақтау (Сохранить)* → *Кілт жолын жасау (Создать ключевое поле)* → *Иә (Да)* деп жауап беріңдер.

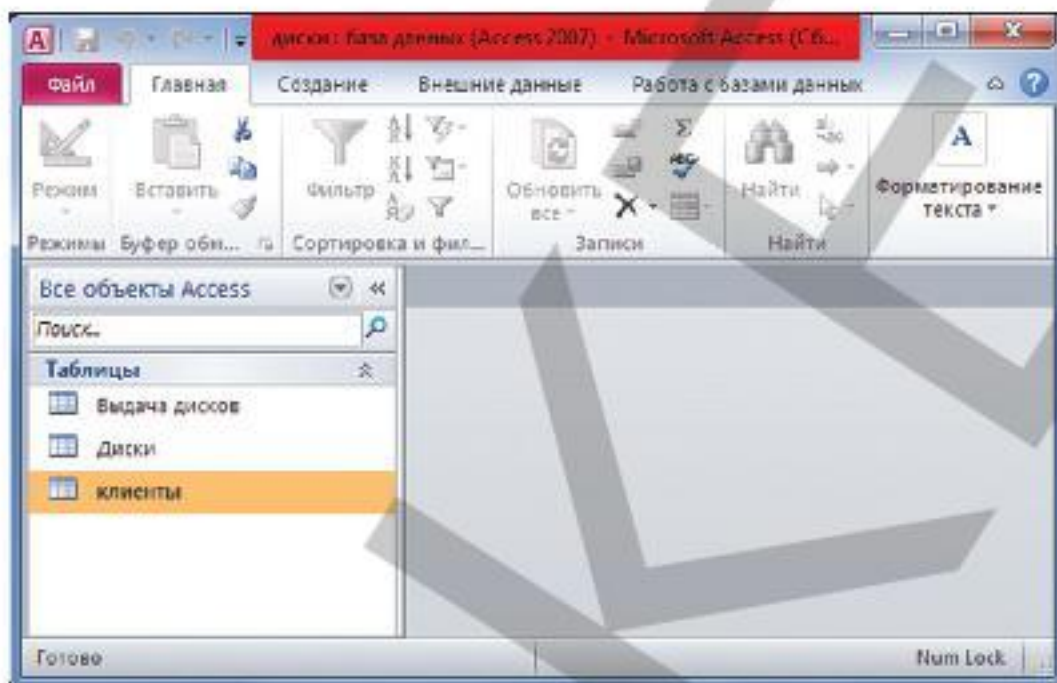
Назар аударыңдар, белсенді өріс (онда қазіргі уақытта курсор орнатылған) сол жақта индикатормен — үшбұрышты бағыттамамен белгіленеді. Белсенді өрістің сипаттары диалогтің төменгі жағында көрсетілген (37.5-сурет).

Деректер типі. Деректер типі мәтін немесе сан сияқты өріске (бағанға) енгізілетін мәндермен анықталады. Access бағдарламасында келесі деректер типін пайдалануға рұқсат етілген:

- мәтіндік;
- МЕМО (үлкен әріптермен мәтіндер енгізу);
- сандық — тұтас, бөлшек, ондық;
- ақша;
- күні / уақыты;
- санауыш — кестенің бірегей жүйелік кілті үшін пайдаланылады;

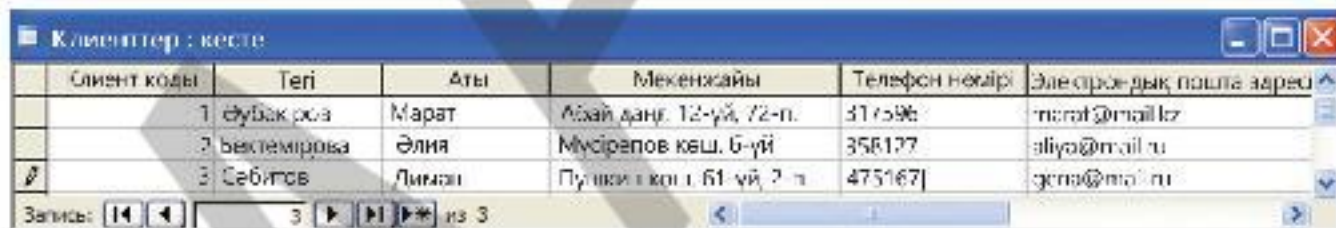
— логикалық — екі мән қабылданады: «иә» — «жоқ» немесе «ақиқат» — «жалған».

Кестедегі деректерді енгізу және өңдеу. Кесте құрылымын жасау жұмысын аяқтағаннан кейін пайдаланушы кестені *Кестелер (Таблица)* режимінде деректермен толтыра бастайды. Бұл әрекетті орындау үшін кестенің атын деректер қорының терезесінен екі рет басыңдар, мысалы, «Клиенттер» немесе кестені басу арқылы *Ашу (Открыть)* батырмасын басыңдар (37.3-сурет).



37.3-сурет. Деректер қорының терезесі

Деректер базасының кестесінде деректерді енгізу және редакциялау кестенің тиісті ұяшығында тікелей орындалады. Енгізу мен өңдеу әрекеттері Word мәтіндік процессор кестесіндегі деректерді өңдеу мен толтыру әрекеттеріне ұқсас (37.4-сурет).



37.4-сурет. «Клиенттер» деректер қорына деректерді енгізу

Деректерді енгізу мен өңдеу кезінде кестедегі жазбаның сол жағына орналасқан *Жазбалар маркері* өрісінде арнайы белгілер болады:

	Белсенді жазба
	Кестенің соңындағы бос жазба
	Өзгерген жазба

Кестедегі жолды жою қажет болса, осы жолды конструктор терезесінде басып, *Өңдеу (Правка) → Жолдарды жою (Удалить строки)* батырмасын басыңдар. Жаңа жолды енгізу үшін қосылған жолдың астындағы өрісті жандандыру керек және *Кірістіру (Вставка) → Жолдар (Строки)* мәзірінің пәрменін орындаңдар.

Кестеге деректерді енгізуді немесе кестені өңдеуді аяқтағаннан кейін *Файл → Сақтаушы (Файл → Сохранить)* пәрменін орындаңдар. Кесте режимінде деректер енгізілгенде, ол автоматты түрде сақтайды.

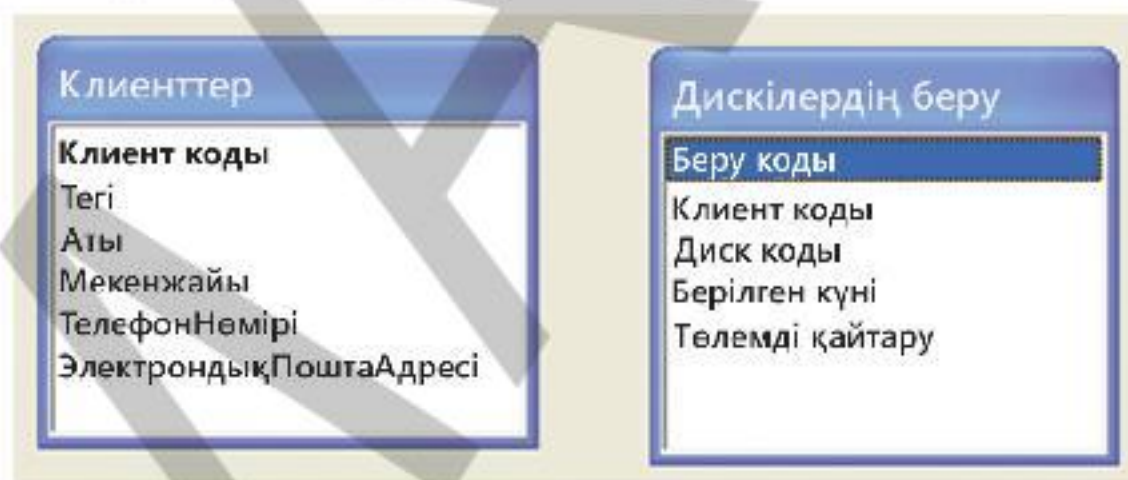
Деректер қорының кестелерін байланыстыру үлгісі. Ассес бағдарламасында байланыстарды орнату өртүрлі кестелердегі деректерді автоматты түрде байланыстыруға мүмкіндік береді, осылайша деректер қорының толықтығы қамтамасыз етіледі.



Кестелер арасындағы қатынасты орнатуды деректер қорының сұлбасын құру деп атайды.

Кестелер арасындағы қатынасты орнату процесі *деректер қорының сызбасы* деп аталады. Екі кестенің арасындағы қатынасты орнату үшін олардың әрқайсысындағы байланыстыруға арналған өрістерді анықтау керек. Бұл өріс атауының бірдей болуы міндетті емес, бірақ деректері бірдей болу керек.

Мысалы. Бұрын құрылған «Диск кітапханасы» деректер қорының «Клиенттер» және «Дискілерді беру» кестелерінің өзара байланысын қарастырыңдар. Әр кестенің құрылымы 37.5-суретте көрсетілген.

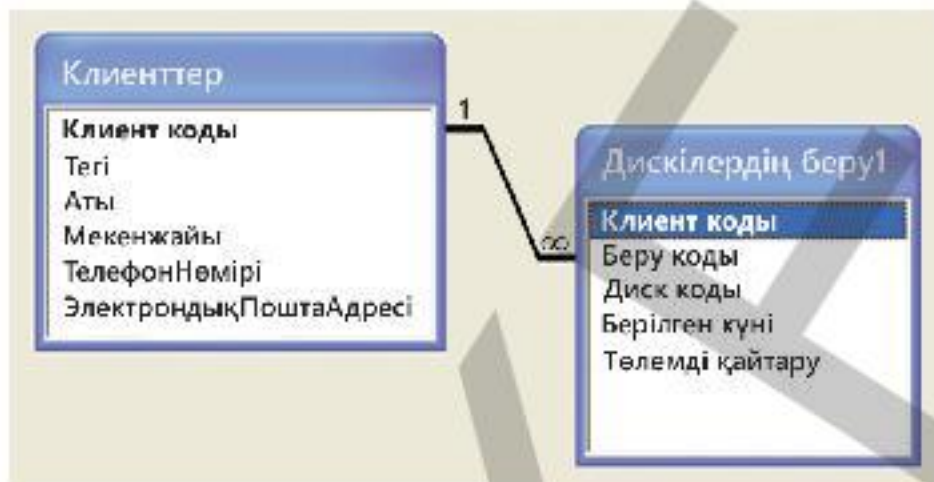


37.5-сурет. «Клиент», «Диск» кестесінің құрылымы

«Клиент» кестесінде «Клиенттік коды» өрісі кілт өрісі болып табылады. Бұл өріс есепші болып табылады және осы кестенің әрбір жазбасы үшін бірегей мәні бар. Осы кестенің өрісі бастапқы кілт деп аталады.

Егер әрбір тұтынушы тек бір дискіні таңдау құқығына ие болса, «Дискілер беру» кестесінің негізгі өрісі бірінші кестедегі «Клиент коды» өріс есептегіші сияқты болуы мүмкін. «Дискілер беру» кестесінде кілттік өріс «Шығару коды сыртқы кілт» деп аталады.

Бұл жағдайда алғашқы және сыртқы кілттер арасында орнатылған байланыс түрі «бірге бір» байланысы деп аталады. Байланыстың бұл түрі 37.6-суретте көрсетілген.



37.6-сурет. Деректер қоры мен кестелер қатынасының мысалы



«Бірге бір» қатынасында негізгі кестедегі бір жазбаға байланысқан кестедегі бір жазба сәйкес келеді.

Егер «Дискілерді беру» кестесінде бір тұтынушы бірнеше тапсырыс беруге құқылы болса, *Клиент коды* өрісі ондай бірегей бола алмайды, өйткені бірнеше рет қайталануы мүмкін. Бұл жағдайда сол өрістегі деректер түрі сандық мәндерді қабылдай алады, ал *Беру коды* өрісі бірегей мәнді кілт өрісі болып саналады. Бұл жағдайда екі кестедегі *Клиент коды* деген аттас өрістер арасында орнатылған байланыс түрі 37.7-суретінде көрсетілгендей, «біреуі көпке» байланысы деп аталады.



37.7-сурет. Деректер қоры мен кестелер қатынасының арасындағы мысал




«Біреуі көпке қатынастар» деген бір жазбаның көптеген басқа кестедегі мәліметке сәйкестігін білдіреді.

Реляциялық деректер қорында кестелер арасында *көбіне-көп қатынастар* жасалуы мүмкін.

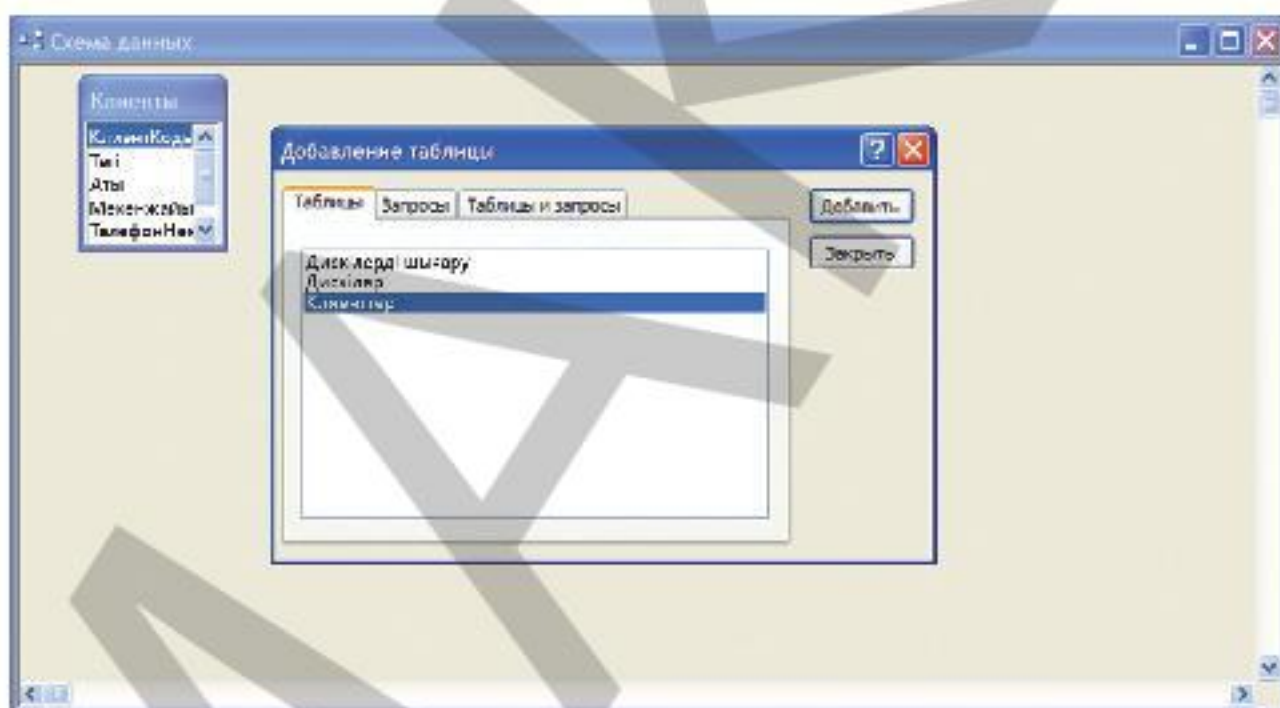
Деректер сызбасын (Схема данных) Access ДҚБЖ ішінде жасау үшін келесі әрекеттерді орындау қажет:

1. Кестелер арасында байланыстар орнатылған, мысалы, «*Диск кітапханасы*» деректер қоры бар көп кестелі деректер қорын ашыңдар.

2. *Деректер схемасы*  белгішесін басыңдар немесе *Стандартты* тақтасында келесі әрекеттерді орындаңдар: *Сервис* → *Деректер схемасы (Схема данных)*.

3. *Кесте қосу (Добавление таблицы)* терезесінде сілтемені орнататын бірінші кестені таңдаңдар. Мысалы, бұл кесте «Клиенттер».

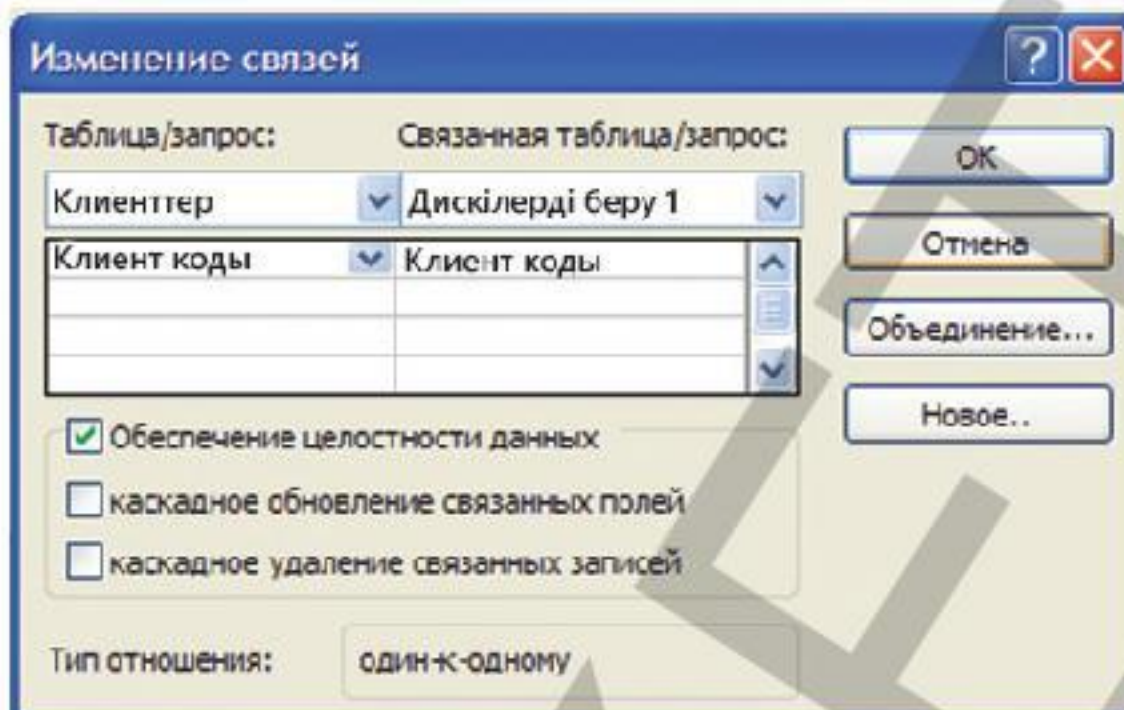
4. *Қосу (Добавить)* батырмасын басыңдар. Экранда 37.8-суретте көрсетілгендей, қосқан кесте көрсетіледі. Содан кейін басқа кестелерді *Деректер схемасына (Схема данных)* қосу керек, мысалы, «*Дискілерді шығару*» (*Выдача дисков*) және *Жабу (Закреть)* батырмасын басыңдар.



