

Қазақстан Республикасы Білім және ғылым министрлігі ұсынған

Г.И.Салғараева
К.А.Калымова
Ж.А.Орынтаева

ИНФОРМАТИКА

Жалпы білім беретін мектептің 9-сыныбына арналған оқулық

9



ӘОЖ 373.167.1
КБЖ 32.973 я 72
С 18

Ғылыми кеңесшісі:

Ж.У.Кобдикова – педагогика ғылымдарының докторы

С 18 Салғараева Г.И., ж.б.
Информатика: Жалпы білім беретін мектептің 9-сыныбына арналған оқулық. / Г.И.Салғараева, К.А.Калымова, Ж.А.Орынтаева – Нұр-Сұлтан: «Арман-ПВ» баспасы, 2019. – 176 бет.

ISBN 978-601-318-206-3

Оқулық негізгі орта білім беру деңгейінің жаңартылған мазмұндағы үлгілік оқу бағдарламасына сәйкес оқушылардың жас ерекшеліктері ескеріле отырып жазылды. Оқулық тілі жеңіл, мазмұны қосымша мәліметтермен және CD дискімен қамтылған.

ӘОЖ 373.167.1
КБЖ 32.973 я 72

ISBN 978-601-318-206-3

© Салғараева Г.И.,
Калымова К.А.,
Орынтаева Ж.А., 2019
© «Арман-ПВ» баспасы, 2019

Барлық құқығы қорғалған. Баспаның рұқсатынсыз көшіріп басуға болмайды.

ШАРТТЫ БЕЛГІЛЕР

Жаңа тақырыпты меңгеру тапсырмалары – функционалдық сауаттылықты қалыптастыру тапсырмалары

- | | | | |
|---|--------------------------|---|-----------------------|
| 1 | Сұрақтарға жауап берейік | 5 | Компьютерде орындайық |
| 2 | Ойланайық, талқылайық | 6 | Ой бөлісейік |
| 3 | Талдап, салыстырайық | 7 | Үй тапсырмасы |
| 4 | Дәптерге орындайық | | |

Естеріңізге түсіріңдер:

Жаңа тақырыпты түсінуге бағытталған қайталау сұрақтары

Меңгерілетін білім:

Тақырыптағы игерілетін мәліметтер; күтілетін нәтижелер, оқу мақсаттары

Шығармашылық тапсырма

Ізденімдік деңгейдегі тапсырмалар

Қызықты ақпарат

Материалды жеңіл меңгеруге жетелейтін ақпараттар

Сөздік

Ғылыми ұғымдардың үш тілдік нұсқасы

Маңызды мәлімет

Тақырыпты тереңірек түсінуге қажетті маңызды мәліметтер



Назар аудар

Электронды қосымша жүктелген CD қолжетімсіз болған жағдайда, қосымшаны *arman-pv.kz* сайтынан тауып, өз компьютеріңізге жүктеп алуыңызға болады

Алғы сөз

Құрметті оқушылар!

Биылғы оқу жылында информатика курсын жалғастырып, осы пәнге қатысты білімдеріңді тереңдете түсесіңдер.

Оқулықтың мазмұны «Информатика» пәнінің түрлі саласын қамтиды. Оқулық бес бөлімнен тұрады.

Бірінші бөлімде сендер ақпарат ұғымы, ақпараттың қасиеттері, компьютерді ұзақ уақыт бойы қолдану барысындағы қауіптерді сыни түрде бағалау, желідегі қауіпсіздік шаралары мен этикет ережелерін үйренесіңдер. Сондай-ақ осы бөлімде бұлттық технологиялар туралы, бұлттағы ресурстардың түрлері мен онлайн ресурстарын пайдаланудың артықшылықтарын қарастырасыңдар.

Екінші бөлімді оқу барысында компьютердің негізгі конфигурация параметрлерін және компьютерді таңдауға әсер ететін сипаттамаларын білесіңдер.

Үшінші бөлімнің мазмұны деректер базасы мен олардың мүмкіндіктерін зерттеп, деректердің түрлері мен нысандарын қарастыруға арналған.

Төртінші бөлімде Python программалау тілінде өз білімдеріңді кеңейте түсесіңдер.

Бесінші бөлім Python программалау тілінде PyGame кітапханасының көмегімен 2D ойынын құруға бағытталған.

Әрбір тақырыптан соң тапсырмалар кешені ұсынылған. «Сұрақтарға жауап берейік», «Ойланайық, талқылайық», «Талдап, салыстырайық», «Дәптерге орындайық» айдарларында берілген тапсырмалар тақырыптың теориялық бөлімін игеруге арналған. Ал «Компьютерде орындайық», «Ой бөлісейік» тапсырмалары практикада алған білімдеріңді бекітіп, оны күнделікті өмірде кездесетін жағдайларды шешуде пайдалануға көмектеседі. «Үй тапсырмасы» тақырыпты үйде қайталап, меңгеруге арналған тапсырмалардан тұрады.

Оқулыққа қосымша электронды оқу құралын (CD диск) пайдаланып, пән бойынша алған білімдеріңді толықтыра түсетін интерактивті қызықты тапсырмалар мен тест сұрақтарын орындайсыңдар.

Сәттілік тілейміз!

1-БӨЛІМ

АҚПАРАТПЕН ЖҰМЫС ЖАСАУ

Күтілетін нәтижелер:

- ақпараттың қасиеттерін анықтау;
- компьютерде ұзақ уақыт жұмыс істеу тәуекелін сыни түрде бағалау;
- бұлттық технологияларды қолданатын құжаттармен бірлескен жұмысты жүзеге асыру;
- желіде этикалық және құқықтық нормалардың бұзылу салдары туралы айту.

§ 1. Ақпараттың сипаттары

Естеріңе түсіріңдер:

- ақпарат дегеніміз не?
- қандай ақпарат түрлерін білесіңдер?
- ақпараттың көлемі деген не?
- ақпараттың қандай қасиеттерін білесіңдер?

Ақпараттың қасиеттері –
Свойства информации –
Information Properties

Меңгерілетін білім:

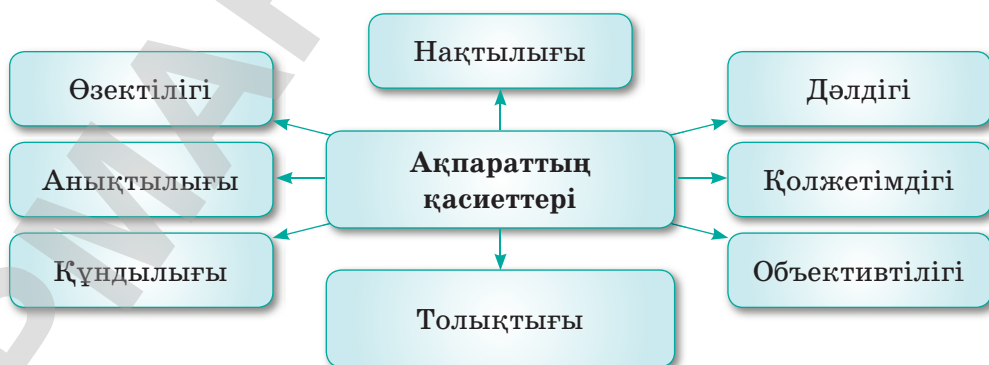
- ақпараттың негізгі қасиеттері;
- ақпараттың қасиеттерін қолдану;
- әртүрлі жағдайдағы ақпараттың қасиеттері.

Ақпараттың негізгі қасиеттері

Кез келген нысан сияқты ақпараттың да өзіндік қасиеттері бар. Ақпараттың табиғат пен қоғам нысандарынан ерекшеленетін белгісі – оның дуалдылығы (екіжақтылығы): біріншіден, деректердің объективтілігіне байланысты ақпарат та объективті болады, екіншіден, қолданылған әдістердің басымдығына қарай субъективті болады.

Информатикада ақпараттың бірнеше қасиетін ерекше атап көрсетуге болады: объективтілігі, сенімділігі, толықтығы, нақтылығы, өзектілігі, пайдалылығы, құндылығы, уақытылы болуы, дұрыстығы, қолжетімдігі, анықтылығы және т.б.

Ақпараттың негізгі қасиеттерін қарастырайық (1-сызба).



1-сызба. Ақпараттың негізгі қасиеттері

Өзектілігі. Дер кезінде алынған ақпарат дәл (тура) ақпаратты береді. *Мысалы:* «Қазіргі уақытта сіздің есепшотыңызда 1360 теңге бар».

Дәлдігі. Бұл – қажетті дәрежедегі нақты нысанға байланысты ақпараттың сәйкестік дәрежесін сипаттайтын қасиет. Дәлдік – жадының негізгі қасиеті. *Мысалы,* « $2 \cdot 25 = 50$ ».

Нақтылығы. Істің ақиқат жағдайын бейнелейтін ақпарат. *Мысалы,* нақты ақпарат – «Бір тәулікте 24 сағат бар»; нақты емес ақпарат – «Марста тіршілік бар».

Құндылығы. Құнды ақпарат қойылған мәселені шешіп, мақсатқа жетуге мүмкіндік береді. *Мысалы,* «Бүгін жаңбыр жауады, қолшатыр алуды ұмытпаңдар».

Толықтығы. Шешім қабылдау үшін мәліметтердің жеткілікті болуын немесе солардың негізінде жаңа деректер құруын сипаттайды. *Мысалы,* толық ақпарат: «Анамның туған күні – 4 маусым»; толық емес ақпарат: «Анамның туған күні жазда».