37.8-сурет. «Кестені қосу» диалогтік терезесі

5. Бірінші кестенің терезесінде қосылымды орнату үшін пайдаланылатын өрісті басыңдар, мысалы, *Клиент кодын* екінші кестенің сәйкес өрісіне апарыңдар.

Өзгерту сілтемелер (Изменение связей) терезесінде міндетті өрістер біріктіріліп, *Деректер тұтастығын тексеру (Обеспечение целостности данных)* жалаушасын қойып, *Жаңа (Создать)* батырмасын басыңдар (37.9-сурет).



37.9-сурет. «Өзгерту сілтемелері» диалогтік терезесі

Деректер схемасы (Схемы данных) терезесінде орындалатын әрекеттердің нәтижесінде екі байланыстырылған өрістер 37.8 және 37.9-суретте көрсетілгендей жолмен қосылады.

Болашақта жасалған сілтемелерді сақтау үшін *Сақтау (Сохранить)* батырмасын басып, *Деректер схемасы (Схемы данных)* терезесін жабындар. Құрылған сілтемелерді *Деректер схемасы (Схема данных)* терезесінде жою үшін сілтемеге нұқып, *Жою (Удалить)* батырмасын басындар. Сілтемені екі рет басқан кезде сілтемелерді өңдеу терезесі ашылады.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Деректер қорын жобалау кезеңдері қандай?
2. Кестелерді жасау кезінде қандай принциптерді қолдану керек?
3. Кесте құрылымын шебер режимінде қалай жасауға болады?
4. Кесте құрылымын жасау үшін конструктор режимі қалай пайдаланылады?
5. Кестеде негізгі өріс дегеніміз не? Негізгі өріс не үшін пайдаланылады?
6. Access ДҚБЖ-да қандай деректерге рұқсат етіледі?
7. Санаушы өрісі не үшін пайдаланылады?
8. Кестеге деректерді қалай енгізуге және өңдеуге болады?
9. Кестеге қалай қосылады және одан шығарылады?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Бір кестелік деректер қорын жасаңдар және өңдеңдер.

1. «Оқушы» деректер базасының деректер қорын құрыңдар, онда келесі өрістер бар: *коды, тегі, аты, мектебі, сыныбы, туған күні, салмағы.*
2. Кестенің бастапқы кiлтiн анықтаңдар.
3. «Туған күн» өрісінен кейін «бойы» өрісін құрылымға қосыңдар.

В ДЕҢГЕЙІ

Үлгіге сәйкес «Елдер» деректер қорын жасаңдар (37.10-сурет).

Елдер	Астанасы	Халқы	Ауданы (ақ км)	Тілі	Жер шарындағы
Аустралия	Канберра	19731984	7636853	ағылшын	Аустралия, Мұхит аралдары
Австрия	Вена	7700003	83855	неміс	Еуропа
Алжир	Алжир	32818500	238740	араб	Оңтүстік Америка
Аргентина	Буэнос-Айрос	38740807	2766390	испан	Оңтүстік Америка
Бельгия	Брюссель	9987003	20780	фламанд, француз, нидерланд	Еуропа
Бразилия	Бразилиа	182032604	850565	португалия	Еуропа
Великобритания	Лондон	60094648	244820	ағылшын	Еуропа
Вьетнам	Ханой	81624716	329560	вьетнам	Азия
Египет	Каир	74718297	1001450	араб	Африка
Үндістан	Дели	1049700113	3287590	хинди	Азия
Қазақстан	Нұр-Сұлтан	8693000	3500000	қазақ, орыс	Азия
Канада	Оттава	32207113	9976140	ағылшын, француз	Солтүстік Америка
Қытай	Пекин	1286975468	9596960	қытай	Азия
Люксембург	Люксембург	454757	2586	неміс, француз	Еуропа
Польша	Варшава	38527560	312585	поляк	Еуропа
Ресей	Мәскеу	144526278	17075400	орыс	Еуропа, Азия
АҚШ	Вашингтон	290342554	9563000	ағылшын	Солтүстік Америка
Тунис	Тунис	9924742	163610	араб	Африка
Украина	Киев	48355439	603700	украин	Еуропа
Франция	Париж	60160529	547030	француз	Еуропа
Чили	Сантьяго	15663216	756950	испан	Оңтүстік Америка

37.10-сурет. «Елдер» деректер қоры

С ДЕҢГЕЙІ

«Клиент», «Дискілер», «Дискілерді шығару» кестелерін қамтитын «Диск кітапханасы» деректер қорын құрыңдар (37.2, 37.3, 37.4-суреттер).

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 39–41 Деректер қорын әзірлеу. Жобалау жұмысы

Бүгінгі сабақта:

- ▶ деректер қорында деректер типін анықтауды;
- ▶ бір кестелі деректер қорын жасауды;
- ▶ көп кестелі деректер қорын жасауды үйренесіндер.

Тірек ұғымдар:

- ▶ деректер қоры;
- ▶ реляциялық деректер қоры;
- ▶ өріс;
- ▶ жазба;
- ▶ индекс;
- ▶ алғашқы кілт.

Адамзат ғасырдан-ғасырға ақпарат жинауға және сақтауға ұмтылды. Ежелгі дәуірдің жарқын мысалдары Майя тайпасының түйінді хаттары, тастағы суреттер, әкесінің ұлына және т.б. берілетін мұралар болып табылады. Өркениеттің дамуымен байланысты ақпарат көлемі бірнеше рет ұлғайды, демек, ақпаратты жинақтау, сақтау және қайта құру қажеттілігі артты, сонымен қатар оны сұрыптау қажеттілігі пайда болды. Осы мақсатта компьютерлік деректер қоры пайдаланылады, сондықтан мектеп қабырғасынан шығып, қажетті ақпаратты деректер қорында іздеуді және оның белгілі бір шарттары бойынша сұрыптауды білулерің керек. Ақпаратпен жұмыс істеудің бағдарламалық өнімдерінің бірі — Microsoft Access.

Жоба бойынша жұмыс 4-5 адамнан тұратын топтарда жүзеге асырылады. «Мектеп», «менің сыныбым», «аурухана», «кадрлар бөлімі» және т.б. мәліметтер базасын құру үшін бірнеше тақырыптарды таңдау ұсынылады.

Топтың әрбір қатысушысы өз функциясын орындайды:

Үйлестіруші-уақыт сақтаушысы — топтың барлық мүшелері арасындағы өзара іс-қимылды жүзеге асырады, өз міндеттерін атқаруын қадағалайды және топтың тапсырманы уақытында орындауына жауап береді.

Редактор-талдаушы топ беретін барлық ақпаратты өңдейді және талдайды.

Орындаушы топ алдына қойылған міндеттерді шешу үшін қажетті ақпаратты іздеуді жүзеге асыруды және осы міндетті шешеді.

Орындаушы-хатшы бүкіл топқа жазба жасайды және қойылған тапсырманы шешуге көмектеседі.

Оратор топ орындаған тапсырманы ұсына отырып, сынып алдында сөйлейді.

Барлық жұмысқа 3-сабақ беріледі (1-сабақ — жұмыспен танысу, топтарға бөлу, топта міндеттерді бөлу, жұмысты жоспарлау; 2-сабақ — жобаны құру бойынша жұмыс; 3-сабақ — жобаны қорғау). Осы уақыт ішінде қажет:

- деректер қорын әзірлеу кезеңдерін қайталау;
- өз тақырыбы бойынша презентацияны кезең-кезеңмен әзірлеу;
- оны компьютерде жасау;
- ақпаратты іздеу және сұрыптау мүмкіндіктерін қарастыру;
- жобаны қорғау нысанын ойластыру.

Жұмыс аяқталғаннан кейін өздерінің деректер қорымен таныстыру керек. Ол үшін кез келген ыңғайлы тәсілмен жасауға болады: деректер базасын көрсету және онда жұмыс істеу ережелері туралы айту. Жұмысты ұсыну барысында міндетті түрде жұмыстың қалай орындалғанын, қандай идеялардың пайда болғанын және кейбіреулерден неге бас тартқанын айту керек. Жобаны қорғаудың бағалау параметрлері:

- қол жеткізілген нәтиже (15 балл);
- безендіру (15 балл);
- ұсыныс (15 балл);
- сұрақтарға жауаптар (15 балл);
- зияткерлік белсенділік (10 балл);
- шығармашылық (10 балл);
- практикалық іс-әрекет (10 балл);
- топпен жұмыс істей білу (10 балл).

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

§ 42–43 Пішіндер

Бүгінгі сабақта:

- ▶ деректерді енгізуге арналған пішін жасауды үйренесіндер.

Тірек ұғымдар:

- ▶ деректер қоры;
- ▶ кестенің құрылымы;
- ▶ деректер типі;
- ▶ формалар құру.

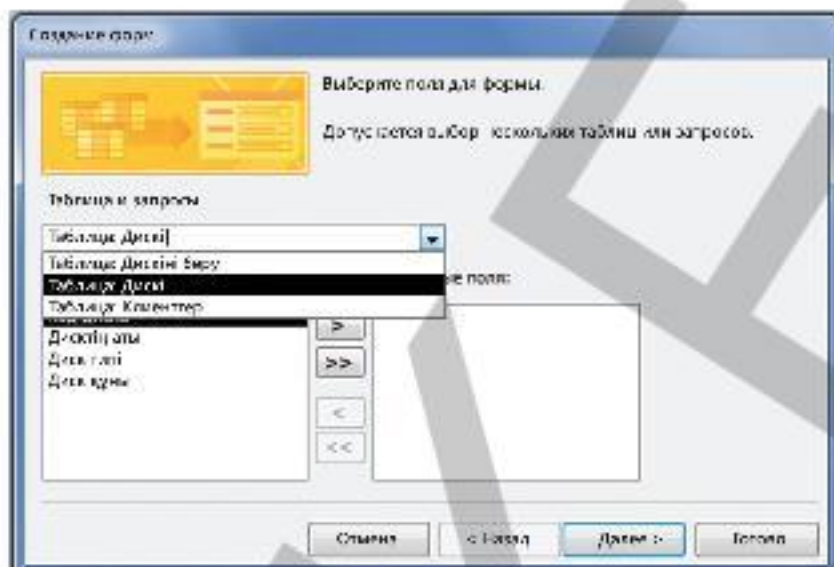
Пайдаланушының диалогтік интерфейсін ұйымдастыру үшін және кестеге деректерді енгізу мен қажетті жағдайда оларды өңдеу үшін *пішінді* (форма) қолдану ыңғайлы.



Пішін (Форма) — ақпаратты енгізу және көрсетуге арналған деректер қорының нысаны.

Пішін жасаудың бірнеше жолы бар. Пішіндер (Формы) қосымша бетін деректер қорының режімінде ашып, Жасау (Создать) батырмасын басқан кезде, пішінді қалай жасау керектігін көрсететін терезе ашылады.

Ашылған Жаңа пішін (Новая форма) диалогтік терезесінде (42.1-сурет) Пішін шебері (Мастер форм) режімін көрсетіңдер және ашылған тізімнен пішін жасауға арналған кестені таңдаңдар.



42.1-сурет. Жаңа пішін диалогтік терезесі

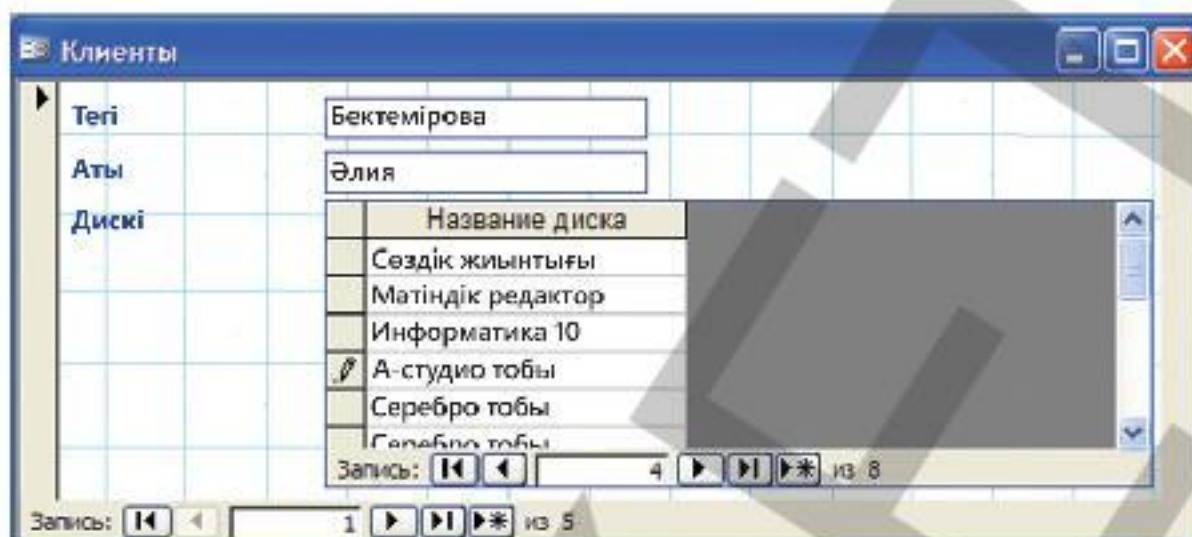
Пішіндерді жасау шеберінің бірінші диалогтік терезесінде пішінде болатын өрістерді көрсетіңдер. Қолжетімді өрістер тізімінде таңдалған өрісті пішінге жылжыту үшін «>» таңбасын, ал барлық өрісті жылжыту үшін «>>» таңбасы бар батырмасын басыңдар. Тасымалдауды аяқтаған соң *Келесі (Далее)* батырмасын басыңдар.

Келесі (Далее) шебер диалогі пішін түрін таңдау үшін қажет. Әдетте, қалпы бойынша енгізу өрістері бағанда орналастырылған пішін ұсынылады. Егер келіссендер, *Келесі (Далее)* батырмасын басыңдар (42.2-сурет).



42.2-сурет. Пішінге аудару мәліметінің мәзірі

Пішіннің атауын енгізу туралы сұрауда атауды енгізіп, *Аяқтау (Готово)* батырмасын басыңдар. 42.3-суретте көрсетілгендей келесі пішін пайда болады.



42.3-сурет. Пішін түрі

Пішінге түсетін деректердің көбісі кестеден немесе сұраудан алынады.

«*⏪*» «*⏩*» батырмасын бассаңдар, алдыңғы немесе келесі жазбаға оралуға болады.

Сүзгілерді пайдалану. Үлкен деректер қорының кестелерімен жұмыс істегенде барлық жазбалар компьютер экранында көрсетілмейді, тек кейбір критерийлер бойынша таңдалған жазбалар тобы ғана көрінеді. Кестенің жазбаларын Access бағдарламасында белгілі бір критерийлер бойынша анықтау үшін сүзгілер қолданылады. Сүзгілерді «Алтын балық аулау» жеке деректер базасына енгізуді қарастырыңдар. Бұл дерекқордың кестесі 42.4-суретте көрсетілген.

Аты	Мекен ететін жері	Ұзындығы (см)
Масажегіш балық	Оңтүстік Америка	5
Қара плятипецилиус	Мексика	4
Жасыл семсерқұйрық	Мексика	11
Хаплохилус	Африка	6
Тетра-фон-рио	Оңтүстік Америка	2
Бархат гирардинусы	Оңтүстік Америка	5
Макропод	Қытай	9
Теңбілкөк балық	Үндістан	10
Лялиус	Үндістан	5
Неон	Оңтүстік Америка	4


42.4-сурет. «Алтын балық» деректер қоры

Access бағдарламасы пайдаланушыға сүзгілеудің әртүрлі әдістерін ұсынады: ерекшеленген деректер және берілген шарт бойынша, ерекшеленген деректерді алып тастау керек.

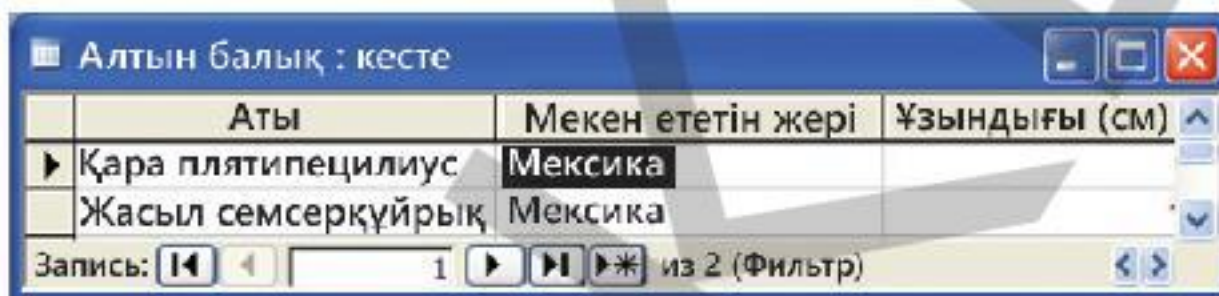
Таңдалған деректер бойынша сүзгілеуді орындау үшін 1-мысалды қарастырыңдар.

1-мысал. Балықтар туралы ақпаратты қамтитын кесте жазбаларын таңдаңдар, олардың мекен ететін жері — Мексика. Ол үшін келесі әрекеттерді орындаңдар:

1. «Мексика» мәнімен шыққан «Мекен ететін» бағанындағы кез келген ұяшықты басыңдар.

2. *Ерекшеленген сүзгілеу (Фильтр по выделенному)*  немесе келесі пәрмендерді орындаңдар: *Жазбалар* → *Сүзгі* → *Таңдау* (*Записи* → *Фильтр* → *Фильтр по выделенному*) бойынша сүзу.


Нәтижесінде, 42.5-суретте көрсетілгендей, мекен ететін жері бағанындағы «Мексика» мәніне ие барлық жазбалар таңдалады.



Аты	Мекен ететін жері	Ұзындығы (см)
Қара плектроцентрус	Мексика	
Жасыл семсерқұйрық	Мексика	

42.5-сурет. Сүзгі нәтижесі

Кестені сақтаған кезде соңғы орнатылған сүзгі Access бағдарламасымен еске алынып, оны қайтадан қолдануға болады.

Орнатылған сүзгіні өшіру үшін *Стандартты тақтасындағы* → *Сүзгіні жою*  (*Удалить фильтр*) батырмасын пайдалануға болады. Осы батырманы қайта басу соңғы сүзгіні кестеге қолданады.

Алынған деректерді алып тастау арқылы сүзгілеу 2-мысалда көрсетілген.

2-мысал. Балық туралы ақпараты жоқ кесте жазбаларын таңдаңдар, олардың ішінде Мексикадан басқа жерлерде мекен ететін балықтарды табу керек.