Объективтілігі және субъективтілігі. Объективті ақпарат – тексеруге және өлшеуге болатын ақпарат. Субъективті ақпараттан айырмашылығы: адами факторларға қатыссыз ақпарат. *Мысалы,* объективті ақпарат: «Қазақстанның астанасы – Нұр-Сұлтан»; субъективті ақпарат: «9-сыныптағы барлық балалар жақсы оқиды».

Қолжетімдігі. Қажет болған жағдайда ақпаратты алу мүмкіндігі. *Мысалы,* егер сенде компьютер жоқ болса, онда CD-ге қарағанда кітаптағы ақпарат саған көбірек қолжетімді.

Анықтылығы. Ақпарат алушы үшін қолжетімді тілде көрсетілген ақпарат, анық ақпарат болып табылады. *Мысалы,* өрт кезіндегі эвакуация туралы ана тілінде жазылған ақпарат анық ақпарат болып табылады; түсініксіз ақпарат: 火灾 报警 信号 撤离.

Түрлі жағдайларға байланысты бір ақпарат өртүрлі қасиеттерге ие болуы мүмкін. Мұндай жағдайда адам немесе техникалық құрал ақпараттар ағымындағы ақпараттық үрдістердің ішінен әрбір жағдай үшін ақпараттардың ең тиімді қасиетін таңдайды.

1

Сұрақтарға жауап берейік

1. Ақпараттың қасиеттерін анықтаңдар.
2. Ақпаратты оның қасиеттері арқылы бағалауға бола ма?
3. Хабарламалардың ақпараттық мазмұнын анықтау мүмкін бе?

2

Ойланайық, талқылайық

1. Неліктен ақпараттың қасиеттері адам үшін маңызды?
2. Ақпараттар қасиетінің өзгеру себебі неде?

3

Талдап, салыстырайық

Ақпараттың дәлдігі мен құндылығы, қолжетімдігі мен өзектілігі арасындағы айырмашылықтарды салыстырыңдар.

4

Дәптерге орындайық

1. Берілген ақпараттың нысандардың ақпараттық сипаттамаларын беріңдер:
 - а) тақырып бойынша эссе;
 - ә) геометриялық тапсырмалар;
 - б) тарих пәнінен реферат.
2. Ақпараттық қасиеттердің атауларын анықтамаларымен сәйкестендіріңдер.

Ақпарат қасиеті	Ақпараттық меншіктің мәні
Өзектілігі	Шешімдер қабылдау үшін ақпараттың жеткіліктілігі
Анықтылығы	Біреудің пікіріне тәуелді еместігі
Объективтілігі	Істердің нақты жағдайын көрсетуі
Қолжетімдігі	Ақпарат алушыға түсінікті тілде берілетін ақпарат
Толықтығы	Алушы түсінетін тілде көрсетілген ақпарат
Дәлдігі	Ақпаратты алу мүмкіндігінің болуы
Нақтылығы	Ақпараттың нақты нысанға сәйкестік дәрежесі

5

Компьютерде орындайық

Топқа бөлініп, берілген тапсырманы орындаңдар:

1. Ақпараттың қасиеттеріне байланысты презентация дайындаңдар.

2. Берілген кестені MS Word-қа енгізіңдер.

Ақпараттың қасиеттері	Анықтамасы	Мысалдары
Өзектілігі		
Анықтылығы		
Объективтілігі		
Қолжетімдігі		
Толықтығы		
Дәлдігі		
Нақтылығы		
Құндылығы		

6

Ой бөлісейік

Күнделікті өмірде ақпараттың қандай қасиеттерін жиі кездестіресіңдер?

7

Үй тапсырмасы

CD дискідегі ақпараттың қасиеттеріне байланысты тапсырмаларды орындаңдар.

Шығармашылық тапсырма

Мысал келтіре отырып, ақпараттың басқа қасиеттерін қосымша зерттеңдер.

§ 2. Компьютерде ұзақ уақыт жұмыс істеу тәуекелін сыни түрде бағалау

Естеріңе түсіріңдер:

- ақпараттың қандай қасиеттерін білесіңдер?
- ақпаратты қасиеттеріне қарай бағалауға бола ма?
- неліктен адамға маңызды ақпарат керек?

Меңгерілетін білім:

- адам ағзасына компьютердің тигізетін кері әсерлері;
- компьютердің тигізетін кері әсерінен қорғанудың әдістері;
- компьютерді ұзақ уақыт қолданудың қауіптері.

Интерфейс – Интерфейс – Interface

Адам мен компьютердің өзара әрекеттесуі –
Человеко-компьютерное взаимодействие – HCI
(human-computer interaction)

XX ғасырдың соңында ғылыми-техникалық прогрестің дамуымен, компьютер және компьютерлік технологиялар қарқынды дами бастады. Қазіргі өмірді ноутбук, планшет, ұялы телефонсыз елестету қиын. Олар біздің өмірімізден айтарлықтай орын алып, кейде біз оларға тәуелді болатынымызды түсінбей қаламыз.

Компьютердің адам денсаулығына әсері біздің заманымыздың маңызды мәселесі болып табылады. Мамандардың пікірінше, адамдардың компьютермен байланысын қатаң қадағалап, компьютерде жұмыс істеудің қарапайым ережелерін ұстану керек. Егер компьютерде күні-түні үзіліссіз отырсаңдар, онда денсаулыққа айтарлықтай зиян келтіресіңдер (*1-кесте*):

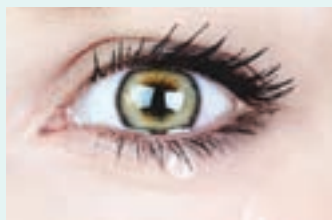
1-кесте. Компьютердің адам ағзасына зияны



Бұлшық еттер проблемасы. Компьютерде ұзақ уақыт отыру кезінде адамның ағзасындағы қан айналымы бұзылады. Бұлшық еттер қозғалыссыз болған кезде тонусын жоғалтады. Тонус аз болған сайын, сүйекке, буындарға және омыртқаға көп ауырлық түседі.



Омыртқаға әсер ету проблемасы. Тап-жылмай ұзақ уақыт отырып жұмыс жасау ерте ме, кеш пе омыртқа ауруына, арқадағы ауырсынуға, остеохондрозға және сколиозға әкеледі. Денені ұзақ уақыт бойы бірқалыпты ұстау қимыл-тірек аппаратына нұқсан келтіреді, қозғалысы аз өмір салты адам денсаулығына зиян.



Көзге әсер ету проблемасы. Компьютермен жұмыс істеу кезінде көз торына және бұлшық еттеріне күш түседі. Көз ауырсынып, жасаурай бастайды, монитордағы сурет айқын көрінбейді. Сондай-ақ бұл көру қабілетінің нашарлауына әкелуі мүмкін.



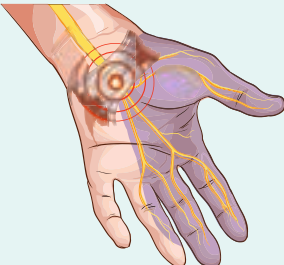
Компьютер және асқорыту жүйесі. Адам мен компьютердің өзара әрекеттестігі жиі болған жағдайда тамақтану режимі бұзылады. Мұндай жағдайда адамдар уақыт өте келе артық салмақ қосып, асқазан-ішек жолдары ауруына шалдығуы мүмкін.



Жүйке жүйесіне әсері. ДК – баяу жүктелетін, арасында ақаулар беретін және істен шығып қалуы мүмкін техника. Мұндай жағдайларда шыдамсыз адамдар ашуланғыш келеді. Айналасындағыларға дауыс көтеру және агрессияның басқа да түрлерінің пайда болуы – жүйке жүйесінің сыр беруінің белгісі.



Компьютердің әсері заманауи зерттеулерде көрсеткендей, жүрек-қантамыр жүйесінің қауіпті ісіктері мен ауруларының пайда болуына себеп болуы мүмкін. Адам басқа тұрмыстық техникадан гөрі компьютерлерден денсаулыққа қауіпті электромагниттік сәуле алады.

	Компьютер және тыныс алу органдары. Компьютермен ұзақ уақыт жұмыс істеу салдарынан тыныс алу органдарында аллергиялық сипатқа ие аурулар пайда болады, өйткені компьютер жұмысы кезінде монитор корпусы мен процессор қызып, ауаға зиянды заттарды бөледі.
	Білезік туннельдік синдромы – білезіктегі нервтің прогрессивті қысылуынан ауырсыну жағдайының туындауы. Күні бойы компьютерде жұмыс істеп, қолдың ұйып қалғанын байқамаймыз, кейде кенет өткір және шаншып ауырғанын сезінеміз. Осы аталған симптомдар – білезік туннельдік синдромы.
	Компьютерлік құмар ойындарға тәуелділік қазіргі қоғамның ең маңызды мәселелерінің біріне айналып отыр. Ол адамды әлеуметтік қарым-қатынас және құндылықтарды бағалау сияқты қасиеттерден айырады. Жүйке жүйесі әлі нәзік және әрі тұрақсыз болып келетін балалар көбіне осындай топтарда жиі кездеседі.

Емдік гимнастика

Арқадағы бұлшық еттердің дамуына арналған әртүрлі бағыттардағы жаттығулар көп. Бұлардың ішінде жүзуге және турникке тартылуға арналған жаттығулар өте пайдалы. Компьютерде жұмыс істеген жағдайда жиі созылып (керіліп) көріңдер және мүмкіндігінше көбірек қимылдаңдар.