Ол үшін келесі әрекеттерді орындаңдар: *Жазба* → *Сүзгі және фильтр* → *Таңдау* → «Мексикада» мекен етпейді (*Главная* → *Сортировка и фильтр* → *Выделенное* → *Не содержит «Мексика»*).

Нәтижесінде «Мекен ететін жері» өрісінде «Мексика» мәніне ие емес жазбалардың барлығы көрсетіледі (42.6-сурет).

Аты	Мекен ететін жері	Ұзындығы (см)
Масажегіш балық	Оңтүстік Америка	
Хаплохилусы	Африка	
Тетра-фон-рио	Оңтүстік Америка	
Бархат гирардинусы	Оңтүстік Америка	
Макропод	Қытай	
Теңбілкөк балық	Үндістан	
Лялиус	Үндістан	
Неон	Оңтүстік Америка	

Запись: 1 из 8 (Фильтр)

42.6-сурет. Сүзгі нәтижесі

Көрсетілген шарт бойынша сүзу *Стандартты* тақтасында *Сүзгіні өзгерту* (Изменить фильтр) батырмасын немесе келесі әрекеттерді орындау арқылы орнату ыңғайлы: *Жазбалар* → *Сүзгі* → *Сүзгіні өңдеу* (Главная → *Сортировка и фильтр* → *Изменить фильтр*). Нәтижесінде *Сүзгілеу* (Фильтр) терезесі ашылып, онда шарттар көрсетіледі.

3-мысал. Оңтүстік Америкада пайда болған және ұзындығы 5 см болатын балық туралы ақпараты бар кестенің тек жазбаларын таңдаңдар.

Сүзгі терезесіндегі ашылмалы мәзірлерді пайдаланып, «Ұзындығы (см) = 5 және мекен ететін жері бағанында Оңтүстік Америкада» шартын орнатыңдар (42.7-сурет).

Аты	Мекен ететін жері	Ұзындығы (см)
	Оңтүстік Америка	5

Найти / Или

42.7-сурет. Белгіленген күйге арналған диалогтік терезені сүзу

Сүзгілеуді қолданғаннан кейін 42.8-суретте көрсетілген екі жазбаның кестесін аламыз.

Аты	Мекен ететін жері	Ұзындығы (см)
Масажегіш балық	Оңтүстік Америка	5
Бархат гирардинусы	Оңтүстік Америка	5
*		0

Запись: 1 из 2 (Фильтр)

42.8-сурет. Сүзгі нәтижесі

Осылайша пішін тек деректерге қол жеткізуге ғана емес, сонымен қатар оларды пайдаланушы үшін ыңғайлы түрде рәсімдеуге мүмкіндік береді.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Пішіндер деген не және ол не үшін қолданылады?
2. Пішіндер үшін деректер көзі болып не табылады?
3. Пішіндердің түрін қалай өзгертуге болады?
4. Пішіндердің деректер көзін қалай өзгертуге болады?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Өткен сабақта жасалған деректер қорына деректерді қарау мен кестеге енгізу үшін пішін құрындар.

В ДЕҢГЕЙІ

Пішінді жасап, оны деректермен жұмыс істеу оңай және түсінікті болатындай етіп рәсімдендер.

С ДЕҢГЕЙІ

«Пішінге растрлық және векторлық суреттерді қалай қосуға болады?» зерттеу жұмысын жүргізіндер. Графикада қолданылатын форма құруға тырысындар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

§ 44–45 Есеп

Бүгінгі сабақта:

- ▶ алынған деректерді пайдаланып есеп құруды үйренесіңдер.

Тірек ұғымдар:

- ▶ деректер қоры;
- ▶ кесте құрылымы;
- ▶ деректер типі;
- ▶ есеп құру.

Базада жиналған деректерді әдеттегі түрде ұсыну үшін есепті қалыптастыру қажет.



Есеп — ол деректерді басуға арналған құжат.

Бұл параграфта *Есеп шеберін (Мастер отчетов) Конструктор және Шебер диаграммасын (Конструктор и Мастер диаграмм)* қарастырамыз.

Есепті құрудың ең қарапайым әдісі — ол шебер мастерін қолдану. Деректер қорының есебін құрастырып, принтерде басып шығарайық. Ол үшін:

1. Деректер қорын ашыңдар. *Есеп (Отчет)* таңдау керек.
2. Создать батырмасын шертіп, ашылған *Жаңа есеп (Новый отчет)* терезесінде *Есеп шеберін (Мастер отчетов)* таңдаңдар.
3. Ашылған тізімде есеп жасау қажет өздеріңнің кестелеріңді көрсетіп, *OK* батырмасын шертіндер.
4. Келесі *Есеп құру (Создание отчетов)* терезесінде *Бос өріс (Доступные поля)* тізімінен барлық өрістерді *Таңдалған өріс (Выбранные поля)* тізіміне көшіріңдер. Содан кейін *Әрі қарай (Далее)* басыңдар.
5. Келесі терезе есеп өрістері үшін топтастыру деңгейін анықтау үшін арналған. Үнсіз келісім бойынша қалдыруға болады. *Әрі қарай (Далее)* басыңдар.
6. Шебердің нұсқауын орындаңдар, яғни келесі терезелерде өрістерді сұрыптау ретін, макет түрін, есепті безендіру стилін, сонымен қатар кесте атын таңдаңдар. *Дайын (Готово)* батырмасын басыңдар.

Нәтижесінде қарап шығу терезесінде шығарылатын есеп аламыз. Бұл терезеде жазбалардың орналасуын және олардың пішімін реттеуге болады. Ол үшін *Түр (Вид)* батырмасы арқылы *Есеп конструкторы* режиміне ауысу керек. *Есеп конструкторының* жеке терезесінде орындалатын бұл операциялар қарапайым және интуитивті түсінікті. Сондықтан оларды өз бетінше құруды ұсынамыз.

Жасалынған есепті басып шығаруға болады, ол үшін саймандар панелінде *Баспа (Печать)* батырмасын басу қажет.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Есеп не үшін қажет?
2. Есепті қалыптастыру кезіндегі әрекеттерді сипаттаңдар.

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

MS Access бағдарламасында есепті қалыптастыру алгоритмін құрастырыңдар.

В ДЕҢГЕЙІ

Деректер қоры бойынша есеп жасап, оны басып шығарыңдар.

С ДЕҢГЕЙІ

Есеп шебері (Мастер отчетов) режимінде екі кестеден тұратын есеп құрыңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 46–47 Сұраныстар

Бүгінгі сабақта:

- ▶ конструктор көмегімен таңдауға сұраныс құруды;
- ▶ қарапайым және қиылысатын сұраныс құруды;
- ▶ іріктеу шарттарын қоюды;
- ▶ есептеу сұранысын құруды үйренесіңдер.

Тірек ұғымдар:

- ▶ деректер қоры;
- ▶ кесте құрылымы;
- ▶ форма;
- ▶ сұраныс;
- ▶ қарапайым сұраныс;
- ▶ қиылысатын сұраныс;
- ▶ есептеу сұранысын құру.

Деректер қорында ақпарат іздеу сұраныстармен орындалады.



Сұраныс — деректер қорының бағдарламалық жасақтамасына жүгінуге, жазбаларды іріктеу немесе басқа операцияларды орындау үшін қажет.

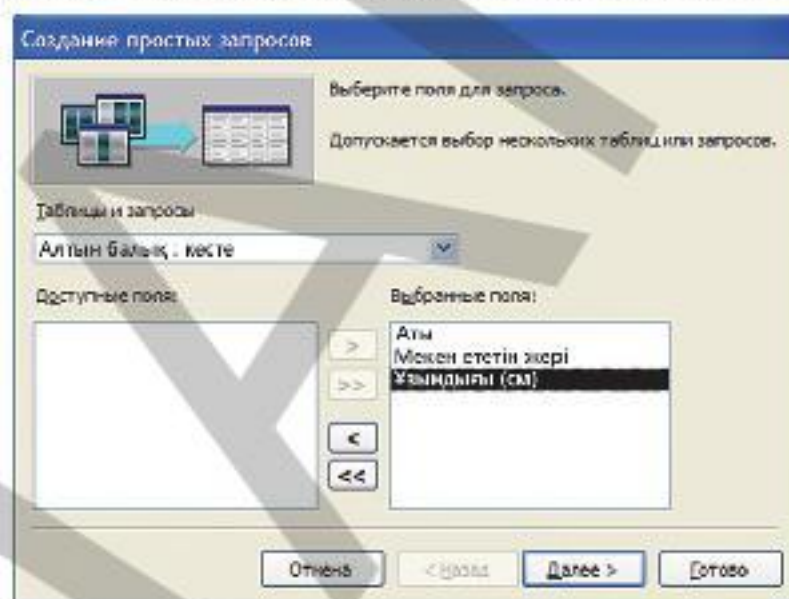
Сұраныс арқылы Access бағдарламасы берілген шарттарға жауап беретін деректер қорының кестелерінен жазбалар жиынтығын таңдайды және көрсетеді. Сұраныс бұрын жасалған бір немесе бірнеше байланысқан кестелер немесе сұраныстар негізінде құрылуы мүмкін.

Деректерді басқару үшін көптеген заманауи ДҚБЖ-да (яғни, сұраныстарды жасау үшін) SQL тілі қолданылады (ағылшынша Structured Query Language — құрылымдық сұрау тілі).

Access бағдарламасы өртүрлі сұрауларды қолдайды: *деректерді алу сұрауы, деректерді қосу және жою туралы сұрау, деректерді жаңарту туралы сұрау.*

Болашақта біз тек сұраныс деректерінің өртүрлі түрлерін қарастырамыз.

Берілген кесте бойынша кестелік деректер сұрауы. Access бағдарламасы *Шебер* мен *Конструкторды* (*Мастер* и *Конструктор*) пайдаланып, таңдау сұрауларын жасауға қолдау көрсетеді. Шебер қарапайым сұраныстарды жасау мүмкіндігін береді. 46.1-суретте көрсетілген «Алтын балық» кестесіне негізделген сұраныстарды жасауды қарастырайық.



46.1-сурет. Қарапайым сұрауды жасау диалогтік терезесі

1-мысал. Шебердің көмегімен балықтың көлемі туралы қарапайым сұраныс жасаңдар. Сұраныс жасау үшін келесі әрекеттерді орындау қажет:

1. Деректер қоры терезесіндегі *Сұраныстар* (*Запрос*) батырмасын басыңдар.

2. *Шеберді* пайдаланып, *сұраныстарды жасау* (*Создание запроса с помощью мастера*) жолын не екі рет басыңдар немесе келесі амалды

орындаңдар: *Құру* → *Жай сұраныстар* → *ОК* (*Создать* → *Простой запрос* → *ОК*).

3. *Сұраныстар жасау терезесін* (*Создание простых запросов*) ашқаннан кейін сұранысты жасау үшін кестені көрсетіндер, мысалы, *Алтын балық* және *Қолжетімді өрістер тізімінен қажетті атын, орнын және ұзындығы өрістерін Таңдалған өрістер* (*Выбранные поля*) тізіміне 46.1-суретте көрсетілгендей апару қажет.

Дайын (*Готово*) батырмасын басқаннан кейін үш таңдалған бағанды қамтитын қалыптасқан сұраныс ашылады (46.2-сурет).

Аты	Мекен ететін жері	Ұзындығы (см)
Масажегіш балық	Оңтүстік Америка	5
Қара платипецилиус	Мексика	4
Жасыл семсерқұйрық	Мексика	11
Хапклохилус	Африка	6
Тетра-фон-рио	Оңтүстік Америка	2
Бархат гирардинусы	Оңтүстік Америка	5
Макропод	Қытай	9
Теңбілкөк балық	Үндістан	10
Лалиус	Үндістан	5
Неон	Оңтүстік Америка	4

46.2-сурет. Деректерді таңдау үшін сұрау нәтижесі

Осындай *Сұраныс* (*Запрос*) «Алтын балық» кестесі арқылы *Конструктор* режимінде жасалуы мүмкін. Бұл жағдайда сұранысты жаңадан жасау керек. Ол үшін келесі әрекеттерді орындаңдар:

1. Деректер қойындысының терезесіндегі *Сұраныс* (*Запрос*) батырмасын басыңдар.

2. Дизайнерді пайдалану арқылы *Сұраныс* (*Запрос*) жасау батырмасын екі рет басыңдар немесе *Құру* (*Создать*) → *Конструктор* → *ОК* тізбегін орындаңдар.

3. *Кесте қосу* (*Добавление таблицы*) терезесінде «Алтын балық» деп аталатын кестені таңдап, *Қосу* (*Добавить*) батырмасын басыңдар.

4. Орындалған әрекеттердің нәтижесінде *Конструкторды таңдау туралы сұраныс* (*Запрос на выборку Конструктора*) терезесі ашылады (46.3-сурет).

Таблиця	Аты	Мекен ететін жері	Ұзындығы (см)
Алтын балық	Алтын балық	Алтын балық	Алтын балық

46.3-сурет. Конструктор көрінісіндегі күрделі сұрау

Осы терезеде солдан оңға жылжыту өрісі жолында тінтуірді басындар және ашылмалы тізімнен сұрауға қосқың келетін өрістерді таңдаңдар: *Балық аты, мекен ететін жері, ұзындығы.*

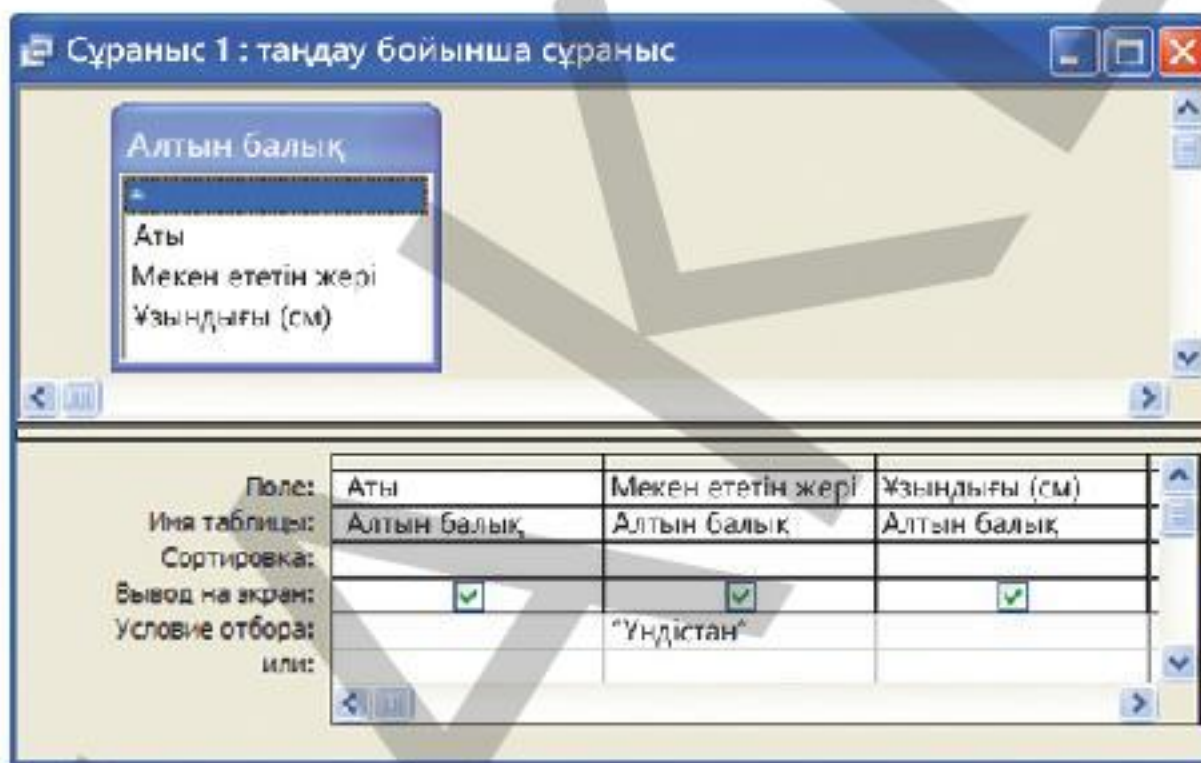
Экранға шығару (Вывод на экран) жолының әрбір таңдалған бағанында жалаушаны орнату, таңдау қажет бағандарды көрсетуге мүмкіндік береді. Сұраныс аяқталғаннан кейін оны сақтау керек.

Сұранысты орындау нәтижесінде жаңа уақытша кестені аласыңдар.

Конструктор терезесінде күрделі сұраулар жасаудың бірнеше мысалын қарастырыңдар.

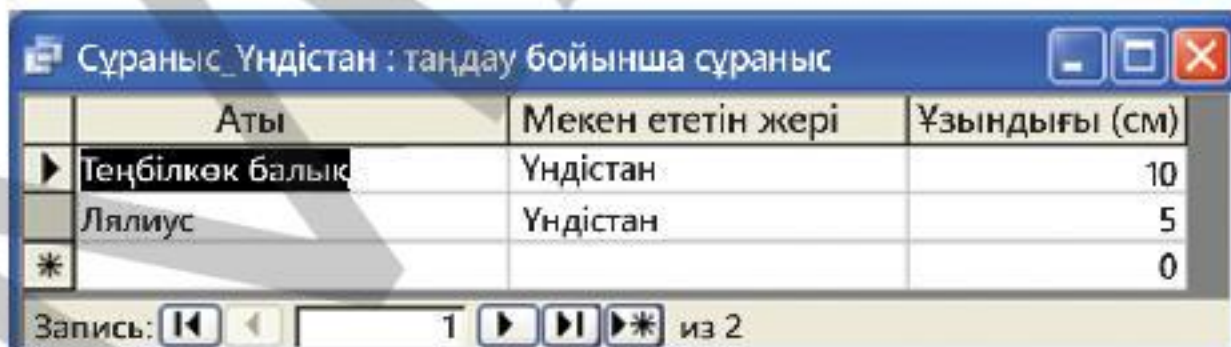
2-мысал. Балық туралы ақпаратты көрсететін сұраныс жасаңдар, «Мекен ететін жері» — Үндістан.

Таңдау критерийлерінің жолындағы *Сұранысты іріктеу шарты (Условие отбора)* терезесінде «Мекен ететін жері» өрісіне «Үндістан» мәнін енгізу керек, 46.4-сурет.