Қолға арналған жаттығулар

- Алақанды күшпен ашып-жұму қимылдарын жасаңдар.
- Білектеріңді босаңсытып, оларды жан-жаққа және жоғарыға көтеріп сілкіңдер.
- Саусақтарыңды айқастырып, қысу қимылдарын жасаңдар.
- Білектеріңді айналмалы қозғалыс арқылы ішке және сыртқа айналдырыңдар.
- Саусақтарыңды бір-бірлеп алға созып, сағат тілі бойынша және кері қарай кезек-кезек айналдырыңдар.

1

Сұрақтарға жауап берейік

1. Компьютермен жұмыс жасау кезінде адам ағзасына кері әсерін тигізетін факторларды атаңдар.
2. Компьютерде ұзақ отырудың салдары қандай келеңсіз жағдайларға әкеледі?
3. Компьютермен жұмыс жасаудағы техникалық қауіпсіздік ережелері қандай?
4. Білезік туннельдік синдромына шалдығудың алдын алу шаралары қандай?

2

Ойланайық, талқылайық

1. Не себепті компьютерде ұзақ отыратын адам түрлі ауруға шалдығуы мүмкін?
2. Сенің досың компьютерлік ойындарды ойнап, жеңіліске ұшыраған жағдайда, ашулы, ызалы күйге түссе, сен оған қалай көмектесер едің?
3. Не себепті компьютермен жұмыс барысында жиі-жиі қозғалып, гимнастика жасап тұру керек?

3

Талдап, салыстырайық

Компьютердің адам бұлшық еттері мен жүйке жүйесіне, көзге, тыныс алу органдарына, жалпы ағзаға зиянды әсерін талдаңдар.

4

Дәптерге орындайық

1. Топқа бөлініп, компьютердің адам ағзасына кері әсерлеріне мысал келтіріп, дәптерге жазыңдар.
 - 1) Көзге түсетін ауырлық;
 - 2) Бұлшық ет және буындарға байланысты мәселелер;
 - 3) Компьютерлік стресс синдромы;
 - 4) Психикаға әсері.
2. Компьютерде жұмыс істеу кезіндегі техникалық қауіпсіздік ережелеріндегі көп нүктенің орнын толтырыңдар.
 - 1) Компьютерде ... жұмыс істемейдер;
 - 2) Компьютерге ... тимеуін қадағалаңдар;
 - 3) Мониторды ... ұстамаңдар;
 - 4) Компьютерде жұмыс істегенде ... сақтаңдар;
 - 5) Компьютерді ... ұстаңдар;
 - 6) Компьютерді ... өшірмейдер.

5

Компьютерде орындайық

1. Компьютердің денсаулыққа зияндығы мен одан қорғану шараларына байланысты кесте құрыңдар (MS Word).
2. Компьютерде ұзақ отырғанда пайда болатын аурулардың алдын алу үшін, компьютерді пайдаланудың ережелерін сақтау нұсқаулығын жазыңдар (MS Word).
3. CD дискіден осы тақырыпқа сәйкес тапсырманы орындаңдар.

6

Ой бөлісейік

Сабақта не білдіңдер? Өз ойларыңды достарыңмен бөлісіңдер. Алған жаңа білімдеріңді күнделікті өмірде қандай жағдайда қолдануға болады? Мысал келтіріңдер.

7

Үй тапсырмасы

Компьютерде ұзақ уақыт жұмыс істегенде пайда болатын ауруларды болдырмау үшін жасалған мониторлардың стандарты туралы мәлімет жазыңдар (салыстырмалы кесте құру).

Шығармашылық тапсырма

«Компьютердің адам ағзасына кері әсері» тақырыбына презентация жасаңдар.

§ 3. Құжаттармен бірлесіп жұмыс жасау

Естеріңе түсіріңдер:

- компьютердің адам ағзасына кері әсері қандай?
- компьютердің кері әсерінен қорғану әдістері туралы айтыңдар;
- компьютерде үзіліссіз жұмыс істейтін адамда қандай аурулар пайда болуы мүмкін?

Бұлттық технологиялар –
Облачные технологии –
Cloud technologies

Меңгерілетін білім:

- бұлттық технологиялар;
- бұлт моделі;
- бұлттық технологиялар тиімділігі;
- бұлттық ресурстардың түрі.



1-сурет. Бұлт моделі

Соңғы кезде үрдіс алып отырған жаңа технологияның бірі – **бұлттық технологиялар (1-сурет)**.

Бұлттық технологиялар, cloud computing (ағылш. *cloud* – «бұлт», *computing* – «есептеулер», *technology* – «технология») ыңғайлы интерфейсті пайдалана отырып тұтынушыға Интернет немесе жергілікті желі арқылы белгілі бір ресурстармен (программалар, деректер, файлдар және т.б.) қашықтан қатынауды ұсынатын қызмет.

Бұлттық технологиялар пайдаланушыға онлайн режимде Интернет арқылы ақпаратты өңдеуге мүмкіндік береді. Бұлттық технологиялар – әртүрлі қызметтердің жиынтығы үшін жалпыланған атау.

Бұлттық технологияларды қолданудың әдеттегі программалардан басты айырмашылығы – тұтынушы өз компьютерінің ресурстарын емес, өзіне Интернет қызметі ретінде берілген шалғайдағы қуатты серверлердің ресурстарын пайдалануында. Сол арқылы тұтынушы өз дереккөздерімен жұмыс істеуіне толық мүмкіндік алады, бірақ сол дереккөздер орналасқан операциялық жүйеге, программалар базасына, есептеуіш серверлердің жұмысына еш кедергі келтіріп, оны өзгерте алмайды.

Бұлттық есептеулер мақсаты мен міндетіне қарай төрт түрге бөлінеді.

Жеке бұлттар (private cloud) – жеке кәсіпорындардың өзіне ғана, сондағы жеке тұлғалар мен олардың тұтынушыларының жұмыс істеуіне арналған инфрақұрылым. Жеке-леген бұлттар осы ұйымдардың серверлерінде, ал басқа тұлға үшін – ірі провайдерлердің орталық серверлерінде орналасады (2-сурет).



2-сурет. Жеке бұлттар

Ортақ бұлттар (public cloud) – Интернетке қолжетімдігі бар барлық пайдаланушылардың ресурстарымен еркін жұмыс істеуіне арналған инфрақұрылым. Интернет желісіндегі Google, Yahoo т.с.с. электрондық пошта жүйелерін, Facebook, Twitter сияқты әлеуметтік желілерді ортақ бұлттарға мысал ретінде қарауға болады (3-сурет).

Қоғамдық бұлттар (community cloud) – мақсаттары ортақ және бірдей сұраныстарға біріккен қоғамдық тұтынушыларға арналған инфрақұрылым (4-сурет).

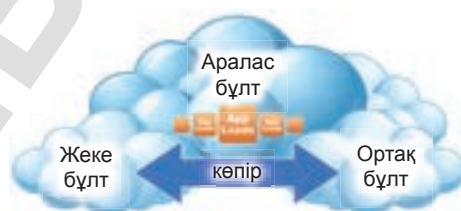


3-сурет. Ортақ бұлттар



4-сурет. Қоғамдық бұлттар

Аралас бұлттар (hybrid cloud) – екі немесе одан көп бұлт түрлерінің (жеке, ортақ, қоғамдық) аралас комбинациясы. Бұлт түрінің осы моделін әр жерде, әртүрлі елді мекенде т.б. орналасқан филиалдары, көптеген программалық жүйелері бар ірі компаниялар қолдануы мүмкін (5-сурет).



5-сурет. Аралас бұлттар

Бұлттық технологиялар қарапайым пайдаланушыларға функционалдығы жағынан бірдей қызметті көрсетеді: көлемі шағын программаны жүктеу және орнату, бұлттағы файлдарды сақтау үшін бума жасау. Бұл әрекетті орындау үшін, пайдаланушы жергілікті компьютерде немесе «Бұлтта» орналасқан файлдар мен бумаларды синхрондау параметрлерін ретке келтіру қажет. Web-интерфейс арқасында бұлтқа сақталған барлық деректер әрқашан кез келген құрылғыдан қолжетімді болады. Электрондық пошта арқылы тиісті сілтемені алған кез келген адам осы бумалар мен файлдарды еркін пайдалана алады.

Файлдарды бұлттық сақтау орны – бұлттық сақтау қызметін көрсететін, жеткізуші жабдықтарында орналасқан, Интернет желісіндегі қолжетімді ақпараттарды сақтайтын орын.

Бұлттық ресурстардың түрлері көп. *Мысалы:* Dropbox, SugarSync, Box.net, GoogleDocs, OpenDrive, Windows Live SkyDrive, Яндекс.Диск т.б. (6-сурет).



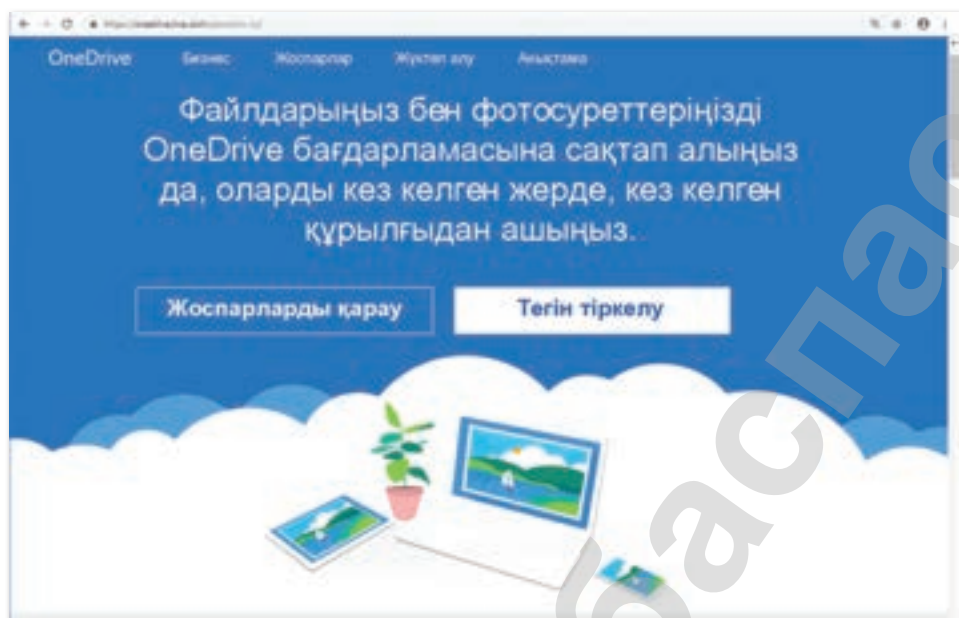
6-сурет. Бұлттық ресурстардың түрлері

Dropbox – компьютерді Интернетке қосқан кезде бірлесіп жұмыс істеу мүмкіндігі бар файлдарды сақтауға арналған сервис. Бұлт серверімен бірге синхронды бума автоматты түрде жасалады.

SugarSync бұлттық деректерді сақтайды, пайдаланушы ақпаратына сақтық (резервтік) көшірме жасайды.

Box.net – дербес аккаунттар мен кәсіпорындарға арналған файлдарды сақтау мен орналастыруды қамтамасыз ететін бұлттық сервис.

Windows Live SkyDrive (OneDrive) – Microsoft Office құжаттарымен және оның web-браузерлерде жұмыс істейтін Microsoft Office Web Apps web-нұсқасымен бірлесіп онлайн жұмыс істеуге арналған қызмет (7-сурет).



7-сурет. OneDrive бұлттық қызметінің жеке беті

Яндекс.Диск – Яндекс компаниясына тиесілі бұлттық сервис, пайдаланушыларға өз деректерін «бұлттық» серверлерде сақтауға және тасымалдауға мүмкіндік береді.

OpenDrive – бұлттық файлдар қоймасы, оларды басқаруға және олардың нұсқаларын қолдануға, сақтық (резервтік) көшіруге және бірлесіп жұмыс істеуге мүмкіндік береді.

Syncplicity кроссплатформалық ортада программалық кешенді синхронизациялау үшін қолданылады.

GoogleDocs – пайдаланушының құжаттармен тегін жұмыс істеуіне, Google қосымшаларын ыңғайлы және қарапайым пайдалануына арналған онлайн-кеңсе. Жасалған құжаттарды кез келген құрылғыдан ашып, өңдеп, жөнелтіп, бұлтта сақтап қоюға болады. Google Docs қолданушылары үшін түрлі қосымшалар мен сайттардың орнына пайдалануға болатын арнайы өзіндік қосымшалар жасалады.

«Бұлттық» сервистің мүмкіндіктеріне қол жеткізу үшін, қарапайым тіркеу процедурасынан өту керек. *Мысалы:* Google Docs қызметін пайдалану үшін міндетті түрде өзіңнің аккаунтын құру қажет.

Аккаунт – сайтқа кіру үшін пайдаланушының жеке ақпараты сақталатын есептік жазба.

Google Docs қызметінде аккаунт құру үшін пайдаланушыда электрондық пошта болуы тиіс. *Www.google.com* сайтында орналасқан Google іздеу жүйесінің негізгі бетін ашып, *Қосымша* сілтемесін басамыз. Осыдан кейін қосымша мәзір ашылады, онда *Құжаттар* сілтемесін таңдау қажет. Google Docs бұлттық қызметінің жеке беті ашылады.

Бұлттық қызмет Google Docs-та құжаттарды онлайн оңай жасауға, ортақ пайдалануға және өзгертуге мүмкіндік береді.

Google құжаттарының көмегімен мәтіндік құжаттар, электрондық кестелер, презентациялар жасауға болады. Жаңа кесте жасау үшін Google Docs негізгі бетіндегі *Жасау* түймесін басып, қосымша мәзірде *Кесте* батырмасын таңдау керек. Браузерде бос кесте автоматты түрде ашылады.

Презентацияны жасау үшін пайдаланушының басты бетінде орналасқан *Жасау* батырмасын басып, қосымша мәзірдегі *Презентация* элементін таңдайды.

Google Docs бұлттық қызметінде графикалық кескіндермен жұмыс істеу үшін web-қосымшаны пайдаланады. Графикалық нысанды құру үшін, пайдаланушының басты бетінде орналасқан *Жаңа* батырмасын басамыз және пайда болған мәзірде *Сурет* элементін таңдау керек. Жаңа бос графикалық файл ашылады.

Онлайн режимде OneDrive программасы пайдаланылады, бұл Skype (Lync) программасын пайдаланушылардың орналасуына қарамастан байланысты жеңілдетуге арналған инновациялық сервисті пайдалануға мүмкіндік береді.

Skype (Lync) мынадай мүмкіндіктерді ұсынады: үнемі және барлық жерде қосылу; пайдаланушыларға Интернет байланысы бар кез келген жерден сенімді түрде байланысу; шартты желіге автоматты түрде бейімделу.

Бұлттық технологияларды пайдаланудың тиімді жақтары:

- күрделі мәселелерді шешу үшін бұлттық технологиялардың көмегімен, түрлі сервердегі программаларды пайдалану мүмкіндігі;
- компьютер арқылы кез келген уақытта әлемнің кез келген жерінен тұтынушы деректерінің ақпарат көздеріне кіру мүмкіндігі;
- кедергісіз ақпарат алмасу мүмкіндігі және тұтынушылардың өзара бірлесіп жұмыс істей алуы;

- басқа жеке компьютерлік программаларға қарағанда, бұлттық қызметтер, негізінен, тегін немесе салыстырмалы түрде арзан;
- кейбір жобаларды «бұлтқа шығарудың» әсіресе ірі компанияларға тиімді болатын жағы – программалық жасақтамаларды басқаруға, қолдауға, жаңартуға, лицензиялауға кететін шығындарды үнемдеуінде болып табылады.

Бұлттық технологиялардың кемшіліктері:

- тұтынушылар cloud computing қызметін көрсетуші ұйымдарға тәуелді болады;
- жаңа («бұлттық») монополистердің пайда болуы.

1

Сұрақтарға жауап берейік

1. Бұлттық технологиялардың қандай түрлерін білесіңдер?
2. Бұлттық технологиялардың қандай пайдасы бар?
3. Бұлттық технологиялар қызметін ұсынатын қандай сервистерді білесіңдер?
4. Бұлттық технологиялар ресурстарына тіркеу рәсімі қалай жүргізіледі?

2

Ойланайық, талқылайық

1. Бұлттық технологиялардың мүмкіндіктерін мақсатына қарай қалай пайдалануға болады?
2. Не себепті жеке бұлттар қолданылады?
3. Skype (Lync) қандай мүмкіндіктерді ұсынады?
4. Бұлттық технологияларды пайдаланудың тиімді жақтары қандай?

3

Талдап, салыстырайық

1. Аралас бұлттар мен ортақ бұлттардың айырмашылығына талдау жасаңдар.
2. OneDrive пен Google Docs ресурстарын салыстырыңдар.

4

Дәптерге орындайық

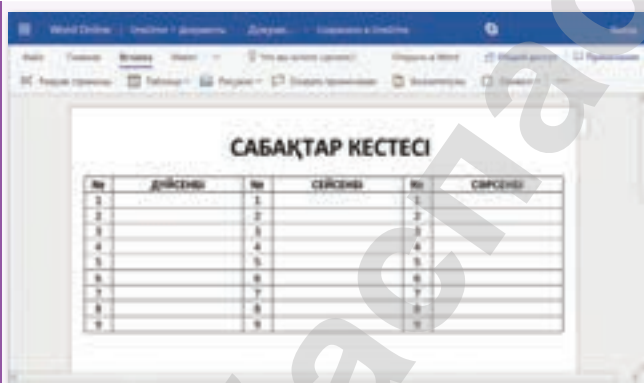
1. Бұлттық технологиялардың негізгі артықшылықтарын жазыңдар.
2. Бұлттық технологиялардың негізгі кемшіліктерін жазыңдар.

5

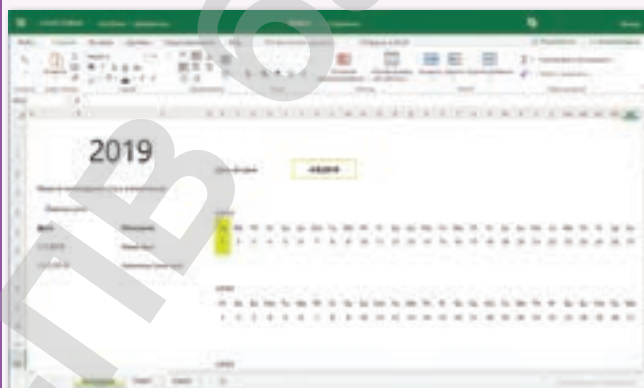
Компьютерде орындайық

Үш топқа бөлініп, берілген тапсырманы OneDrive бұлттық қызметінің жеке бетінде жасаңдар:

1. Сервис қызметіндегі Құжаттар қосымшасын пайдалана отырып, «Сабақтар кестесі» құжатын жасаңдар



2. Сервис қызметтерін пайдалана отырып, «2020 жылғы ҚР мерекелерінің күнтізбесін» құрыңдар



3. Сервис қызметтерін пайдаланып, «Отбасы – адамзат тербеткен ұя» тақырыбына презентация жасаңдар



6

Ой бөлісейік

Сыныптастарыңмен OneDrive пен Google Docs бұлттық ресурстарын (құжат, кесте, презентация, форма, графикалық кескіндер) жасау әдістері туралы пікірлесіңдер. Бұлттық ресурстарды қолдану барысында қайсысы сендерге ең ыңғайлысы болды?