46.4-сурет. Конструктор режимінде сұрау жасау

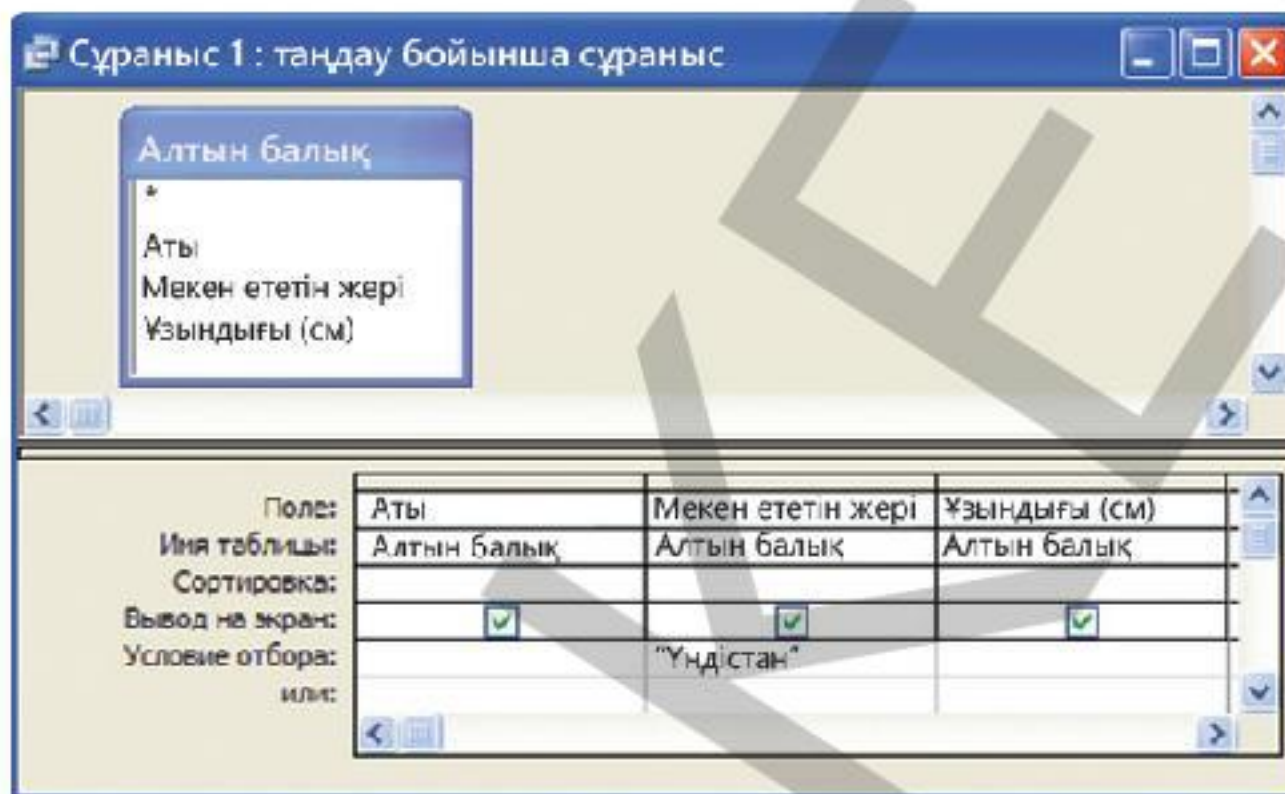
Қорыта келе, кестеден келесі нәтижені аласыңдар (46.5-сурет).



46.5-сурет. Сұрау нәтижесі

3-мысал. Шыққан жері Үндістан және ұзындығы 5 см-ден асатын балық туралы ақпаратты көрсететін сұраныстар жасаңдар.

Сұраныс конструкторы (Конструктор запроса) терезесіндегі Іріктеу шарттары (Условие отбора) жолында, 2-мысалдағы сұраныстар қосымша, Ұзындығы (см) жолында «> 5» мәнін 46.6-суретте көрсетілгендей енгізіңдер. Нәтижесінде 46.6-суретте көрсетілген кестені аласыңдар.



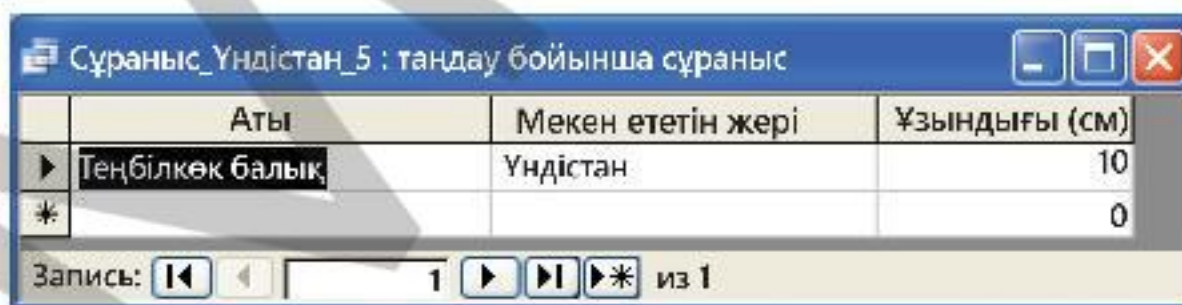
46.6-сурет. Конструктор режиміндегі сұраныс

Құрылымдық сұрау (SQL). SQL = *Structured Query Language* — басқаруға арналған құрылымдық сұраныстар тілі.

SELECT көмегімен бір немесе бірнеше кестелерден белгілі бір деректерге қызмет ету туралы кез келген сұрау. SELECT нәтижесі басқа кесте болады.

Осы кесте үшін «Алтын балық» 46.7-суретте берілген.

SELECT «Балық аты», «Мекен ететін жері», «Ұзындығы» FROM «Алтын балықтар».



46.7-сурет. Сұраныс нәтижесі

БАҚЫЛАУ СҰРАҚТАРЫ



1. Сұраныс не үшін пайдаланылады?
2. Access бағдарламасы қандай сұраныстар түрін қолданады?
3. Шебердің көмегімен қарапайым сұраныстарды қалай жасайды?
4. Конструктордың көмегімен қарапайым сұранысты қалай құрады?
5. Конструктор қолданушысы сұранысты құру үшін қандай шарттар ұсынады?
6. Фильтр не үшін қолданылады?
7. Access бағдарламасы қандай сұрыптаудың түрлерін ұсынады?
8. Белгіленген аймақ бойынша сүзгілеу дегеніміз не?
9. Белгіленген аймақ бойынша сүзгілеу қалай жүреді?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Бұрын жасалған деректер қорын ашып, таңдалған деректер бойынша кестенің қажетті жазбаларын таңдаңдар.

В ДЕҢГЕЙІ

Бұрын жасалған деректер қорын ашып, берілген шарт бойынша және таңдалған деректерді жою бойынша кестенің қажетті жазбаларын таңдаңдар.

С ДЕҢГЕЙІ

«Елдер» деректер қорын ашып, келесі сұраныстарды орындаңдар:
а) *мемлекет, астаналар, халқы, тілі* деген қарапайым сұранысты ашу;

- ә) < 10 млн адамнан аз халқы бар елдер туралы деректерді шығару;
- б) ағылшын тілінде сөйлейтін елдердің астаналарын көрсетіңдер.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 48–49 Құрылымдалған сұраныстар

Бүгінгі сабақта:

- ▶ құрылымдалған сұраныс тілін қолдануды үйренесіңдер.

Тірек ұғымдар:

- ▶ деректер қоры;
- ▶ кесте құрылымы;
- ▶ форма;
- ▶ сұраныс;
- ▶ қарапайым сұраныс;
- ▶ айқас сұраныс;
- ▶ есептеу сұранысын жасау;
- ▶ SQL тілі.

SQL тілі кестелермен (құру, жою, құрылымын өзгерту) және кесте мәліметтерімен (таңдау, өзгерту, қосу және жою) жүргізілетін операцияларды орындауға арналған.

SQL = Structured Query Language — деректерді басқару үшін құрылымдық сұраныс тілі.

Деректер базасымен жұмыс істеу кезінде кестеден қандай да бір ақпаратты қалай таңдау керек сұрағы жиі туындайды.

Бір немесе бірнеше кестеден белгілі бір деректерді алу үшін кез келген сұрау **SELECT** ұсынысымен орындалады. **SELECT** ұсынысының нәтижесі — басқа кесте.

Бір кестеден деректерді таңдау үшін қарапайым сұраныс келесі түрде жазылады:

`SELECT өрістер тізбегі FROM кесте аты`

1-мысал. Алдыңғы параграфтан «Алтын балық» кестесі үшін сұраныс:

`SELECT Балықтың аты, Мекен ететін жері, Ұзындығы, FROM Алтын балықтар.`

Бұл сұраныс «Алтын балықтар» кестесінен Балық атын, Мекен ететін жерін, Ұзындығын таңдайды.



SQL артық бос орындар мен мәтіндегі жаңа жолға өту пермендеріне қарамайды, сондықтан бұл сұранысты бірнеше жолға жазуға болады.

Сұраныс былай жазылады:

`SELECT Балықтың аты, Мекен ететін жері, Ұзындығы,
FROM Алтын балықтар`

2-мысал. Келесі мысалды қарастырайық.

`SELECT shortname FROM Алтын балықтар`

Бұл сұраныс алтын балықтар кестесінен тек аты қысқа балық атауларын таңдайды.

Ереже:

1. SELECT-тен кейін кесте өрістерінің бірін, бірнеше өрістерді немесе кестеде бар барлық өрістерді көрсетуге болады. Деректер қоры кестесінде бар өрістердің бір бөлігін ғана таңдайық. Бұл желі арқылы берілетін ақпараттың көлемін азайтады және қашықтықтағы деректер қорларымен жұмыс істеу кезінде үлкен жылдамдықты қамтамасыз етеді.

2. Өрістер кестеде жазылған ретпен емес, кез келген ретпен аталуы мүмкін.



3-мысалдағы келесі екі сұраныстың айырмашылықтары қандай?

3-мысал

SELECT Балық аты, Мекен ететін жері, Ұзындығы, FROM Алтын балықтар

SELECT Ұзындығы, Мекен ететін жері, Балық аты, FROM Алтын балықтар

Тек шығыс деректерінде өрістің басқа тәртібі болатынымен ерекшеленеді.

Өрістер тізімінің орнына * таңбасын көрсетуге болады, ол кестеде бар барлық өрістерді шығыс деректерінде көрсету керек екенін білдіреді. Бұл ретте өрістердің тәртібі кестедегі сияқты болады.

Бұл кестеде қандай өрістер бар екенін білмеген жағдайда қолдану ыңғайлы.

4-мысал

SELECT * FROM Алтын балықтар

Бұл сұраныс «Алтын балықтар» кестесінің барлық өрістерін шығарады. Іс жүзінде ол осы кестеде бар барлық ақпаратты шығарады.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Сұраныс дегеніміз не? Сұраныстар не үшін қолданылады?
2. ДҚБЖ-да сұраныстар қай тілде жасалады?
3. Бар сұранысты қалай өзгертуге болады?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Зерттеу жұмысын жүргізіндер. SQL сұраулары арқылы деректерді таңдаудан басқа қандай операцияларды орындауға болады?

В ДЕҢГЕЙІ

Атауында «а» әрпі бар барлық балық туралы деректерді іріктейтін сұраныс құрастырыңдар.

С ДЕҢГЕЙІ

Атауында үшінші әрпі «а» болатын барлық балық туралы деректерді алатын сұраныс құрастырыңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§50–52

Құрылымдалған сұраныстар. Жобалау жұмысы

Бүгінгі сабақта:

- ▶ құрылымдық сұраныс тілін қолдануды;
- ▶ конструктор арқылы іріктеуге сұраныс жасауды;
- ▶ қарапайым және айқас сұраныс жасауды;
- ▶ іріктеу шарттарын қоюды;
- ▶ есептеу сұранысын жасауды үйренесіндер.

Тірек ұғымдар:

- ▶ деректер қоры;
- ▶ кесте құрылымы;
- ▶ форма;
- ▶ сұраныс;
- ▶ қарапайым сұраныс;
- ▶ айқас сұраныс;
- ▶ есептеу сұранысын жасау.

1-тапсырма. Құрылыс ұйымы бірнеше жеткізушіден құрылыс бұйымдары мен материалдарды алады. Деректер қорында келесі мәліметтер көрсетілуі тиіс:

а) жеткізуші (жеткізуші туралы деректер: оның нөмірі, индексі, атауы, мекенжайы болып табылады, 50.1-кесте);

50.1-кесте

Жеткізуші №	Атауы	Мекенжайы	Қызметкерлер саны
2368	ООО «Гранит»	Абай даңғ., 30	30
3820	ЖП «Әубәкіров»	Төле би көш., 26	52
5024	ЖАҚ «Альфа»	Абай даңғ., 12	108
3015	АҚ «СТРОЙ»	Жұмабаев көш., 3	24
9543	ЖШС «Проект»	Ш. Уәлиханов көш., 9	185

ө) алынатын бұйымдар (бұйым туралы деректер: оның шифры, атауы, өлшем бірлігі, саны, жеткізуші болып табылады, 50.2-кесте).

50.2-кесте

Өнім				
Шифр	Атауы	Өлшем бірлігі	Саны	Жеткізуші
1238	Күлблок	дана	50	9543
1237	Цемент	т	18	3820
1247	Қабырға панелі	дана	60	5024
7421	Гипсокартон	дана	320	2368
1241	Есік блогы	дана	280	3820
5421	Терезе блогы	дана	82	1237
3248	Жабынды плита	дана	240	3820

1. Барлық жеткізушілердегі қызметкерлер санын 2-ге қысқартатын сұраныс жазыңдар.

2. ЖП «Әубәкіров» жеткізушісі өнімдерінің санын 5 данаға ұлғайтатын сұраныс жазыңдар.

3. Жеткізуші кестесінен атауын, мекенжайын және қызметкердің санын шығаратын сұраныс жазыңдар.

4. «Альфа» компаниясы жеткізетін барлық бұйымдардың тізімін шығаратын сұраныс жазыңдар.

5. Өнім кестесін кері тәртіппен шығаратын сұраныс жазыңдар.

6. Өнім кестесінен жеткізушілер нөмірлері тізімін алатын сұраныс жазыңдар. Нөмірлер қайталанбауы керек.

7. Қызметкерлердің ең аз саны жұмыс істейтін жеткізушілер атауы мен мекенжайын шығаратын сұраныс жазыңдар.

8. Қызметкерлер саны 100-ден кем емес барлық жеткізушілер үшін өнім атауы мен оның жеткізушісінен тұратын тізімді жасауға сұраныс жазыңдар.

9. № 2450 жеткізуші беретін барлық бұйымдарды жоюға сұраныс жазыңдар.

2-тапсырма. Дайындаушы зауыт бірнеше алушыларға құрылыс бұйымдары мен конструкцияларын жеткізеді. Келесі деректер туралы мәліметтер көрсетілуді тиіс:

а) алушылар (алушы туралы деректер оның коды, атауы, мекенжайы, зауыттан қашықтығы болып табылады, 50.3-кесте);

50.3-кесте

Жеткізуші №	Атауы	Мекенжайы	Қызметкерлер саны
5241	ОО О «Гранит»	Абай даңғ., 30	30
3820	ЖП «Әубәкіров»	Төле би көш., 26	52
2450	ЖАҚ «Альфа»	Абай даңғ., 12	108
3054	АҚ «СТРОЙ»	Жұмабаев көш., 3	24
1568	ЖШС «Проект»	Ш. Уәлиханов көш., 9	185

ә) жеткізулер (жеткізу туралы деректер оның шифры, бұйымның атауы, өлшем бірлігі, өлшем бірлігінің бағасы, алушы болып табылады, 50.4-кесте).

50.4-кесте

Шифр	Өнім	Өлшеу бірлігі	Бағасы	Алушы
1238	Шлакоблок	дана	5000	3054
1237	Цемент	т	900	3820
1247	Қабырға панелі	дана	6000	2450
7421	Гипсокартон	дана	1020	1568
1241	Есік блогі	дана	28000	1568
5421	Терезе блогі	дана	44000	3054
3248	Жабынды плита	дана	240	3820
5421	Шеге	кг	70	2450

1. Барлық жеткізу бағасын 1000 т-ге арттыратын сұраныс жазыңдар.
2. ЖАҚ «Альфа»-дан ЖШС «Проект»-ке жеткізуді беретін сұраныс жазыңдар.
3. Алушы кестесінен атауын, қашықтығын және мекенжайын шығаратын сұраныс жазыңдар.
4. ЖП «Әубәкіров» жеткізілімдері туралы барлық ақпаратты шығаратын сұраныс жазыңдар.
5. Жеткізулер кестесін бағандарын кері тәртіппен шығаратын сұраныс жазыңдар.
6. Алушылардың тізімін жеткізу кестесінен алатын сұраныс жазыңдар. Алушылар қайталанбауы тиіс.

7. Орташа жеткізу бағасын есептейтін сұраныс жазыңдар.

8. Зауыттан 70 км әрірек орналасқан барлық алушылар үшін өнімнің атауынан және алушының атауынан тұратын тізімді жасауға сұраныс жазыңдар.

9. Коды 2450 алушының барлық жеткізілімдерін жоюға сұраныс жазыңдар.

3-тапсырма. Құрылыс бөлімшесі бірнеше нысандарға жұмыс жүргізеді. Деректер базасында келесі мәліметтер болуы тиіс:

а) нысандар туралы (нысан туралы деректер: оның нөмірі, атауы, жұмыстардың сметалық құны (млн теңге), жұмыстардың орындалу пайызы болып табылады, 50.5-кесте);

50.5-кесте

Нысан

Нысан №	Атауы	Құны	Орындау
1	Емхана	150	70
2	Мектеп	82	60
3	Абай даңғылындағы тұрғын үй	140	90
4	Сауда орталығы	350	20
5	Балабақша	650	50

ә) ресурстарды жеткізу туралы (ресурстарды жеткізу туралы деректер: жеткізу коды, ресурстың атауы, өлшем бірлігі, саны, нысаны болып табылады, 50.6-кесте).

50.6-кесте

Ресурстарды жеткізу

Код	Ресурс	Өлшеу бірлігі	Саны	Нысан
3258	Цемент	кг	680	1
4342	Бояу	кг	350	4
5428	Шпатлевка	кг	260	3
5321	Кірпіш	м ³	36	2
4418	Құм	т	9	4
3021	Өк	т	4	5
5152	Бояу	л	120	2

1. Барлық нысандар бойынша орындауды 1% -ға арттыратын сұраныс жазыңдар.

2. «Абай даңғылындағы тұрғын үй» нысанына арналған ресурстарды «Мектеп» нысанына ауыстыратын сұраныс жазыңдар.

3. Нысан кестесінен атауын, құнын, орындауын шығаратын сұраныс жазыңдар.

4. Емхана нысаны үшін барлық ресурстарды жеткізу тізімін шығаратын сұраныс жазыңдар.

5. Ресурстарды жеткізу кестесі бағандарын кері тәртіппен шығаратын сұраныс жазыңдар.

6. Ресурстарды жеткізу кестесінен нысандар тізімін алатын сұраныс жазыңдар. Нысандар қайталанбауы тиіс.