7

Үй тапсырмасы

1. Графикалық бейнелерді пайдалана отырып, салтанатты іс-шараға құттықтау ашықхатын немесе шақыру билетін жасаңдар. Бұлт ресурстарының бірінде (GoogleDocs, OneDrive және т.б.) сыныппен бөлісіңдер.
2. CD дискіден осы тақырыпқа сәйкес тапсырманы орындаңдар.

Шығармашылық тапсырма

«Қазақстанда бұлттық технологиялар қалай дамыған?» сұрағына жауап дайындаңдар.

§ 4. Желілік этикет

Естеріңізге түсіріңдер:

- бұлттық технологиялардың қандай түрлері бар?
- тұтынушыларға қызмет көрсету модельдерін атаңдар;
- бұлттық технологиялардың қандай пайдасы бар?
- бұлттық технологиялардың көлемі қандай болуы мүмкін?

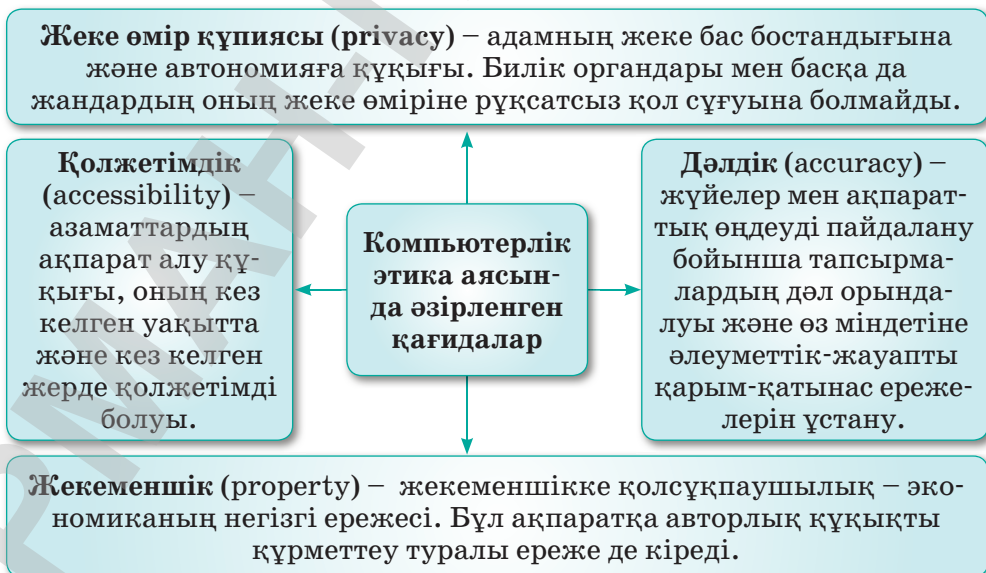
Желілік этикет – Сетевой этикет – Netiquett



Меңгерілетін білім:

- желілік этикет;
- желілік этикет ережесі;
- флейм ұғымы.

Интернет қарыштап дамып келе жатқандықтан, пайдаланушылар саны да өсіп келеді. Желіде қарым-қатынас жасаудан бастап интернет-бизнеске дейін жұмыс істеу барысында, адамдар кейде қателесіп жатады. Мұндай қателер желілік этикетті білмеуден туындайды. Бұл сұхбаттасушыға ғана емес, өзіңе де зиян келтіруі мүмкін. Интернетте қиындықтар тудырмау үшін компьютерлік этика аясында әзірленген қағидаларды сақтау керек (2-сызба):



2-сызба. Компьютерлік этика

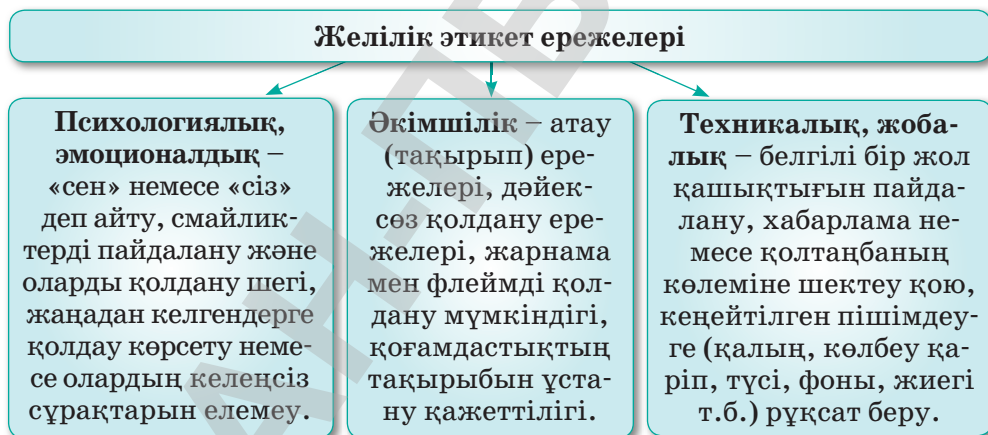
Желілік этикет

Көп жағдайда желілік этикеттің сақталмауы, жеке өмір және меншік құқығының бұзылуы, тақырыптан ауытқу жағдайлары жарнама және өзін-өзі жарнамалау салдарынан болады. Сондай-ақ қате ақпарат, зиянды ақпараттар немесе плагиат этикеттің бұзылуы болып табылады.

Желілік этикеттің ережелері:

1. Адаммен сөйлесіп отырғаныңды ұмытпа.
2. Күнделікті өмірдегі тәртіп ережелерін ұстан.
3. Киберкеңістікте екенінді ұмытпа.
4. Басқалардың уақыты мен мүмкіндігін сыйла.
5. Абыройыңды сақта.
6. Шамаңа қарай басқа адамдарға көмектес.
7. Дау-жанжалдарға араласпа және оларды құптама.
8. Жеке жазба құқығын сыйла.
9. Өз мүмкіндігіңді теріс ниетпен пайдаланба.
10. Басқалардың қатесін кешіруді үйрен.

Жалпы желілік этикеттің ережелерін үш санатқа бөлуге болады (3-сызба).



3-сызба. Желілік этикет ережелері

Өз ережелерін қолданатын бір желі қауымдастығының мүшелері басқалардың ережелерін еріксіз бұзуы мүмкін. Сондықтан барлық Интернет-қауымдастықтарда ортақ ережелермен танысып, олардың сақталуына ресми келісім білдіру керек. Сондай-ақ кез келген даулы мәселелерді шешуге болатын жалпы ережелер бар. Желілік қауымдастықтың белсенді қатысушысы болмас бұрын, оның ережелері туралы біліп алыңдар.

Жаңа өтініш жазып, сұрау жібермес бұрын, тұтынушылардың танымал сұрақтарының тізімін зерттеп алындар.

Флейм – әңгімеге қатысушылардың пікірін ескермей, жиі айтылған эмоционалдық ескертулер. Мұндағы мақсат – тұтынушылардың реакциясын тудыру. Флейм – «дау тудыру». Флейм қолданушылар әңгіменің не себепті басталғанын ұмытып, бір-бірімен қатты ұрыса бастаған жағдайда ең жоғары дәрежеде көрінеді. Әдетте Интернеттегі флейм ресурс модераторлары арқылы жазаланады. Сондықтан қарым-қатынастың жағымсыз салдарынан аулақ болу үшін, Интернетте этикет ережелерін сақтау керек.

1

Сұрақтарға жауап берейік

1. Этикетке адамдар арасындағы қарым-қатынасты реттейтін нормалар мен талаптарды жатқызуға бола ма?
2. Этикет сөзінің «Информатика» пәніне қатысы неде?
3. Интернетте қандай сөздер дерекілікке жатады?
4. Жеке өмір және жекеменшік қағидаларының айырмашылығы неде?

2

Ойланайық, талқылайық

1. Не себепті жоспарланбаған орындарға жарнамаларды жариялауға болмайды?
2. Күнделікті өмірде тұлғааралық қарым-қатынас қалай пайда болады?
3. Не себепті этикеттің бұзылуы мінез-құлыққа әсер етеді?
4. Ақпаратпен жұмыс істеу кезінде неліктен этикалық нормаларды сақтау қажет?

3

Талдап, салыстырайық

Оқушылар екі командаға бөлінеді. Топ басшысы сайланады. Бірінші топ «Шабуылшы», екінші топ «Қорғанушы».

Бірінші топ – желіні жақсы пайдаланушылар. Екінші топ – желінің талаптарына бағынбайтын қолданушылар.

Бірінші топ желіні пайдаланудың кері әсерлері туралы сұрақтарға жауап берсе, екінші топ желінің пайдалы жақтарын айтады. Нәтижелер тақтаға жазылады.

Сұрақтар:

- Интернетте жұмыс істеудің қауіптерін болдырмау және алдын алу құралдары қандай?
- Желіде жұмыс істеу қаупі қандай?

- Желіде жұмыс істеген кезде пайдаланушылардың қандай құқық бұзушылықтары белгілі?
- Қоғам мен мемлекетте желіде құқық бұзушылыққа қарсы қандай шаралар қолданылады?
- Сен жоба авторысың. Оны қорғау үшін қандай әрекеттер жасайтыныңды жаз. Өз құқықтарыңды ата.
- Желілік этикет ережелерін кім көбірек біледі? Қорытынды жасаңдар.

Тапсырма. Интернеттегі авторлық құқық туралы заңдар, жарлықтар, шешімдерді табыңдар.

4

Дәптерге орындайық

1. Нормативтік ережелер неше бөліктен тұрады?
2. Компьютер этикасы ережелерін жазып алыңдар.

5

Компьютерде орындайық

MS Word-та кесте сызып, желідегі қауіпті және қауіпсіз әрекеттерді бөліп жазыңдар.

- Құпиясөзді жасырын ұстаңдар;
- Вирусқа қарсы программаны жаңартыңдар;
- Жеке басқа қатысты ақпаратты құпия ұстаңдар;
- Әлеуметтік желілерде сыпайылықпен әрекет етіңдер;
- Достарыңа хат жазыңдар;
- Ересектердің көмегінсіз арнайы желілерде сауалнамаларды толтырмаңдар;
- Басқа адамдарға әлеуметтік желілерде тіл тигізбеңдер;
- Күдікті хабарламаларды ашыңдар;
- Бейтаныс адаммен онлайн режимінде сөйлесіңдер;
- Күдікті мазмұндағы хабарламаларды ашыңдар;
- Ақылы ойындарға, байқауларға, лотереяға қатысыңдар;

6

Ой бөлісейік

Желіде жұмыс істеу барысында этикалық және құқықтық нормалардың бұзылу салдары туралы салыстырмалы мысалдар келтіріңдер.

7

Үй тапсырмасы

1. Желіні пайдаланушының әрекеттерінің басқа да ережелерін жазыңдар.
2. CD дискіде берілген тапсырманы орындаңдар.

1-БӨЛІМДІ ҚОРЫТЫНДЫЛАУҒА АРНАЛҒАН ТЕСТ ТАПСЫРМАЛАРЫ

1. Қоғамдық бұлт – ...

- A) жеке кәсіпкерлік және олардың өкілдері мен тұтынушылары ғана жұмыс істеуіне арналған жүйе.
- B) қоғамның Интернетті еркін пайдалануына арналған жүйе.
- C) ортақ мақсаттары бар адамдардың бірлесе пайдалануына арналған жүйе.
- D) екі немесе одан да көп бұлт түріндегі аралас комбинацияларды ұсынушы жүйе.
- E) бұлттың бірнеше бөлігінің комбинациясын ұсынушы жүйе.

2. Гибридті бұлт – ...

- A) жеке кәсіпкерлік және олардың өкілдері мен тұтынушылары ғана жұмыс істеуіне арналған жүйе.
- B) қоғамның Интернетті еркін пайдалануына арналған жүйе.
- C) ортақ мақсаттары бар адамдардың бірлесе пайдалануына арналған жүйе.
- D) екі немесе одан да көп бұлт түріндегі аралас комбинацияларды ұсынушы жүйе.
- E) жергілікті және ауқымды желілердің бірнеше бөлігін үйлестіруді ұсынушы жүйе.

3. Мәліметтерді бұлттық сақтау орны, тұтынушы ақпараттарының резервтік көшірмелеріне арналған сервис:

- A) SugarSync.
- B) GoogleDocs.
- C) Box.net.
- D) Dropbox.
- E) OpenDrive.

4. Флейм – ...

- A) желідегі қарым-қатынастың бір түрі.
- B) адамның жеке кеңістігіне рұқсатсыз кіру.
- C) жекеменшікке қол сұқпау.
- D) ақпараттық өңдеу бойынша тапсырмалар.
- E) кез келген уақытта қолжетімді болу.

5. Ақиқаттың, яғни іс-әрекеттің нақты жағдайын көрсететін ақпарат:

- A) түсінікті.
- B) шынайы.

- С) объективті.
- Д) өзекті.
- Е) толық.
- 6. Адам мен компьютер арасындағы қарым-қатынас:**
 - А) интерфейс.
 - В) файл.
 - С) каталог.
 - Д) жарлық.
 - Е) бума.
- 7. «Бұлттық технологиялар» мақсатына қарай ... бөлінеді:**
 - А) жеке, жалпы, қоғамдық, аралас.
 - В) толық, аралас, іріктелген.
 - С) жаңа, қарама-қайшы, таңдамалы.
 - Д) құнды, тұтас, толық, дәл.
 - Е) өзекті, аралас, сенімді, толық.
- 8. Ақпаратты және ақпараттық процестерді зерттейтін ғылым – ...**
 - А) информатика.
 - В) криптоанализ.
 - С) криптография.
 - Д) кибернетика.
 - Е) криптология.
- 9. Адам тілі қандай ақпаратқа жатады?**
 - А) Тактильді.
 - В) Графиктік.
 - С) Мәтіндік.
 - Д) Дыбыстық.
 - Е) Сезімдік.
- 10. Жеке пікірге немесе шешімге байланысты емес ақпаратты қалай атауға болады?**
 - А) Объективті.
 - В) Шынайы.
 - С) Өзекті.
 - Д) Толық.
 - Е) Пайдалы.
- 11. Қандай ақпараттың көмегімен қойылған мәселені шешуге болады?**
 - А) Түсінікті.
 - В) Өзекті.
 - С) Шынайы.

- D) Пайдалы.
- E) Объективті.

12. Dropbox – ...

- A) коммерциялық деректерді сақтау орны.
- B) бірлесе жұмыс істеу мүмкіндігі бар дербес файлдарды сақтау орындары.
- C) мобильді телефон арқылы көруге және өзгертуге болатын сақтау орны.
- D) Google қосымшаларын тегін пайдалануға арналған онлайн-кеңсе.
- E) компанияның web-серверінде орналасқан деректерді сақтау орны.

13. Интернет желісінде тілдесуге қажетті қағидалар:

- A) басқа пайдаланушыларға құрмет көрсету.
- B) өз пікіріне сенімді болу және ісінің дұрыстығын дәлелдеу.
- C) өз аты-жөні бойынша тіркелу.
- D) хабарламаларға жүйелі түрде жауап беру.
- E) үнемі қолжетімді болу.

14. Ақпараттың құндылығы:

- A) нысанның, процестің, құбылыстың және т.б. нақты жағдайына жақындық дәрежесі.
- B) нақты нысанға, процеске, құбылысқа байланысты ақпараттың көмегімен құрылған бейненің сәйкестік деңгейі.
- C) ақпараттың сапасы мен жеткіліктілігін сипаттайды.
- D) пайдаланушыға ақпараттың қолжетімді болуы.
- E) ақпараттың пайдасы немесе пайдасыздығының көрсеткіші.

15. Желілік этикет:

- A) Интернет желісіндегі мінез-құлық ережесі.
- B) қоғамдағы мінез-құлық ережесі.
- C) хабарламаларды жіберу ережесі.
- D) сайттарға кіру ережесі.
- E) Интернетке қосылу ережесі.

2-БӨЛІМ

КОМПЬЮТЕР ТАҢДАЙМЫЗ

Күтілетін нәтижелер:

- мақсатына байланысты компьютердің конфигурациясын таңдау;
- пайдаланушының қажеттілігіне қарай программалық жасақтаманы таңдау;
- процесс модельдерін электрондық кестеде әзірлеу және зерттеу.

§ 5. Компьютердің конфигурациясы

Естеріңе түсіріңдер:

- процессор дегеніміз не?
- компьютердің қызметі қандай?

Меңгерілетін білім:

- компьютердің конфигурациясы;
- мақсатына байланысты компьютердің конфигурациясын таңдау.

Компьютердің конфигурациясы –
Конфигурация
компьютера –
Computer configuration

Ақпараттық және компьютерлік жүйелер саласында *конфигурация* компоненттердің мақсатына, санына және негізгі сипаттамаларына негізделген белгілі бір құрамдас бөлшектердің жиынтығын білдіреді. Компьютердің негізгі конфигурациясы жүйелік блоктан, монитордан, пернетақта мен тінтуірден тұратын компоненттердің ең аз жиынтығын қамтиды. Жүйелік блоктың негізгі конфигурациясына кіретін ішкі құрылғылар – аналық тақша, процессор, жедел жады құрылғысы (ЖЖҚ немесе RAM), қатқыл диск, бейнекарта, қуат көзі. Конфигурация компьютердің жұмысына өсер етеді (4-сызба).



4-сызба. Жүйелік блоктың ішкі құрылғылары

Аналық тақша – барлық негізгі бөлшектер мен тораптар орнатылған компьютердің басты микросызбасы. Аналық тақшаны таңдау үшін алдымен, жүйелік шинаның жиілігіне қарау керек. Сонымен қатар аналық тақшамен жабдықталған түрлі слоттардың саны да маңызды.

Процессор ақпаратты өңдеуге жауапты. Дәл осы жерде мәшине жұмысы кезінде көптеген есептеу операциялары жасалады. Оның маңызды сипаттамасы – тактілік жиілік. Тактілік жиілік неғұрлым жоғары болса, оның жұмыс істеуі соғұрлым жақсы болады. ДК-ға арналған қазіргі заманғы процессорлар 2–6 ядродан тұрады (екі, төрт ядролық және т.б.).