7. Ең жоғары сметалық құны бар нысан үшін аталған орынның атауын және орындалу пайызын көрсететін сұраныс жазыңдар.

8. Ресурстың атауынан және нысанның атауынан тұратын тізімді жасауға сұраныс жазыңдар, ол барлық нысандар үшін қолданылады, ол бойынша орындалуы 60%-дан аспайды.

9. №1-нысан үшін арналған ресурстарды жеткізу кестесінен барлық жазбаларды жоюға сұраныс жазыңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

Өзіңді тексер!

1. Егер бір мұғалім әртүрлі оқушыларды оқытса, «мұғалімдер» мен «оқушылар» кестелерінің арасындағы байланыс түрін анықтаңдар:

а) «бірдің-бірге»;	б) «көптің-бірге»;
ә) «бірдің-көпке»;	в) «көптің-көпке».
2. ДҚ деректерді сақтау үшін қандай нысан қолданылады?

а) есеп;	б) сұраныс;
ә) форма;	в) кесте.
3. Деректерді басқару мен мониторингке арналған бағдарлама немесе бағдарлама кешені:

а) деректер қорын басқару ортасы;	
ә) деректер қорын көбейту жүйесі;	
б) деректер банкі.	
4. Егер әрбір қалаға бірнеше аудан сәйкес келсе, «қала» мен «аудан» кестелерінің арасындағы байланысты анықтаңдар

а) «бірдің-бірге»;	б) «көптің-бірге»;
ә) «бірдің-көпке»;	в) «көптің-көпке».

5. Реляциялық деректер қоры өрісінің типі анықталады:
- а) өрістің аты;
 - б) ұяшық атауы;
 - ә) деректер типі;
 - в) кілттің типімен.
6. Иерархиялық деректер қоры — бұл?
- а) ақпарат тікбұрышты кесте түрінде ұйымдастырылған ДҚ;
 - ә) жазудағы элементтері реттелген ДҚ, яғни бір элемент басты, қалғандары бағынышты болып саналады;
 - б) жазулары еркін түрде орналасқан ДҚ;
 - в) вертикальды иерархиялық байланыстарға қосымша орнату мүмкіндігі бар ДҚ.
7. Пішін (Форма) пайдаланылады:
- а) деректерді енгізу үшін;
 - ә) деректерді таңдау үшін;
 - б) деректерді баспаға шығару үшін;
 - в) деректерді қарау үшін.
8. Кестедегі қайталанатын жазбалардың болуын қалай жоюға болады?
- а) кесте жолдарын реттеу арқылы;
 - ә) кілттік өрісті анықтау арқылы;
 - б) сыртқы кілтті анықтау арқылы;
 - в) кесте өрісін индекстеу арқылы.
9. MS Access бір типті деректер бағаны аталады:
- а) жазба;
 - б) есеп;
 - ә) сұраныс;
 - в) өріс.
10. Жазбаларды таңдау және деректер қорының бір немесе бірнеше кестелерінен деректерді жаңарту үшін қолданады:
- а) есеп;
 - б) сұраныс;
 - ә) пішін;
 - в) кесте.
11. Деректер қоры — бұл:
- а) қолданушы бағдарламаларының пакеті;
 - ә) қатты дискідегі файлдар жиынтығы;
 - б) нақты әлемнің нысандарын, процестерін немесе құбылыстарын сипаттайтын мәліметтер жиынтығы;
 - в) деректер қорын басқару жүйесі.
12. ДҚБЖ — бұл:
- а) деректер банкіні басқару құралдарының жүйесі;
 - ә) деректер банкіні мұрағаттау және резервтік көшіру құралдары жүйесі;
 - б) деректерді басқару және оларға қол жеткізуді қамтамасыз ету үшін арнайы бағдарламалық кешен;
 - в) транзакцияларды басқару құралдарының жүйесі.

13. MS Access кестесі өрістерінің деректер типі (артығын алып тастаңдар):
- а) логикалық;
 - б) жалпы;
 - ө) санауыш;
 - в) мәтін.
14. Реляциялық деректер қорының құрылымы өзгереді:
- а) өрістің бірін жойған кезде;
 - ө) бір жазбаны жойған кезде;
 - б) бірнеше жазбаны жойған кезде;
 - в) барлық жазбаларды жойған кезде.
15. Келесі мәнді жазу үшін кесте өрісіне қандай деректер типін таңдау керек (72) 23-89-49:
- а) логикалық;
 - б) сандық;
 - ө) санауыш;
 - в) мәтін.
16. Мәтін типтегі өрістер үшін үнсіз келісім бойынша қандай өлшем анықталады?
- а) 255 таңба;
 - б) 65536 таңба;
 - ө) 100 таңба;
 - в) 50 таңба.
17. Өртүрлі жазбаларда қайталанбайтын өріс аталады:
- а) өріс типі;
 - б) сыртқы кілт;
 - ө) алғашқы кілт;
 - в) негізгі кілт.

V

ТАРАУ

WEB-БАҒДАРЛАМА

Бұл тараудан білесіңдер:

- ▶ белгілеу тілін тағайындауды;
- ▶ тегтер тұжырымдамасы, элементтер, атрибуттарын;
- ▶ HTML құжаттың құрылымын;
- ▶ Мәтінді пішімдеудің қарапайым әдістерін;
- ▶ CSS;
- ▶ деректер базасымен web-беттердің байланысын.

Мынаны үйренесіңдер:

- ▶ Web-беттерді әзірлеу кезінде HTML-тегтерді пайдалануды;
- ▶ Web-беттерді әзірлеу кезінде CSS-ні пайдалануды;
- ▶ Web-беттерді әзірлеу кезінде скриптерді пайдалануды;
- ▶ Web-бетте мультимедиа нысандарын енгізу үшін HTML тегтерін қолдануды;
- ▶ Web-беттерді әзірлеу кезінде дайын скриптерді пайдалануды;
- ▶ Web-беттердің деректер базасымен байланысын орнатуды.

§ 53–54

**HTML (HyperText Markup Language)
web-сайттарын әзірлеу әдістері**

Бүгінгі сабақта:

- ▶ Web-беттерді жасауда HTML-тегтерін қолдануды үйренесіңдер.

Тірек ұғымдар:

- ▶ тег;
- ▶ атрибуттар;
- ▶ пікірлер.

World Wide Web әлемдік топ негізі болып HTML (HyperText Markup Language) гипермәтіндік белгілеу тілі есептеледі.

БҰЛ ҚЫЗЫҚ!

WWW негізіндегі гипермәтіндік тұжырымдаманы 1960 жылдары Теодор Холман Нельсон ұсынды. XX ғасырда Гипермедианың технологиясы гипермәтіннің әрі қарай дамуы болып табылады, ол гиперсілтемелерді тек фрагменттерді ғана емес, әртүрлі типтегі деректерді (графика, дыбыс жазу, сандық бейне және т.б.) байланыстыруға мүмкіндік береді.

Интернетке гипермәтінді қолдану идеясын алғаш рет 1989 жылы Ядролық зерттеулердің еуропалық орталығындағы (CERN, Женева) элементар бөлшектер физикасы зертханасының қызметкері Тим Бернерс-Ли жүргізді.



Белгілеу тілінің арқасында пайдаланушы өз экранында Web-құжатты әзірлеушінің ойластырған түрінде көруі мүмкін: белгілі бір қаріп мөлшерімен және абзацтарға бөлумен, көрсетілген өлшемдермен және суреттердің орналасуымен және т.б.



HTML — браузер бағдарламасы арқылы оларға кіру кезінде компьютер экранында құжаттардың пайда болуын анықтайтын құжаттарды таңбалау туралы келісімдер жиынтығы.

HTML түзету тілімен жасалған құжат мәтіндік файл болып табылады. Мұндай файлды әдеттегі мәтін редакторында (Блокнот немесе WordPad) теруге және редакциялауға болады. Сонымен бірге HTML-құжаттарды (визуалды HTML редакторлары және HTML-мәтін редакторлары) дайындау үшін ыңғайлы және тамаша бағдарламалар бар.

**HTML көмегімен құжатты қалай белгілейді?**

HTML тілінде көптеген тегтер бар, олардың ішінде: құжат тақырыбын жасау, қаріп параметрлерін орнату, графиканы енгізу тегтері және т.б.



Тег (ағылшынша tag — нұсқаушы, таңба) — бұл HTML құжаттың нақты элементін сипаттайтын код үзіндісі және ол `< >` жақшаға алынады.

Интернет парақшасы бірнеше бөліктен тұрады: тақырыбы `<HEAD>` және денесі `<BODY>`. Бұл құрылым қарапайым түрде келесі 53.1-кестеде берілген:

53.1-кесте

<code><HTML></code>	парақша нысанының басы HTML
<code><HEAD></code>	нысан басының тақырыбы
<code><TITLE></code>	нысан жолының басы — парақшаның атауы
...	парақшаның атауы жазылған жол
<code></TITLE></code>	нысан жолының соңы — парақшаның атауы
<code></HEAD></code>	нысан тақырыбының соңы
<code><BODY></code>	нысан денесінің басы
...	парақшаның денесі (барлық мәліметтер)
<code></BODY></code>	нысан жолының соңы
<code></HTML></code>	HTML документі нысанының соңы

Мысал қарастырайық.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Сәлем! Hello! </TITLE>
```

```
HEAD>
```

```
<BODY> Бұл менің алғашқы WEB-бетім! Бастысы ол жұмыс істейді!!!
```

```
</BODY>
```

```
</HTML>
```

Егер қажет болса, HTML-мәтініне өзгерістерді, әртүрлі таңбамен пікірлерді жазуға болады. `<! -- және -->` Оны «жартылай тегі» деп жазады.

Көптеген тегтер атрибуттардан деп аталатын қосымша элементтерді қамтиды. Мысалы, <BODY> құжатының дене тегіне қосымша элемент енгізіңдер:

<BODY bgcolor= «yellow»> — бұл документ сары фонда жазылуы керек деген сөз. bgcolor сөзі атрибут, ал yellow — атрибут мәні.

БАҚЫЛАУ СҰРАҚТАРЫ



1. HTML белгілеу тілінің мақсаты қандай?
2. HTML құжаты дегеніміз не?
3. HTML тегтері және элементтері дегеніміз не?
4. Документтің тақырыбы және денесі деген не?
5. Атрибуттардың мақсаты қандай және қай бөліктерде орналасқан?
6. Пікірді қалай жазуға болады?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Блокнот мәтіндік редакторын ашыңдар және оқулықтың параграфында көрсетілген мәтінді теріп, Index.html файлын сақтаңдар. Содан кейін құрылған және сақталған файлды іске қосыңдар. Нәтижесін талдап қорытынды жасаңдар.

В ДЕҢГЕЙІ

Интернет-ресурсты қолдана отырып, 53.2-кестені толтырыңдар.

53.2-кесте

Практикалық жұмысқа мысал

Тегтер	Сипаттамасы
1	2
<HTML>	Құжаттың ашылуы
<HEAD>	
<BODY>	
 </BR>	
<HR></HR>	
<CENTER></CENTER>	
<ADDRESS>	
	
<I></I>	

1	2
<SUB>	
<SUP>	
	
	
	
 Элемент 1 Элемент 2 Элемент 3 	
 Элемент 1 Элемент 2 Элемент 3 	
	
<BODY BACKGROUND= «URL»>	
<BODY BGCOLOR= «x»>	
<BODY TEXT= «x»>	
<BODY VLINK= «x»>	
<BODY LINK= «x»>	
<BODY ALINK= «x»>	

С ДЕҢГЕЙІ

Бос web-бет үшін HTML-кодтың қандай нұсқасы дұрыс?

а)	ә)	б)
<pre><html> <head> <title> </head> <body> </body> </html></pre>	<pre><html> <head> <title> </title > </head> <body> </body> </html></pre>	<pre><html> <head> <title> </title > <body> </body> </html></pre>

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 55 Мәтінді форматтау (қаріп, абзац, тізімдер)

Бүгінгі сабақта:

- ▶ CSS web-бетін дайындауда қолдануды үйренесіңдер.

Тірек ұғымдар:

- ▶ HTML құжатта жолдарды, абзацтарды, тақырыптарды форматтау.

Абзацтар мен жолдарды қалыптастыру. Бетке жазғыларын келетін мәтін алдын ала дайындалып, содан кейін оған қажетті тегтер қосылады. HTML тілінде абзац жасау үшін бірнеше мүмкіндіктер бар. Олардың ең қарапайымы — бұл `<P>` және `</P>` тегтерін пайдалану, олардың арасында абзацтың мәтінін орналасады.

Жолды келесі жолға көшіру үшін `
` тегі қолданылады. Бұл тег браузерді мәтінді жаңа бетке көшіреді. Әсіресе өлеңдер тергенде ыңғайлы.

Тақырыпты жазу. Таңдалған абзацты тақырыппен немесе кіші тақырыппен жариялау үшін (ол үшін қалған мәтінге қарағанда үлкен немесе қалың қаріпті таңдап) осы абзацты `<H*> ... </H*>` аралығына жазу жеткілікті.

1 саны — тақырып үшін ең үлкен қаріпті көрсетеді;

2 саны — сәл аз (кіші тақырыптар үшін);

3 саны — одан да аз және т.б.

Абзацтарды туралау. `<P>` және `
` тегімен берілетін абзацтар үнсіз келісім бойынша беттің сол жақ шетінде тураланады. `Align=«center»` ортасы бойынша туралау, `align=" right "` — беттің оң жақ шеті бойынша туралау, тиісінше `aling=«left»` — сол жақ шеті бойынша туралау орындалады.

Қаріп параметрлерін өзгерту: түрі (гарнитура), таңбалар өлшемі, түсі. Web-беттегі мәтінді көрсету үшін қолданылатын қаріп параметрлері ` ... ` тегі көмегімен жасалады. Бұл элемент үшін келесі атрибуттар бар:

— `face` — қаріп гарнитурасы немесе рұқсат етілген қаріптер тізімі;

— `түс` — қаріп түсі;

— `size` — қаріп өлшемі.

Мысалы, Arial қаріпін және қызғылт-сары түсті мәтінді қосыңдар. Ол үшін құжатқа келесі тег енгізу қажет:

```
<FONT face= «Arial» color = «orange» </FONT>.
```

Қаріптің қандай да бір пішінін көрсету үшін келесі тегтерді пайдалану қажет: ` ... ` — жартылай қалың мәтін үшін; `<I> ... </I>` — курсив мәтін үшін және `<U> ... </U>` — асты сызылған мәтін үшін.

Қаріп түсі мен беттің фонын жасау. Егер барлық бет үшін қаріп түсін анықтау керек болса, онда `text` атрибутын `<BODY>` тегінде қолданады. Мысалы,

`<BODY text = «red» >` — барлық мәтін үшін қызыл түсті көрсетеді.

Барлық HTML құжат фоны түсі `bgcolor` тегінің `<BODY>` атрибутымен анықталады. Мысалы, келесі тег фон үшін зәйтүн түсін тағайындайды:

`<BODY bgcolor = «olive»>`

55.1-кесте

Мәтінді рәсімдеу үшін тегтер

<code><hr></code>	Экрандағы мәтін фрагменттерін көзбен көру үшін дөңес көлденең сызық жасайды. Бұл тег жабатын тегті талап етпейді
<code><p> мәтін </p></code>	Жеке абзац қалыптастырады: — абзац алдында шағын шегініс қосылады; — абзац сол жақ жиегіне тураланады; — сөздер арасында бір бос орын (олардың бастапқы мәтінде қанша жеткізілгеніне қарамастан); — егер кезекті сөз экрандық жолға сыймаса, мәтін автоматты түрде тасымалданады
<code><p align= center> </p></code> <code><p align= left> </p></code> <code><p align= right> </p></code> <code><p align= justify> </p></code>	Align абзац туралауын көрсетеді: — <code>center</code> — абзацтың барлық жолдарын орталықтандыру, соның ішінде <code>
</code> тегінің көмегімен жолдың мәжбүрлі үзілулері болған кезде; — <code>left</code> — сол жақ жиегі бойынша туралау; — <code>right</code> — оң жақ шеті бойынша туралау; — <code>justify</code> — енін туралау
<code><h1> тақырып мәтіні </h1></code> <code><h2> тақырып мәтіні </h2></code> <code><h3> тақырып мәтіні </h3></code> <code><h4> тақырып мәтіні </h4></code> <code><h5> тақырып мәтіні </h5></code> <code><h6> тақырып мәтіні </h6></code> <code><h1 align= justify </h1></code>	Тақырыптар ақпаратты жеке логикалық бөлімдерге бөледі және осылайша қабылдауды едәуір жақсартады. HTML тақырыптары мәні бойынша (деңгейі бойынша) өртүрлі болады. Тақырып <code><h1 > ... < / h1></code> — ең үлкен, <code><h6 > ... < / h6></code> — ең ұсақ
<code> мәтін </code>	Мәтінді жартылай қалыңдатып жазу
<code><i> мәтін </i></code>	Мәтінді курсивті жазу
<code> </i> мәтін </i></code>	Мәтіннің жартылай қалың және курсивті жазу комбинациясы

БАҚЫЛАУ СҰРАҚТАРЫ



1. HTML-құжаттағы жеке жолды (абзац) қалыптастыратын элементтерді атаңдар.
2. Иерархиялық тақырыптарды қандай элементтер арқылы жасайды?
3. Абзацтарды туралау үшін қандай атрибут қолданылады?
4. Қаріп параметрлері қандай тегтер мен атрибуттар арқылы беріледі?
5. Қандай тег курсивті және асты сызылған қаріпті жасайды?
6. Беттегі қаріп түсін қалай орнатуға болады?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Өздеріңнің Web-беттерінде өртүрлі безендіру элементтерін қолданыңдар.

В ДЕҢГЕЙІ

Маркерленген және нөмірленген тізімді өздерің қарап, web-беттерінде қолданыңдар.

С ДЕҢГЕЙІ

Беттердің фондық және қаріптік безендірілуін дайындаңдар. Кесте қосып, оны толтырыңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 56–57 Кестелер

Бүгінгі сабақта:

- ▶ web-беттерді жасау кезінде HTML тегтерін қолдануды;
- ▶ Web-құжаттарда кестелер жасауды;
- ▶ кестелерді форматтауды;
- ▶ кестелерде ұяшықтарды біріктіруді үйренесіндер.

Тірек ұғымдар:

- ▶ кесте элементтері;
- ▶ кестелерді форматтау;
- ▶ кестелерде ұяшықтарды біріктіру.

Web-құжаттардағы кестелер кестелік деректерді орналастыру үшін ғана қолданылмайды. Олар суреттер мен сілтемелерді кірістіру үшін, Web-беттерді тиімді орналастыру үшін қызмет етеді.

Кесте элементтері. Кестелер салу принципі бойынша құрылады және бірқатар элементтердің көмегімен Web-бетке енгізіледі. Әрбір кесте `<TABLE>` тегімен басталып, `</TABLE>` тегімен аяқталады. Құрылатын кесте жолдар бойынша ашылады, ал жолдар ұяшықтармен толтырылады. Сонымен қатар тег ішіне `<TABLE>...</TABLE>` келесі элементтер енгізілуі мүмкін:

TR — жолды құру элементі;

TD — ұяшықтың мазмұнын анықтайтын элемент;

TH — тақырып ұяшығын анықтайтын элемент.

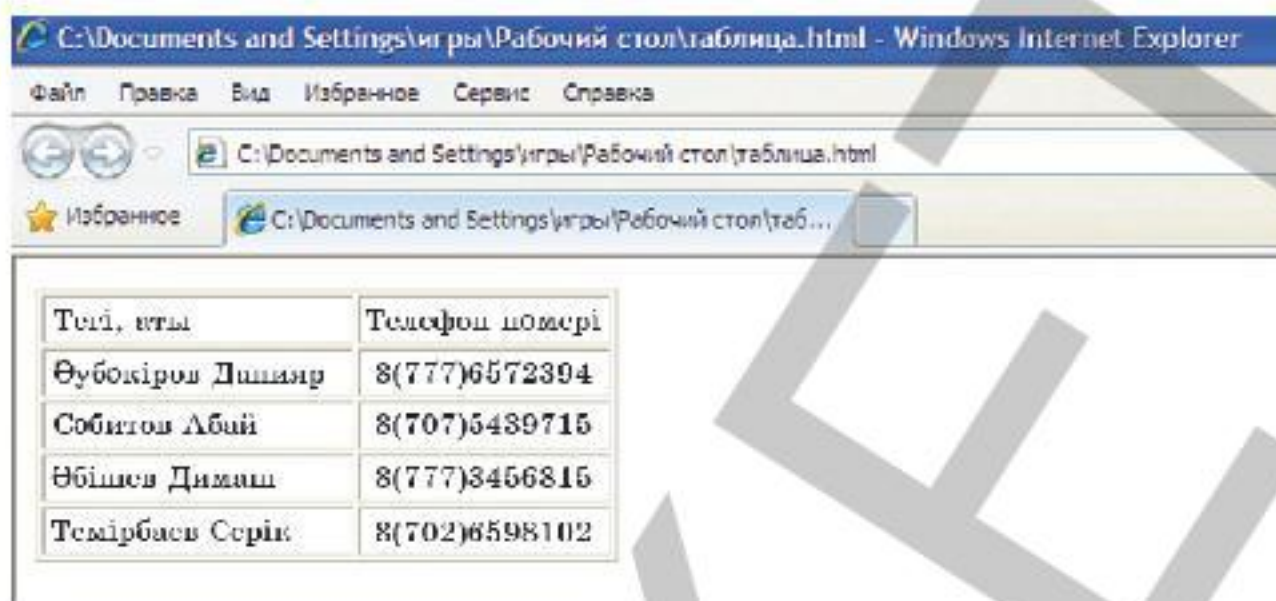
Мысалы, 3×2 кестесін жасау үшін келесі үлгі қолданылады:

```
<TABLE>
<TR><TD> . . . </TD><TD> . . . </TD></TR>
<TR><TD> . . . </TD><TD> . . . </TD></TR>
<TR><TD> . . . </TD><TD> . . . </TD></TR>
</TABLE>
```

Осы үлгі бойынша достарыңның телефондарының тізімі сияқты кестесін құрыңдар:

```
<TABLE border>
<TR><TD> Тегі, аты</TD><TD> Телефон </TD></TR>
<TR><TD> Әубәкіров Данияр </TD><TD> 8(777)6572394 </TD></TR>
<TR><TD> Сәбитов Абай </TD><TD> 8(707)5439715 </TD></TR>
<TR><TD> Әбішев Димаш </TD><TD> 8(777)3456815 </TD></TR>
<TR><TD> Темірбаев Серик </TD><TD> 8(702)6598102 </TD></TR>
</TABLE>
```

Ескерту. <TABLE> тегіне қалыңдығы 1 пиксел сыртқы және ішкі жақтауларын көрсететін border атрибуты енгізілді. Бұл кесте Web-бетте суретке сәйкес болады (56.1-сурет).



56.1-сурет. Сыртқы және ішкі жиектері бар кесте

Кесте параметрлерін орнату. Біздің мысалда кесте ұяшықтардағы мәтіннің ең үлкен ұзындығына тең бағандардың еніне ие болады. Кесте браузердің сол жақ шетінде тураланған, ал оның мазмұны Times New Roman қарпімен көрсетіледі. Тақырып ұяшықтарындағы мазмұн ортасы бойынша, ал басқа ұяшықтарда сол жақ бойынша тураланады. Үнсіз келісім бойынша орнатылған параметрлерді өзгерту үшін 56.1-кестесінде ұсынылған әртүрлі атрибуттар қолданылады.

56.1-кесте

TABLE элемент атрибуттары

Атрибут атауы	Сипаттама
width — кестенің енін көрсетеді	Оның мәні пиксельдермен немесе пайызбен (браузердің толық енінен) көрсетіледі. Мысалы, тег <TABLE= «40%» > терезенің енінің 40%-ына тең барлық жолдардың ұзындығы бар кестені анықтайды. Кестедегі жолдар браузер терезесінде толық көрсетілгендіктен енін пайызда көрсеткен дұрыс
Align — кестені құжатта туралауды анықтайды	Бұл атрибут үш мәнді бірін қабылдай алады: left — кестені құжаттың сол жақ жиегіне орналастыру; center — құжаттың ортасында; right — оң жақ жиегі бойымен
border — кесте жиектерін көрсетеді	Егер бұл атрибуттың мәні анықталмаса, мысалы, <table border>, барлық жақтаулар 1 пиксель қалыңдығына ие болады. Егер, мысалы, border = 5 болса, пиксельдің қалыңдығы тек сыртқы жақтауға беріледі. Ішкі жақтаудың қалыңдығы бұрынғыша 1 пиксельге тең болады.

Кестедегі түс. Кестенің түсін басқаратын атрибуттарды сипаттайық (56.3-кесте).

56.3-кесте

Атрибут атауы	Сипаттама
<code>bgcolor</code> — кестедегі фон түсін анықтайды	Бұл атрибуттың қандай тегке енгізілуіне байланысты (<code><TABLE></code> , <code><TR></code> , <code><TH></code> немесе <code><TD></code>) барлық кестенің фоны, жолдың фоны, тақырып ұяшығының фоны немесе деректер фоны қойылады. Мысалы, тег <code><TABLE bgcolor = «red»></code> барлық кестенің қызыл фонын тағайындайды, ал тег <code><TD bgcolor = «yellow»></code> деректер ұяшығының сары фонын орнатады
<code>bordercolor</code> — кесте жиектерінің түсін белгілеу	Егер <code>bordercolor</code> атрибутын <code><TABLE></code> тегіне қойсақ, онда ол кестеде жиектер болған кезде, яғни <code>bordercolor</code> атрибуты болған кезде өрекет етеді. Егер белгілі бір ұяшықтардың түсін орнату қажет болса, <code>bordercolor</code> атрибуты <code><TR></code> , <code><TH></code> немесе <code><TD></code> тегтеріне орналастырылады. Мысалы, тег <code><TR bordercolor = «FF0000»></code> барлық жол ұяшықтарының қызыл жиектерін көрсетеді

Кесте ұяшықтарын біріктіру. HTML тіліне шектес ұяшықтарды біріктіру мүмкіндігін қарастырайық. Бұл үшін бастапқы `<TH>` немесе `<TD>` тегтерде кестеде сипатталған келесі атрибуттар қолданылады (56.4-кесте).

56.4-кесте

Ұяшықтарды біріктіру атрибуттары

Атрибут атауы	Сипаттама
<code>rowspan</code> — шектес жолдар ұяшықтарын біріктіреді	Атрибуттың мәні біріктірілетін ұяшықтар санын анықтайды. Мысалы, ұяшықтың бастапқы тегі <code><TD rowspan=2></code> шектес жолдардан екі ұяшықты біріктіреді
<code>colspan</code> — шектес бағандар ұяшықтарын біріктіреді	Мысалы, <code><td colspan=3></code> шектес бағандардың үш ұяшығынан бір деректер ұяшығын қалыптастырады

Бір уақытта екі атрибутты — `rowspan` және `colspan` қолданса, шектес жолдар мен бағандардан біріктірілген ұяшықты аламыз. Мысалы, `<TD rowspan=2 colspan=4>` тегі екі жол мен төрт бағанның қиылысында орналасқан ұяшықты көрсетеді.

БАҚЫЛАУ СҰРАҚТАРЫ



1. Кестені жасайтын HTML-элементтерді атаңдар.
2. Бағдарламада орнатылған шартқа сәйкес кестенің параметрлері қандай?
3. Web-құжатқа 3×3 кестесін қалай қою керек? HTML кодын жазыңдар.
4. Кесте енін қалай көрсету керек?
5. Кестені құжаттың ортасына, оң жақ шетіне қалай туралау керек?
6. Кесте жақтауларының қалыңдығы мен түсі қандай атрибуттармен беріледі?
7. Жол ұяшықтарын белгілі түспен қалай құю керек?
8. Ұяшықтардың мазмұнын оң және жоғары шеттері бойынша туралау үшін атрибуттардың мәнін жазыңдар.
9. Кесте ұяшықтарын біріктіру қандай атрибуттардың көмегімен орындалады?

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Web-беттеріне 3×2 кестесін қосып, оны толтырыңдар.

В ДЕҢГЕЙІ

Web-беттегі кестені түспен безендіріңдер.

С ДЕҢГЕЙІ

Кестедегі бірнеше ұяшықтарды біріктіріңдер.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 58–60 CSS (Cascading Style Sheets)

Бүгінгі сабақта:

- web-беттерді жасауда CSS қолдануды үйренесіңдер.

Тірек ұғымдар:

- CSS технологиясы.

HTML элементтерін рәсімдеудің әдістерін жақсарту үшін CSS HTML енгізілді.

Каскадты стиль кестелері (Cascading Style Sheets) — беттің мазмұнын өңдеу және рәсімдеу процесін жеңілдетуге мүмкіндік беретін Web-беттің дизайнын басқарудың технологиясы.



CSS технологиясы не үшін қажет?

50 беттен тұратын сайтты елестетіп көріңдер, мәтіннің түсі, фоны, әр құжаттың тақырыптарын өзгерту және т.б. өзгертуді шештіңдер делік. CSS-ті қолдану арқылы бұл бірнеше секундтан кейін орындалуы мүмкін. CSS файлының тек бір жолын өңдей отырып, тораптың барлық тақырыптары мен кестелерінің көрінісін өзгертуге болады. Бұл әзірлеушіге күнделікті жұмыста емес, дизайнға көңіл аударуға мүмкіндік береді.

Ол үшін сыртқы CSS файлында тиісті атрибуттардың мәндерін өзгерту керек.

Барлық h2 тақырыптарын сұр түсте жасайық.

HTML-де:

```
<h2> <font color = «сұр»> тақырып </ font> </ h2>
```

h2-нің көптеген элементтері бар болса, онда бұл — ыңғайсыз процесс. Егер қажет болса түсті жасылға өзгерту керек пе?

CSS-те бір жазба жасаймыз:

```
h2 {түсі: сұр;}
```

— барлық тақырыптар сұр түсті, ал қажет болса оларды жасылға ауыстырасыңдар, сонда бір ғана сөзді өзгертесіңдер.

Ереженің құрылымы 58.1-суретте көрсетілген.



58.1-сурет. Бетті редакциялау фрагменті

Селектор — жиі (бірақ әрқашан емес) HTML элементі — `h1`, `p`, `em`, `table` болып табылады

```
P {color: silver;}      em {background: red;}
```

Жарнамалық бірлікте бір немесе бірнеше жарнама бар. Кейін «:» және «;» кеңістіктердің ерікті саны болуы мүмкін.

Мәндер — бос орынмен бөлінген бір кілт сөз немесе бірнеше жарамды кілт сөздер:

```
P {font: medium Helvetica;}
```

Жарамсыз сипатты немесе мәнді көрсетесіңдер, бүкіл жарнама толығымен еленбейді.

Мысалы, келесі типтегі тізім бар:

```
<ul>
<li> Әйгерім </ li>
<li> Ержан </ li>
<li> Димаш </ li>
<li> Серік </ li>
</ ul>
```

Осы тізімнің нәтижесі төменде келтірілген:

- Әйгерім
- Ержан
- Димаш
- Серік

Каскадты кестелерді қолданамыз:

```
<title>Заманауи сабақтар CSS</title>
<style type=»text/css»>
ul li {list-style:none; display:inline; background-
color:#99ccff; padding:2px}
</style>
</head>
<body>
<ul>
<li>Әйгерім </li>
<li>Ержан</li>
<li>Димаш</li>
<li>Серік</li>
</ul>
</body>
```

Нәтижесінде төмендегілерді аламыз:

Әйгерім	Ержан	Димаш	Серік
---------	-------	-------	-------

58.1-кесте

Қаріп және абзац қасиеттері

font-family	Мәтін көрсетілетін қаріпті немесе қаріптік топты көрсетеді (өйтпесе қаріптерді пайдаланушы браузері таңдайды). Пайдаланушыда қажетті қаріп болмауы мүмкін болғандықтан, қаріптердің бірнеше түрін олардың қалауы бойынша көрсеткен дұрыс. Мысалы, <code>p {font-family: Verdana, sans-serif}</code>
font-weight	Қаріп қалыңдығын анықтайды: <code>lighter</code> — қарапайым, <code>bold</code> — жартылай қалың, <code>bolder</code> — қалың. Қаріп қалыңдығы 100-ден 900-ге дейінгі сандармен анықталуы мүмкін. Мысалы, <code>p {font-family: bolder}</code>
font-size	Қаріп өлшемін анықтайды. Мән пайыздарда, пиксельдерде және тармақтарда, сондай-ақ <code>smaller</code> — өте кішкентай, <code>small</code> — шағын, <code>medium</code> — орташа, <code>large</code> — үлкен, <code>larger</code> — өте үлкен мәндерде көрсетілуі мүмкін. Мысалы: <code>h1 {font-size: 200%}</code> <code>h2 {font-size: 200px}</code> <code>h3 {font-size: 400pt}</code>
font-variant	Мәтін түрін қарапайым немесе капитель — бас әріптермен жазуды анықтайды. Рұқсат етілген мәндер: <code>normal</code> немесе <code>small-caps</code> . Егер мән көрсетілмесе, әдетте, <code>normal</code> мәні қолданылады
font-style	Қаріптің курсивті жазылуын анықтайды: <code>normal</code> — өзгеріссіз, <code>italic</code> — курсив
text-decoration	Астын сызу немесе сызу сияқты қаріпті ресімдеу әсерлерін анықтайды. Рұқсат етілген мәндер: <code>none</code> — жоқ, <code>overline</code> — үстін сызу, <code>underline</code> — астын сызу, <code>line-through</code> — сызып тастау. Мысалы, <code>h1 {text-decoration: underline}</code>
text-align	Абзацты туралауын анықтайды Мысалы, <code>p {font-align: left}</code> <code>p {font-align: right}</code> <code>p {font-align: justify}</code> <code>p {font-align: center}</code>
text-indent	Бірінші («қызыл») жол үшін шегіністі белгілейді Мысалы, <code>p {font-indent: 60pt}</code>
text-transform	Мәтінді түрлендірудің өртүрлі режимдерін анықтайды. Рұқсат етілген мәндер: <code>capitalize</code> — әр сөздің бірінші әрпін үлкен әріппен жазады, <code>uppercase</code> — барлық әріптерді бас әріппен, <code>lowercase</code> — барлық әріптерді кіші әріппен жазады, <code>none</code> — барлық қондырғыларды жояды

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Стильдердің каскадтық кестелері web-құжаттың мазмұны мен безендірілуін кез келген мәтіндік редактордағы стильдер сияқты бөлуге мүмкіндік береді. Олардың көмегімен көрсетілетін құжаттың сыртқы түрі анықталады: мәтін түсі, қарпін (гарнитура), ережелерді туралау және т.б. Кез келген HTML тегімен берілетін әрбір элемент үшін браузер терезесінде тиісті форматтау нұсқауларын өзгерте отырып, көрсету стилін анықтауға болады. Құжаттың стильдік белгісі туралы ақпарат HTML құжатының `<head>` бөлімінде орналасқан `<style>... </style>` контейнерінің ішінде болады.

Мысал:

```
<html>
  <head>
    <title> ... </title>
    <style type= «text/
css»>
  <! -
  h1 {
    font-family:Verdana
  }
</style>
</head>
<body>
.....
</body>
</html>
```

CSS жағдайында **Type** атрибуты қолданылатын стиль кестелерінің түрін көрсетеді. Мұнда **Verdana** тақырыбының қарпін өзгерту үшін **h1** тегіне қолданылатын бір стиль туралы айтылады.

Стильдің бірінші бөлігі *селектор* деп аталады. Ол осы стильді қолдану аймағын анықтайды (біздің жағдайда **h1** селекторын көрсетеді, онда бұл стиль web-құжат құрамында барлық **h1** контейнерлерінің ішіндегі мәтінге қолданылады).

Фигуралы жақшалар стиль денесін шектейді. Оның ішінде қасиеттері мен мәндері бар жол бар: сипат — қос нүкте символының алдында, ал тиісті мән қос нүктеден кейін болады.

В ДЕҢГЕЙІ

Үлгі бойынша ақпаратты форматтау үшін мәтінге **h1**, **h2**, **p** стильдерін қолданыңдар.