Жедел жадының (ЖЖҚ немесе RAM) қажетті көлемі компьютердің шешетін міндеттеріне байланысты. Көптеген жағдайларда ДК-де ыңғайлы жұмыс істеу үшін 4 Гбайт жадымен қамтамасыз етілуі жеткілікті, бірақ ойындар мен бейнені өңдеу үшін оның көлемін ұлғайту қажет болуы мүмкін. Жады өлшемінен басқа, таңдалған аналық тақшаға байланысты да сипаттамасы ескеріледі.

Қатқыл магниттік дискідегі жинақтауыш (ағылшынша hard (magnetic) disk drive, HDD, HMDD) қатқыл диск, компьютерлік сленгте «винчестер» – магнитті жазба принципіне негізделген есте сақтау құрылғысы. Қатқыл диск – бірнеше өзара байланысты дискілерден тұратын аккумулятор. Дискілердің шеткі жақтары концентрациялық жолдарға, ал жолдар секторларға бөлінеді.

Бейнекарта (графикалық карта, бейнеадаптер) графикалық ақпаратты мониторға шығарады және өңдейді. Қазіргі ойындарда ақпаратты өңдеудің жылдамдығы мен сапасы өте маңызды.

Кейбір маңызды жұмыс үшін, сондай-ақ ойындар үшін жеке жады мен бейнепроцессоры бар қосымша бейнекарта қажет. Оны қолдану сурет сапасын айтарлықтай арттырады және жүйенің жалпы өнімділігін жақсартады.

Қуат көзі. Көптеген конфигурациялар үшін 450 Ватт қуаттылығы жеткілікті. Бірақ сенің компьютеріңде жақсартылған бейне жүйесі бар ойын құрылғысы болса, онда жоғары қуат көзі талап етілуі мүмкін.

Дыбыс картасы – дыбысты өңдеуге мүмкіндік беретін құрылғы (акустикалық жүйелерге шығару немесе дыбыс жазу).

Акустикалық жүйе және микрофон музыканы, фильмдерді, сөйлесуді және үндеткіштер, құлаққаптар т.б. ойнату кезінде компьютерден дыбысты шығару үшін қолданылады.

Дербес компьютердің конфигурациясы негізгі функционалдық мақсатқа байланысты жасалады. Олар кеңсе, үй, дизайнерлік, ойын компьютерлері болып жіктеледі. Бұдан басқа, осы топтағы компьютерлер бастапқы, орта, жоғары деңгейді көрсете отырып, өнімділік деңгейі бойынша да қосымша жіктеледі.

Кеңселік (офистік) дербес компьютерлер, негізінен, кеңселік программалармен және қосымшалармен жұмыс істеуге арналған, әдетте жергілікті желілерге қосылады, жоғары өнімділігі немесе жылдамдығымен ерекшеленбеуі мүмкін. Олардан талап етілетіні – тұрақтылық пен үздіксіз жұмысты жалғастыру.

Үйге арналған дербес компьютерлер үлкен қуат пен өнімділікті қажет етпейтін міндеттерді шешу үшін қолданылады. Мәселен, мәтіндік құжаттармен жұмыс істеу, әлеуметтік желілерде және форумдарда сөйлесу, музыка тыңдау туралы ақпаратты іздеу. Дегенмен үйдегі компьютерлердің конфигурациясы әдетте сапалы мониторды, тиімді процессорды, графикалық картаны, DVD дискісін және акустиканы қамтиды. Интернетке қосылу, қажет болса, теледидарға қосылу қамтамасыз етіледі. Конфигурацияға сканер, принтер, теледидар тюнері, веб-камера және т.б. кіреді.

Ойынға арналған дербес компьютерлер басқа компьютерлерден қуатты графикалық шағын жүйесімен ерекшеленеді. Олардың конфигурациясының негізгі элементтері – қуатты процессор, қуатты Radeon немесе Geforce бейнекартасы және жеткілікті үлкен RAM. Құрамдас бөліктеріне ойынға қажетті қосалқы құралдар (джойстик, руль, педальдар және т.б.) міндетті түрде енгізіледі.

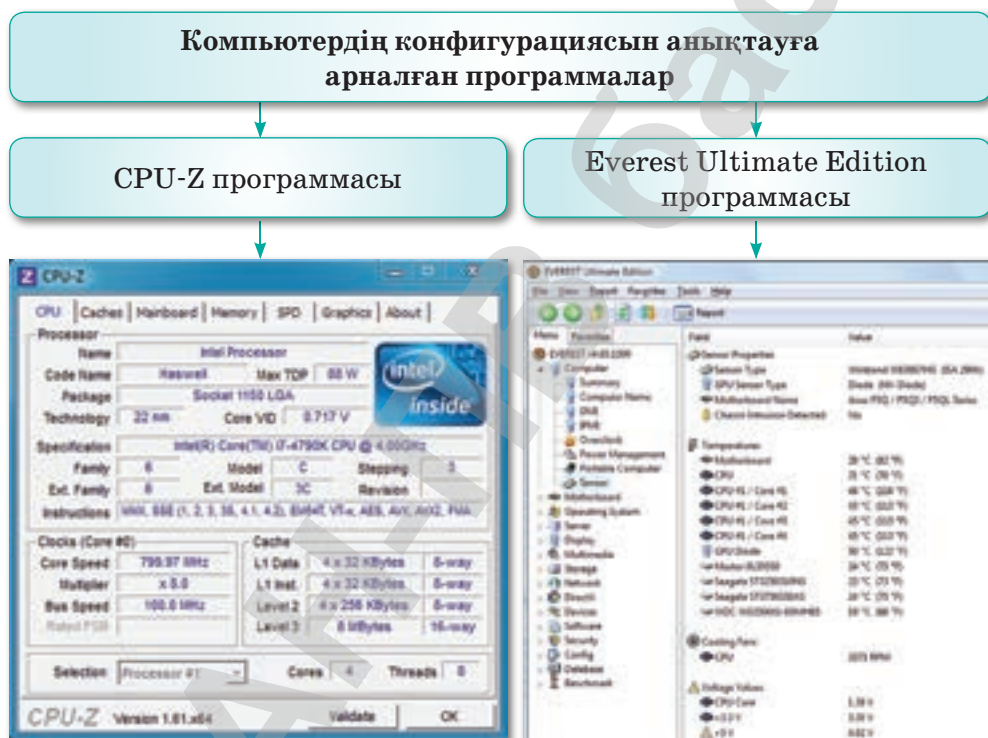
Дизайнерлік дербес компьютерлер немесе графикалық стансылар үлкен және күрделі графикалық қосымшалармен, фотосуреттер және бейнефайлдарды жоғары сапалы өңдеумен жұмыс істеуге арналған. Конфигурация нақты орындалатын тапсырмаларға байланысты өзгеруі мүмкін: 3D графикамен жұмыс істеу қуатты бейнекартасын қажет

етеді, жоғары өнімді процессор бейнефайлдармен жұмыс істейді және т.б.

5-сызбада компьютерлердің конфигурациясын тексеретін қолданыстағы программалар ұсынылған.

CPU-Z (Central Processing Unit – Z) – қатқыл дискіден басқа, орталық процессордың, аналық тақшаның және BIOS, жедел жадының, бейнекартаның техникалық сипаттамасын анықтайтын программа.

EVEREST Ultimate Edition – компьютердің аппараттық және программалық құралдарының жұмысының тиімділігін диагностикалау, тестілеу және баптауға арналған программа.



5-сызба. Компьютердің конфигурациясын анықтауға арналған программалар

1

Сұрақтарға жауап берейік

1. Конфигурация дегеніміз не?
2. Компьютер конфигурациясын қалай таңдауға болады?
3. Дербес компьютерлер не үшін қолданылады?

4. Қандай құрылғылар ДК-нің ең төменгі құрамын қамтамасыз етеді?
5. ДК аналық тақшасының негізгі компоненттері қандай?
6. Дербес компьютерлік шиналардың қызметі қандай?
7. Процессордың өнімділігін қандай параметрлер сипаттайды?

2

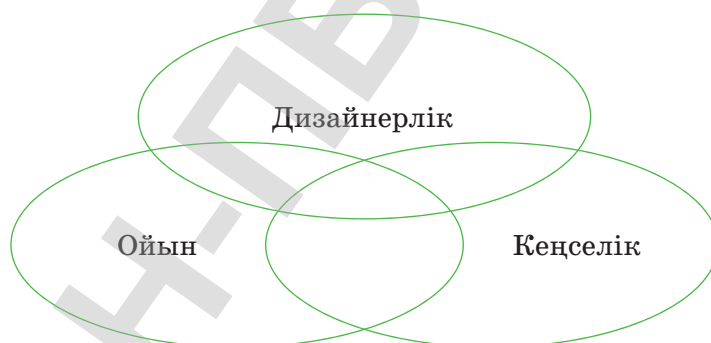
Ойланайық, талқылайық

1. Күнделікті өмірде компьютердің конфигурациясын қалай пайдалануға болады?
2. Компьютер компоненттерінің функционалды мақсаты қандай?
3. Ойын компьютері мен дизайнерлік компьютердің айырмашылықтары неде?

3

Талдап, салыстырайық

1. Венн диаграммасын пайдалану арқылы әртүрлі компьютерді салыстырыңдар.