```

<style>
<!--
h1 { font-family: Courier New;
font-size: 20pt;
font-weight: bolder;
text-align: center;
}
h2 { font-family: Courier New;
font-size: 14pt;
font-weight: bolder;
text-align: justify;
}
p { font-family: Arial;
font-size: 18pt;
text-align: left;
}
-->
</style>

```

h1 стилі үшін Courier New қарпі, өлшемі — 20 пункт, қою қаріп, ортасында туралау

h2 стилі үшін sans-serif қарпі, өлшемі — 14 пункт, қою қаріп, ені бойынша туралау

P стилі үшін қаріп Arial, өлшемі — 18 пункт, сол жақ жиегі бойынша туралау

Үлгі

Растрлық және векторлық графика

Растрлық графика

Растрлық сурет жолдар мен бағандарды құрайтын әртүрлі түсті нүктелердің (пиксельдердің) матрицасы ретінде қалыптасады. Әрбір пиксель он мыңдаған немесе тіпті ондаған миллион түстері бар палитрадан кез келген түсті қабылдай алады. Сондықтан растрлық суреттер түстердің және жартылай тондардың жоғары дәлдігін қамтамасыз етеді.

Векторлық графика

Векторлық бейнелер графикалық примитив деп аталатын нысандардан (нүкте, сызық, шеңбер, тіктөртбұрыш және т.б.) қалыптасады. Әрбір примитив үшін тірек координаталары мен түсі беріледі.

□ ДЕҢГЕЙІ

HTML құжат тегіне CSS қосындар. Кестемен тәжірибе жасаңдар. Нәтижесін қорытындылаңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 61 Мультимедианы енгізу

Бүгінгі сабақта:

- Web-бетке мультимедиа нысандарын кірістіру үшін HTML тегтерін қолдануды үйренесіңдер.

Тірек ұғымдар:

- якорь;
- якорь элементі.

Интернет-парақтарды суреттері бар қарапайым мәтіннен қалай ажыратады? *Сілтеме (гиперсілтеме)* — бұл беттегі кез келген мәтін фрагментін (сөзді, фраза немесе бүкіл абзацты) немесе басқа интернет бетімен немесе деректер файлымен (мысалы, GIF немесе JPEG графикалық файл) байланыстыру мүмкіндігі. Сондықтан егер осындай мәтінді немесе суретті бассандар, сілтемеде көрсетілген бетке автоматты түрде кіресіңдер. Бет басқа серверде болған жағдайда байланыс бар болса, онымен байланыс автоматты түрде орнатылады. Беттегі сілтемелер болып табылатын мәтіннің фрагменттері түсі өзгеше (әдетте, көк) және асты сызылған болып көрінеді.

Сілтеме бастау нүктесі `<A>` тегімен анықталады. Бұл тегтің атауы сөздің бірінші әрпінен алынады. `A` элементі байланыстырушы элемент немесе **якорь элементі** деп аталады. `<A>` тегінің ішінде міндетті түрде `href` атрибуты орнатылған, ол сілтеме (мақсатты ресурс) тағайындалған пунктті анықтайды. `<A>` және `</ A>` тегтерінің арасында сілтеме мәтіні немесе сызба элементі орналастырылған. Осылайша қарапайым сілтеме келесідей болуы мүмкін:

```
<A href= «rest.html»> Менің демалысым </ a>
```

Web-бетте бұл сілтеме «Менің демалысым» мәтіні ретінде көрсетіледі. Осы сілтемені басқанда `rest.html` HTML файлы жүктеледі.



Href атрибуты түпнұсқа құжаттың сол папкада орналасқан файлға салыстырмалы сілтемеге сәйкес келетін файл атауын ғана көрсетеді.

WWW-те орналастырылған ресурсқа сілтеме жасағың келсе, `href` атрибуты `href` осы ресурстың URL-мекенжайы көрсетіледі, мысалы, ` Samsung өнімдері ` `href` атрибутының мәні үшін `mailto` ресурсын (SMTP электрондық пошта хаттамасының қоңырауы) көрсетуге болады. Мысалы, пішіннің анықтамасы:

```
<A href= «mailto://mektap@mail.ru»> Letter </A>
```

парақшаны келушісіне тікелей `mekter@mail.ru` мекенжайына хабарлама жіберу мүмкіндігін береді.

Кескінді кірістіру. Суреттері жоқ WWW парақшасын табу қиын. Кескінді веб-параққа қосу бір `` тегі арқылы орындалады. Бұл тегтің ішіне сурет URL мекенжайын қамтитын `src` атрибуты әрдайым жазылады. Бұл атрибуттың атауы `source` дереккөз сөзінен алынған.



Қалай ойлайсыңдар, бетте суретті қалай орналастыру керек?

Мұны істеу үшін суретті файлды белгілі бір бумада сақтаңдар (мысалы, HTML құжатының өзі сияқты папкаға) және құжатқа `<IMG src = «picture.jpg»` тегін енгізіңдер.

Әдетте, шолғыш беттің сол жағына сәйкес келетін суретті көрсетеді.



Веб-беттегі кескін қандай өлшемде болады?

Әдетте, браузер графикалық файлда сақталған кескіннің нақты өлшемдерін пайдаланады. Бұл өлшемдерді өзгерту үшін `` тегінің ені мен биіктігі атрибуттары қолданылады, сурет өлшемінің мәні, әдетте, пиксельдермен орнатылады, мысалы, `width = «133»` биіктігі `«33»`.

Сондай-ақ ені мен биіктігін сыртқы элемент өлшеміне қатысты пайызбен (бет) көрсетуге болады. Интернетте ең танымал екі графикалық пішім (61.1-кесте) берілген.

61.1-кесте

JPG — суреттер және реңктер арасында тегіс ауысулары бар түс гаммасы бойынша күрделі суреттер үшін; GIF — нақты түсті шегаралары бар логотиптер, жазулар, тақырыптар мен суреттер үшін	
<code></code>	Тег <code></code> мәтіннің қажетті жерінде суретті орналастыруға мүмкіндік береді
<code></code>	<code>Src</code> параметрі сурет файл атауын көрсетеді
<code></code>	<code>alt</code> параметрі суреттерді мәтіндік түсініктемен жабдықтауға мүмкіндік береді
<code></code>	<code>Width</code> параметрі сурет енін көрсетеді

<code></code>	Height параметрі сурет биіктігін көрсетеді
<code></code>	Border параметрі суреттің жиегінің қалыңдығын анықтайды
<code></code>	Hspace параметрі суреттің сол және оң жақтағы өрістерді анықтайды
<code></code>	Vspace параметрі суреттің жоғары және төменгі жиектерін көрсетеді
<code></code>	Align параметрі құжаттың көршілес элементтеріне (ең алдымен — қоршаған мәтінге) қатысты иллюстрацияның орнын анықтайды; top — жоғары жиегі бойынша тік туралау; middle — тік ортаға туралау; bottom — төменгі жиегі бойынша тік туралау; left — сол жақ жиегі бойынша көлденең туралау; right — оң жақ шеті бойынша көлденең туралау

БАҚЫЛАУ СҰРАҚТАРЫ



1. HTML қандай элементтерге сілтеме жасайды?
2. Web-бетте қандай элемент және атрибут бейнеленген?
3. Суреттің өлшемін қалай орнату керек?
4. Web-бетте орналастыратын суреттің графикалық форматы қандай болуы керек?

ПРАКТИКАЛЫҚ ЖҰМЫС

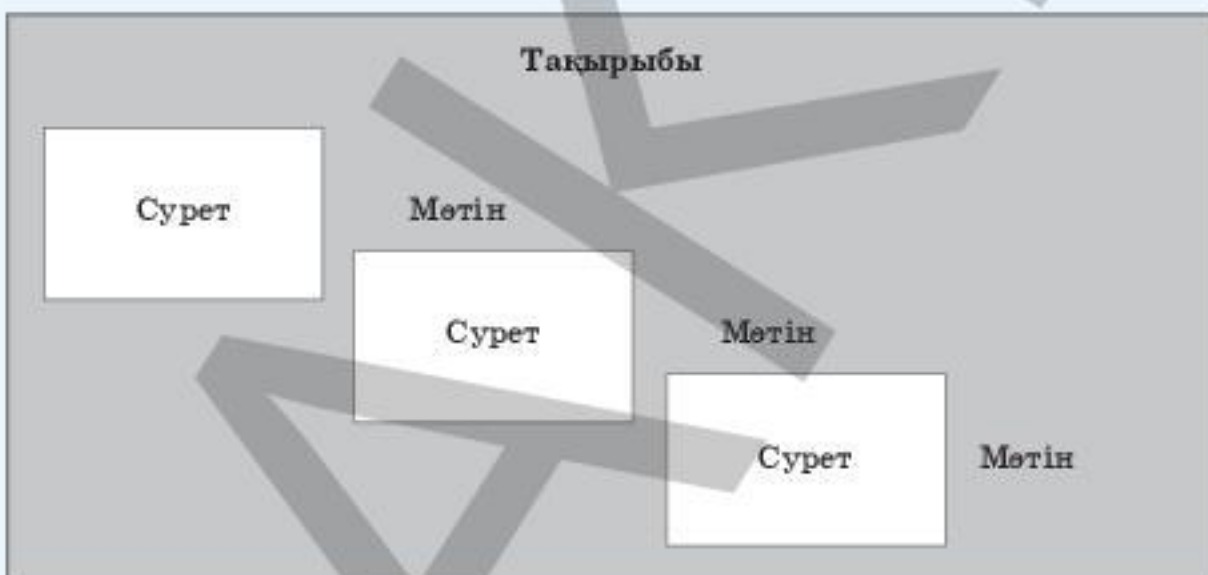
А ДЕҢГЕЙІ

Web-бетті келесі үлгілер бойынша жасаңдар:

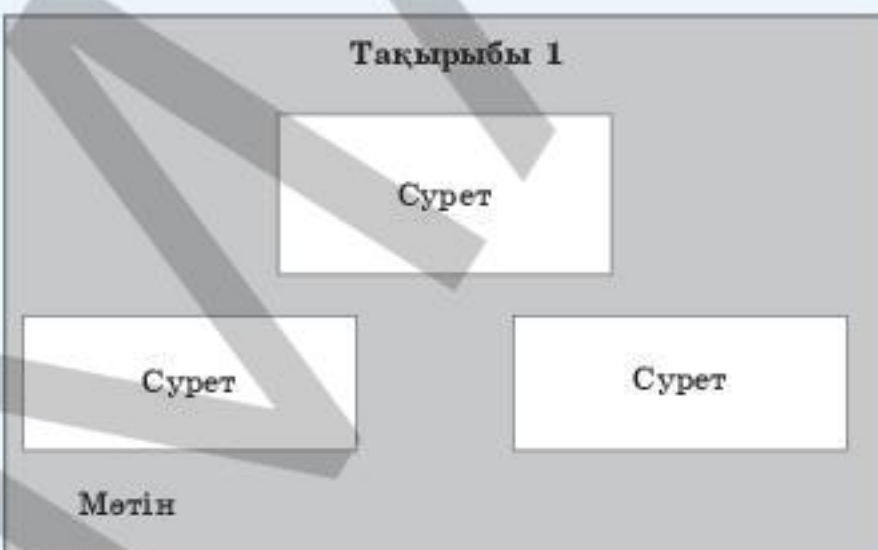
1.



2.



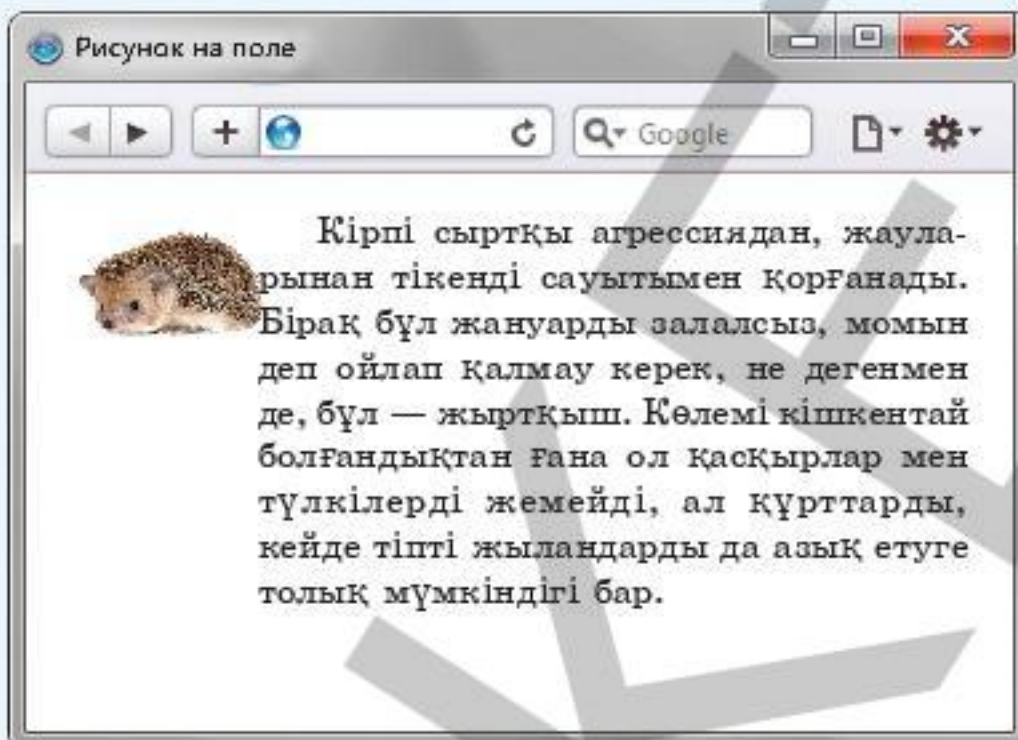
3.



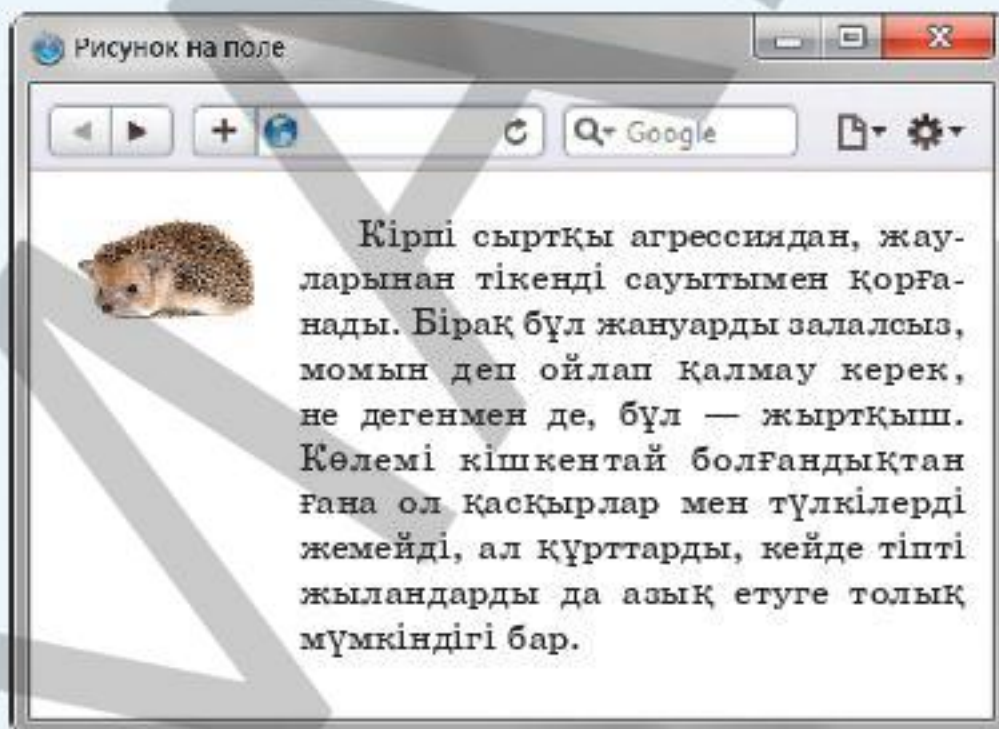
В ДЕҢГЕЙІ

Суреттер және мәтінді орналастыру.

61.1-суретте көрсетілген Web-бетті қараңдар. Осы суреттегі қатені түзетіп, Web-бетті 61.2-суреттегі сияқты түрлендіріңдер.



61.1-сурет. Қатесі бар web-бет



61.2-сурет. Web-бетті редакциялау

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласындар?

§ 62 Скриптерді қолдану

Бүгінгі сабақта:

- ▶ Web-беттерді жасауда скриптерді қолдануды үйренесіндер.

Тірек ұғымдар:

- ▶ сайт скрипті.

Скриптер статистиканы (келушілер туралы, пікірлер т.б.) жинау, сайтта іздеуді оңтайландыру, сайт құрылымын жеңілдету, форумдардың жұмысын ұйымдастыру үшін қолданылады.

Скриптер бөлек файлға орналастырылып, серверге жүктеледі. Оны орындау қажет болғанда файлға хабарлама келеді.



Сайт скрипті — бұл ресурста іске асырылатын және порталдың функционалдығын кеңейтетін код. Басқаша айтқанда, скрипт — белгілі бір тапсырманы орындау үшін веб-сервер бетінен келетін арнайы сұрау арқылы сервер іске қосылған кезде орындалатын процесс.

Қарапайым мысал қарастырайық. Автоматты кіржуғыш мәшине — бірнеше режімде кір жуу үшін оны дұрыс бағдарламалау керек. Батырманы басқан кезде белгілі бір жуу режімінің скрипті іске қосылады.

Сайтта скрипт осыған ұқсас жұмыс істейді. Белгілі бір шарт бойынша жұмыс істейді және өз жұмысын орындайды.

Скриптерді жасайтын бірнеше бағдарламалау тілдері бар.

Мысалы,

- Jscript;
- Python;
- JavaScript;
- PHP;
- Perl;
- AngelScript.

Тест құруға мысал (JavaScript бағдарламасының пішін элементтерімен өзара әрекеттесуі) қарастырайық.

JavaScript — интерактивті тесттермен жабдықталған HTML негізіндегі электрондық оқулықтарды жасайтын бағдарламалардың бірі болып табылады. Ол кейбір пішіні бар (сұрақтың мәтіні мен диалог элементтері бар) web-бет болуы мүмкін. Пішін жауапты таңдауға немесе енгізуге мүмкіндік береді.

Сынақ пішіндерін және тиісті Java-скрипттерін жасау нұсқаларының бірін қарастырыңдар. Ең қарапайым тәсілі — жауаптардың тізімін қамтитын пішінді жасау, оның біреуі дұрыс, мұндағы жауап мәтіндері батырмалар болып табылады. Бұл жағдайда жауаптарды талдау тек *Verno ()* немесе *Neverno ()* функциясының батырмаларының әрқайсысы үшін «Дұрыс жауап» және «Дұрыс емес жауап» хабарлары бар ескерту alert терезелерін шығаратын onclick оқиға шақыруына ғана азаяды.

Формалардың мысалдары мен есептері:

```
<H2 ALIGN=CENTER><FONT COLOR= red>
Тест 1<FONT></H2>
<P ALIGN=JUSTIFY><B> Сұрақ.</B> Келденең құрылу үшін
қандай тәг қызмет етеді? </P>
<FORM NAME= «TEST1»>
<INPUT TYPE= BUTTON VALUE= «1» NAME= «otv1»
onClick= «javascript: Neverno ();»>
&lt; TABLE&GT; <BR>
<INPUT TYPE= BUTTON VALUE= «2» NAME= «otv1»
onClick= «javascript: Verno ();»>
&lt; HR&gt; <BR>
<INPUT TYPE= BUTTON VALUE= «3» NAME= «otv3»
onClick= «javascript: Neverno ();»>
&lt; LINE&gt; <BR>
<INPUT TYPE= BUTTON VALUE= «4» NAME= «otv4»
onClick= «javascript: neverno ();»>
&lt; Pow&gt;
</FORM>
...
<SCRIPT LANGUAGE= javascript >
// дұрыс жауап - №2
}
Function Verno () {

Alert («дұрыс жауап»);
}
Function Neverno () {
}
Alert («жауап дұрыс емес»);
}
</ SCRIPT >
```

БАҚЫЛАУ СҰРАҚТАРЫ



1. Скрипт дегеніміз не?
2. Скрипті жазуға арналған бағдарламалық тілдерді атап шығындар.

ПРАКТИКАЛЫҚ ЖҰМЫС

А ДЕҢГЕЙІ

Жоғарыда оқулықтың параграфында берілген тапсырманы анықтаңдар.

В ДЕҢГЕЙІ

5 сұрақ пен жауапты қамтитын интерактивті тест жасаңдар.

С ДЕҢГЕЙІ

5 сұрақ пен өртүрлі жауаптардан (мәтінді енгізу өрісіне жауапты енгізу, жауаптың бірнеше нұсқалары бар сұрақ, дайын фрагменттерден жауап дайындау және т.б.) интерактивті тест жасаңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағың келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

§ 63

Web-беттің деректер қорымен байланысы

Бүгінгі сабақта:

- Web-беттің деректер қорымен байланысын орнатуды үйренесіңдер.

Тірек ұғымдар:

- Web-бет;
- деректер қоры.

Параграфта Web-беттерді деректер қорымен қалай байланыстыруды қарастырайық. Деректер базасына қосылу өте оңай:

```
db = openDatabase("ToDo", "0.1", "A list of to do items.",
200000);
```

Бұл код деректер қорын ұсынатын нысанды жасауға мүмкіндік береді, ал егер деректер қоры жоқ болса, ол құрылады. Аргументте деректер қорының аты, нұсқасы, көрсетілетін аты және шамамен өлшемі көрсетіледі. Шамамен берілген өлшемі шектеулі емес екенін айта кеткен дұрыс. Деректер қорының нақты өлшемі өзгеруі мүмкін.

Деректер қорына қосылымды келесі команда арқылы тексеруге болады:

```
if(!db){alert("Failed to connect to database.");}
```

Деректер қорына сұраныстарды орындау үшін `database.transaction()` функциясын шақыра отырып, алдын ала транзакцияны жасау қажет. Оның бір аргументі бар, яғни транзакция нысанын қабылдайтын және деректер қорына сұраныс қабылдайтын JavaScript функциясы.

SQL сұраныстар транзакция нысаны `executeSql` функциясын шақырып, орындауға болады.

`executeSql` функциясының мысалы:

```
db.transaction(function(tx) {
tx.executeSql("SELECT COUNT(*) FROM ToDo", [],
function(result){}, function(tx, error){});
});
```

Бұдан әрі кодты «ToDo» кестесінен (әлі бұл кесте жоқ) таңдау мүмкін болмаған жағдайда бұл кесте құрылатындай етіп өзгертесіңдер.

```
db.transaction(function(tx) {
tx.executeSql("SELECT COUNT(*) FROM ToDo", [], function
(result) { alert('dsfsdf') }, function (tx, error) {
tx.executeSql("CREATE TABLE ToDo (id REAL UNIQUE, label
TEXT, timestamp REAL)", [], null, null);
}});
```

Сондай-ақ, SQL сұраныстары арқылы кестеге жолдарды қосуға және жоюға болады.

ПРАКТИКАЛЫҚ ЖҰМЫС

A ДЕҢГЕЙІ

Web-бет пен деректер қоры арасында байланыс орнатыңдар.

B ДЕҢГЕЙІ

«ToDo» кестесіне жаңа жазбаларды қосыңдар.

C ДЕҢГЕЙІ

«ToDo» кестесіндегі бос өрістерді толтырыңдар.

Рефлексия:

- ▶ Қандай ақпарат қызықтырды?
- ▶ Қандай қиындық туындады және оны кіммен талқылағын келеді?
- ▶ Қазір қандай дағдыны қолдана аласыңдар?

Өзіңді тексер!

- Суретті жүктеу мүмкін болмаған жағдайда суреттің мәтінін қай атрибуттың көмегімен қоюға болады?
 - caption;
 - alt;
 - title;
 - popup.
- Кестедегі ұяшықтың ішіндегі кері шегінуді қандай қасиет арқылы жасауға болады?
 - case;
 - margin;
 - space;
 - padding.
- Мәтінді курсивпен қалай ерекшелеуге болады?
 - `<p>курсив</p>`;
 - `<hr>курсив</hr>`;
 - `курсив`;
 - `<c>курсив</c>`.
- Пішін өрістеріне жазуды қандай тег арқылы жазуға болады?
 - id;
 - label;
 - type;
 - field.
- Суретке қалай сілтеме жасайды?
 - ``;
 - ``;
 - ``.
- Бірінші деңгейдегі тақырыптардың саны қандай болады?
 - 3;
 - 4;
 - 1;
 - 2.
- Дәйексөзді (цитата) қалай дұрыс ресімдеу керек?
 - `<blockquote> дәйексөз мәтіні <cite> дәйексөз авторы </cite></blockquote>`;
 - `<blockquote> дәйексөз мәтіні </blockquote><cite> дәйексөз авторы </cite>`.
- Мәтінді қалай қою түспен жазуға болады?
 - `
 қою </br>`;
 - `<p> қою </p>`;
 - `<a> қою `;
 - ` қою `.
- HTML-ге суретті қалай қою керек?
 - `<image>http://site.com/image.jpg</image>`
 - `http://site.com/image.jpg`
 - `<image source="http://site.com/image.jpg">`
 - ``
- Қандай қасиет арқылы кестенің ені өзгереді?
 - count;
 - length;
 - size;
 - width.

Глоссарий

IP-адрес — Интернеттегі компьютердің орнын анықтайтын мекенжайы. IP-адрес 0 мен 255 аралығында бір-бірінен нүкте арқылы бөлінген төрт саннан тұратын жазбаны білдіреді, мысалы: 192.168.0.13.

HTML — компьютер экранында браузер бағдарламасы арқылы ашылатын құжаттардың сыртқы келбетін анықтап, таңбалауға арналаған элементтер жиынтығы.

URL — Файлдың немесе беттің Интернеттегі адресі.

Web-бет — «Бүкілләлемдік өрмектің» негізгі құрылымдық элементі, мәтіндік және (немесе) графикалық ақпаратты, сонымен қатар Internet-тің басқа құжаттарына сілтемелерді қамтитын құжат.

WWW (Word Wide Web) — Интернетте гипермәтіндік ақпараттық-іздеу жүйесі.

1 бит — екілік алфавит (екі символды) символының ақпараттық мөлшері.

Авторизациялау — бұл кез келген ресурсқа (мысалы, құпия сөзді енгізгеннен кейін электрондық поштаға, саусақ ізін сканерден өткізгеннен кейін смартфонды бұғаттаудан шығару және т.б.) кіру мүмкіндігі.

Айналдыру жолағы — құжат терезесінің немесе тізімнің оң жағында немесе төменгі жағында орналасқан. Құжат өзінің терезесіне сыймаған жағдайда құжат бойымен жылжытуға арналған. Бағытты көрсететін жүгіргіш және нұсқауышты қамтиды. Сызғыштан өзгеше болады.

Ақпарат тасымалдаушы — ақпаратты сақтауға арналған материалдық нысан.

Ақпараттық қауіпсіздік — ақпараттық ресурстардың сақталу және ақпараттық ортада қоғам мен тұлғаның заңдық құқығының қорғалу жағдайы.

Ақпараттық модель — нысан немесе процесс туралы ақпараттардың жиынтығы.

Алгоритм — есептің шешімін қамтамасыз ететін нақты әрекеттерді орындау ережелерінің жиынтығы.

Алфавит — символдардың анықталған жиынтығы, символдар алфавиттің әріптері деп аталады.

Бағдарламалау — компьютерлердің бағдарламалық жасақтама етуін әзірлеу бойынша кәсіби іс-әрекет.

Байт — 8 бит жиынтығы, компьютермен бір бүтін ретінде қабылданады.

Бит — табиғатта белгілі ақпараттың ең кіші бірлігі. Биттің мағынасы — 0 немесе 1, оны «ажыратылды-қосылды», «жоқ-иә», «ақиқат-жалған» сияқты түсіндіруге болады.

Гипермәтін — басқа құжаттарға белсенді сілтемелер қамтитын құжат.

Деректер — компьютер жадысында екілік түрінде беріліп, өңделетін ақпарат.

Деректер қоры — бірдей қасиеттер жинағына ие нысандар тобы туралы мәліметтерді реттелген түрде сақтауға мүмкіндік беретін ақпараттық модель.

Деректер қоры жазбасы — мәліметтер қоры өрістерінде орналасқан мәндерді қамтитын кестенің жолы.

Дизъюнкция — әрбір екі қарапайым сөзбен біріктірілген мәлімдеме болып табылатын логикалық операция. Екі мәлімдеменің мәні бір уақытта жалған болған кезде ғана мәні жалған болады және түпнұсқа мәлімдемелерінің кемінде біреуі шын болғанда мәні шын болады.

Код — ақпаратты көрсетуге арналған таңбалар жинағы (шартты белгілер).

Кодтау — ақпаратты код түрінде көрсету процесі.

Кодтық кесте — алфавит символдары мен екілік сандар арасында сәйкестікті орнататын кесте. Бұл сандар символдардың кодтары деп аталады және символдардың компьютерде берілуіне жауапты.

Компьютердің логикалық элементі — бұл қарапайым логикалық функцияны іске асыратын электрондық логикалық схеманың бөлігі.

Компьютерлік желі — байланыс желісі арқылы байланысатын компьютерлер тобы.

Компьютерлік этика — компьютерді қолданатын адамдардың тәртібін зерттейтін пән.

Конъюнкция — әрбір екі қарапайым пікірге құрамдас пікірді сәйкестендіретін логикалық операция және екі бастапқы пікір ақиқат болғанда ғана құрамдас пікір ақиқат болады.

Кроссплатформалық бағдарлама — бұл әртүрлі операциялық жүйелерге (Windows және Linux) арналған нұсқасы бар бағдарлама.

Лексема — белгіш символдармен (әдетте, бос орын немесе тыныс белгілер) бөлінген символдардың тізбегі.

Логика — пікірлер және олардың байланыстары туралы ғылым.

Мультимедиа — ол бағдарламалық қамтамасыз етудің көмегімен компьютерде әдеттегі ақпаратты (мәтін немесе графика) дыбыс және қозғалатын суреттермен (бейнефильмдерді де жасауға болады) біріктіруге мүмкіндік беретін арнайы технология.

Объектілердің қасиеті — объектіні басқа объектілерден айыруға болатын белгілерінің жиынтығы.

Провайдер — пайдаланушыларға Интернетке шығуға мүмкіндік беретін фирма.

Пішін (форма) — ақпаратты енгізу мен көрсетуге арналған деректер қорының объектісі.

Сайт скрипті — бұл ресурста іске асырылатын және порталдың функционалдығын кеңейтетін код. Басқаша айтқанда, скрипт — белгілі бір тапсырманы орындау үшін web-сервер бетінен келетін арнайы сұраныс арқылы сервермен іске қосылуы орындалатын процесс.

Сәйкестендіру — бұл субъектіні оның идентификаторы бойынша тану рәсімі. Басқаша айтқанда, бұл атау, логин немесе нөмір және т.б. анықтамасы.

Символдың ақпараттық салмағы — алфавиттің бір символынан тұратын ақпарат саны.

Сұраныс — жазбаларды іріктеу немесе басқа операцияларды орындау үшін ДҚБЖ-не жүгіну.

Сұрыптау — (ағылшынша «*sorting*» — «жіктеу, реттеу») — таңдалған критерийге байланысты ақпаратты топтарға біртіндеп орналастыру немесе бөлу.

Тег («tag» — «сілтеме, таңба») — бұл HTML құжаттың белгілі элементін сипаттайтын код үзіндісі және ол «*< >*» жақша арқылы жазылады.

Терістеу — логикалық операция, әрбір қарапайым пікірді жоққа шығаратын құрама пікірді сәйкестендіру.

Түпнұсқалық (аутентификация) — бұл түпнұсқалылықты тексеру процедурасы. Басқаша айтқанда, пайдаланушыны құпия сөз арқылы, хатты электрондық пошта және т.б. арқылы тексереді.

Пайдаланылған әдебиеттер

1. *Андреанов В. В.* Обеспечение информационной безопасности бизнеса. М.: Альпина Паблишерз, 2011. 373 с.
2. *Бейли Л.* Изучаем SQL. СПб.: Питер, 2012. 592 с.
3. *Бирюков А. А.* Информационная безопасность: защита и нападение. М.: ДМК Пресс, 2012. 474 с.
4. *Бураков П. В., Петров В. Ю.* Введение в системы баз данных. СПб.: Питер, 2015. 186 с.
5. *Компьютерные сети. Принципы, технологии, протоколы: Учебник.* 5-е изд. СПб.: Питер, 2016. 992 с.
6. *Лутц М.* Изучаем Python. Символ-Плюс, 2018.
7. *Лутц М.* Программирование на Python. Том 1. Символ-Плюс, 2012.
8. *Мейер Э. А.* CSS-каскадные таблицы стилей. Подробное руководство. Символ-Плюс, 2013.
9. *Нестеров С. А.* Базы данных: учебник и практикум. М.: Издательство Юрайт, 2018. 230 с.
10. *Седер Н.* Python. Экспресс-курс. СПб.: Питер, 2018.
11. *Сергеев А.* Основы локальных компьютерных сетей. 2016.
12. *Таненбаум Э., Уэзеролл Д.* Компьютерные сети. 5-е изд. 2016.
13. *Филиппов Б. И.* Информационная безопасность. Основы надежности средств связи. М.: Директ-Медиа, 2019. 241 с.
14. *Фримен Э.* Изучаем HTML, XHTML и CSS. СПб.: Питер, 2019.
15. *Царев Р. Ю.* Программные и аппаратные средства информатики. Красноярск, 2015. 160 с.

МАЗМҰНЫ

I тарау. КОМПЬЮТЕРЛІК ЖЕЛІЛЕР ЖӘНЕ АҚПАРАТТЫҚ ҚАУІПСІЗДІК

§ 1. Компьютерлік желілердің жұмыс жасау принциптері. Желі компоненттері.....	5
§ 2. Компьютерлік желілердің жұмыс істеу принциптері. IP-адрес.....	8
§ 3. Компьютерлік желілердің жұмыс істеу принциптері. DNS	11
§ 4. Компьютерлік желілердің жұмыс істеу принциптері. Жеке виртуалды желі.....	13
§ 5. Ақпараттық қауіпсіздік	15
§ 6. Ақпаратты қорғау әдістері	19
§ 7. Тұлғаны сәйкестендіру (идентификациялау) әдістері.....	22

II тарау. ДЕРЕКТЕРДІ ҰСЫНУ

§ 8. Сандарды бір санау жүйесінен екінші санау жүйесіне аудару	27
§ 9. Практикум. Сандарды бір санау жүйесінен екіншісіне аудару	34
§ 10-11. Логикалық операциялар. Ақиқат кестесін құру	35
§ 12. Практикум. Логикалық операциялар. Ақиқат кестесін құру	41
§ 13. Компьютердің логикалық элементтері	43
§ 14. Компьютердің логикалық негіздері	46
§ 15. Практикум. Логикалық есептерді шешу	49
§ 16. Мәтіндік ақпаратты кодтау принциптері	55
Өзіңді тексер!.....	59

III тарау. АЛГОРИТМДЕУ ЖӘНЕ БАҒДАРЛАМАЛАУ

§ 17-18. Пайдаланушы функциялары мен процедуралары	63
§ 19. Пайдаланушы функциялар мен процедуралардың мысалдары	67
§ 20. Жолдармен жұмыс жасау	70
§ 21. Жолдармен жұмыс жасаудың мысалдары.....	74
§ 22. Файлдармен жұмыс жасаудың мысалдары.....	78
§ 23. Файлдармен жұмыс жасаудың мысалдар	80
§ 24-25. Сұрыптау әдістері.....	84
§ 26—28. Сұрыптау әдістерінің мысалдары	88
§ 29-30. Графтар және оларды алгоритмдеуде қолдану	92
Өзіңді тексер!.....	96

IV тарау. АҚПАРАТТЫҚ ЖҮЙЕЛЕР

§ 31. Bigdata.....	100
§ 32-33. Bigdata. Жобалау жұмысы	103
§ 34-35. Деректер қорының негізгі ұғымдары.....	107
§ 36. Деректер қорындағы алғашқы кілт	112
§ 37-38. Деректер қорын жасау.....	115
§ 39—41. Деректер қорын әзірлеу. Жобалау жұмысы.....	127
§ 42-43. Пішіндер.....	128
§ 44-45. Есеп.....	134

§46-47. Сұраныстар.....	135
§48-49. Құрылымдалған сұраныстар.....	141
§50—52. Құрылымдалған сұраныстар. Жобалау жұмысы	143
Өзінді тексер!.....	147

V тарау. WEB-БАҒДАРЛАМА

§53-54. HTML (HyperText Markup Language) web-сайттарын өзірлеу әдістері	151
§ 55. Мәтінді форматтау (қаріп, абзац, тізімдер).....	155
§56-57. Кестелер.....	158
§58—60. CSS (Cascading Style Sheets).....	162
§ 61. Мультимедианы енгізу.....	167
§ 62. Скриптерді қолдану.....	172
§ 63. Web-беттің деректер қорымен байланысы	175
Өзінді тексер!.....	177
Глоссарий	178
Пайдаланылған әдебиеттер	181



Учебно-методическое издание
Кольева Наталья Станиславовна
Шевчук Елена Владимировна

ИНФОРМАТИКА

Учебник для 10 классов
естественно-математического направления
общеобразовательных школ
(на казахском языке)

Редакторы *А. Сабдалиева*
Көркемдеуші редакторы *А. Сланова*
Техникалық редакторы *Л. Садықова*
Корректоры *Ж. Баймағамбетова*
Компьютерде беттеген *И. Алмабаева*

Баспаға Қазақстан Республикасы Білім және ғылым министрлігінің
№ 0000001 мемлекеттік лицензиясы 2003 жылы 7 шілдеде берілген

ИБ № 5881

Басуға 26.06.19 қол қойылды. Пішімі 70×100^{1/16}. Офсеттік қағаз.
Қаріп түрі «SchoolBook Kza». Офсеттік басылыс. Шартты баспа табағы 14,84.
Шартты бояулы беттаңбасы 60,01. Есептік баспа табағы 6,8.
Таралымы 15 000 дана. Тапсырыс №

«Мектеп» баспасы, 050009, Алматы қаласы, Абай даңғылы, 143-үй
Факс: 8(727) 394-37-58, 394-42-30
Тел.: 8(727) 394-42-34
E-mail: mektep@mail.ru
Web-site: www.mektep.kz