2. Дербес компьютер мен ноутбук арасындағы айырмашылықты салыстырыңдар. Талдау жасаңдар.

4

Дәптерге орындайық

Қазіргі уақыттағы заманауи компьютерлердің қандай түрлерін білесіңдер? Олардың конфигурацияларын Интернет көмегімен тауып, дәптерлеріңе жазыңдар.

5

Компьютерде орындайық

1-тапсырма.

Белгілі бір міндеттер диапазонын (ойын компьютері, кеңселік компьютер) шешуге арналған компьютер компоненттерін алыңдар. Бұл компьютердің құнын есептеңдер. Осы мәселені шешудің түрлі нұсқаларын таңдау үшін кестені (электрондық кестені) пайдаланыңдар. Барлық компоненттер қосылыс және өткізу қабілеті интерфейсі арқылы аналық тақшамен өзара байланысуы керек.

2-тапсырма.

- а) Интернет, газет және басқа да ақпарат көздерінен өз компьютерлеріңе сәйкес келетін қорек блогы жайында ақпарат іздеп, төмендегі кестені толтырыңдар.

Нарықтық белгісі мен модель нөмірі	Сипаттамасы	Бағасы

- ә) Интернет, газет және басқа да ақпарат көздерінен өз компьютерлеріңе сәйкес келетін процессор туралы деректерді төмендегі кестеге енгізіңдер.

Нарықтық белгісі мен модель нөмірі	Сипаттамасы	Бағасы

6

Ой бөлісейік

Сабақта не білдіңдер? Не үйрендіңдер? Өз ойларыңды достарыңмен бөлісіңдер. Алған жаңа білімдеріңді күнделікті өмірде қандай жағдайда қолдануға болады?

7

Үй тапсырмасы

- Өздерің қолданып жүрген компьютер (ноутбук) конфигурациясының сипаттамасын жазыңдар (Интернеттен қажетті ақпараттарды іздеңдер).
- CD дискіден осы тақырыпқа сәйкес тапсырманы орындаңдар.

§ 6. Практикум. Компьютердің конфигурациясы

Компьютерді функционалдық қызметін жұмыс өнімділігі мен өлшеміне байланысты жіктеуге болады.

Бүгінгі таңда «Cray», «NEC», «Fujitsu» компанияларының компьютерлері қуатты суперкомпьютерлер өндірісіндегі көшбасшылар болып табылады. Олар белгілі бір температура режимі бар арнайы бөлмелерді қажет етеді. Суперкомпьютердің температуралық режимі сумен немесе сұйық азотпен салқындатылады.

1-тапсырма. 2-кестені толтырыңдар.

2-кесте. Құрылғылар

Құрылғылар	Негізгі енгізу құрылғысы	Қосымша шығару құрылғысы	Негізгі шығару құрылғысы	Қосымша енгізу құрылғысы
Монитор				
Тінтуір				
Принтер				
Пернетақта				
Плоттер				
Сканер				
Микрофон				

2-тапсырма. Компьютерді іске қосып, құрылғылардың атқаратын қызметі мен қасиеттерін анықтаңдар (3-кесте).

3-кесте. Құрылғылардың атқаратын қызметі

Құрылғылар	Ішкі сақтау құрылғысы	Сыртқы сақтау құрылғысы
Флеш жады		
Жедел жады		
CD диск		

Құрылғылар	Ішкі сақтау құрылғысы	Сыртқы сақтау құрылғысы
DVD диск		
Қатқыл диск		
Тұрақты жады		

3-тапсырма. «Кубизм» әдісі пайдаланылады.

Қарапайым қағаздан жасалған текшені алып, оның қырларына мына тапсырмаларды жазыңдар.

1. Компьютер конфигурациясын анықтауды *үйреніңдер*.
2. Артур Беркс, Герман Голдстайн, Джон фон Нейманның мақалаларын *сипаттаңдар*.
3. Екілік принцип және жадтың біртектілік принципін *салыстырыңдар*.
4. Ең қолайлы принципті *қолданыңдар*.
5. Дәлелдемелер келтіріп, *түсіндіріңдер*.
6. *Ұсыныңдар*: Болашақта дербес компьютерге тағы қандай қосымша құрылғы қосар едіңдер?

§ 7. Программалық жасақтаманы таңдау

Естеріңізге түсіріңдер:

- конфигурация дегеніміз не?
- компьютердің міндетіне қарай конфигурациясын қалай таңдауға болады?

Меңгерілетін білім:

- программалық жасақтама;
- программалық жасақтаманың түрлері;
- программалық жасақтаманың сапа критерийлері.

Программалық жасақтама –
Программное обеспечение –
Software

Өздеріңіз білетіндей, дербес компьютердегі мәселелерді шешуге арналған программалардың жиынтығы программалық жасақтама деп аталады.

Программалық жасақтама (software) – компьютерді басқаруға арналған программа немесе программалар жиынтығы.

Программаларды құқықтық мәртебесі бойынша үш топқа бөлеміз: **жүйелік** (жалпы пайдалану программалары), **қолданбалы** (арнайы), **инструменталдық** (программалау жүйелері) (*6-сызба*).



6-сызба. Программалық жасақтама түрлері

Жүйелік программалық жасақтама – компьютерлік жүйенің жұмыс істеуін жалпы басқаруды және оның техникалық қызмет көрсету функцияларын орындауды қамтамасыз ететін программалар жиынтығы.

Қолданбалы программалық жасақтама – компьютерді пайдаланушыларға арналған белгілі бір мәселелерді шешуге көмектесетін программалар жиынтығы.

Инструменталдық программалық жасақтама программалау тілінде компьютерге арналған жаңа программаларды құруға ықпал етеді.

Инструменталдық программалардың түрлері: Borland Delphi, Borland C ++ Builder, Microsoft Visual Basic, Microsoft Visual C ++.

Программаларды **құқықтық мәртебесі бойынша** үш топқа бөлеміз: *лицензияланған, шартты тегін және еркін таратылатын.*

- 1. Лицензияланған программалар.** Пайдаланушы лицензиялау туралы ережелерді орындаған жағдайда оған программалық өнімді пайдалану құқығын беретін лицензиямен бірге программалық жасақтама ұсынылады. Әдетте бұл шарттар қолданушының программалық өнімді басқа пайдаланушыға беру, кодты өзгерту мүмкіндігін шектейді.
- 2. Шартты тегін программалар.** Бұл жағдайда тұтынушы белгілі бір сынақ мерзіміне дейін бірнеше мүмкіндігі шектеулі программалық өнімнің демонстрациялық нұсқасын тегін алады, ал ол аяқталғаннан кейін өнімді сатып алуға міндетті.
- 3. Еркін таратылатын программалар.** Көптеген программалық жасақтаманы және компьютерлік техниканы өндірушілер еркін программалық жасақтаманың кең ауқымына қызығушылық танытады.

Қазіргі уақыттағы программалық жасақтаманың сапа критерийлері:

- функционалдық;
- сенімділік;
- тиімділік;
- эргономикалық;
- мобильділік.

Функционалдық – программалық жасақтаманың нақты немесе болжалды пайдаланушы қажеттіліктерін қанағаттандыратын функциялар жиынтығын орындау мүмкіндігі. Белгіленген функциялар жиынтығы программалық жасақтаманың сыртқы сипаттамасында анықталады.

Сенімділік – берілген шарттарда және нақты уақыт кезеңінде белгілі бір функцияларды қалтқысыз орындау мүмкіндігі. Мысалы, программалық жасақтамадан бас тарту қателіктің көрінісі ретінде қабылданады.

Тиімділік – берілген шарттарда пайдаланушыға ұсынылатын программалық жасақтама қызметтерінің деңгейінің және осы үшін пайдаланылатын ресурстар көлемінің арақатынасы. Мұндай ресурстардың қатарына қажетті аппараттық құралдар, программалардың орындалу уақыты, деректерді дайындауға және нәтижелерді түсіндіруге арналған шығындар жатады.

Эргономикалық – бастапқы деректерді дайындау, программалық жасақтаманы қолдану және алынған нәтижелерді бағалау бойынша жұмсалатын күшті азайтуға, сондай-ақ белгілі бір пайдаланушының оң эмоцияларын туғызуға мүмкіндік беретін программалық жасақтаманың сипаттамалары.

Мобильділік – программалық жасақтама қабілеті бір ортадан (қоршаған ортадан) екіншісіне, яғни бір аппараттық платформалардан басқасына ауыстырылу мүмкіндігі.

Серверлік программалық жасақтама – ақпараттық технологияларда белгілі бір ресурстарға немесе қызметтерге қол жеткізе отырып, тапсырыс берушінің сұрауы бойынша сервистік функцияларды орындайтын есептеу жүйесінің программалық компоненті.

Бір компьютерде бірнеше операциялық жүйелерді іске қосу үшін виртуалды мәшинелер жиі қолданылады.

Виртуалды мәшине – компьютердің құрамдас бөліктерінің жұмысы үшін қолданатын программа, нәтижесінде кез келген операциялық жүйені орнатуға және іске қосуға мүмкіндік беретін виртуалды орта құрылады. Бірнеше виртуалды мәшине жасай аламыз, оларға түрлі операциялық жүйелерді орната аламыз, оларды бір уақытта іске қосып, виртуалды жергілікті желі жасауға болады.

1

Сұрақтарға жауап берейік

1. Компьютерлік программалық жасақтама қызметі және оның алатын рөлі қандай?
2. Программалық жасақтаманың қандай түрлерін білесіңдер?
3. Қазіргі уақыттағы программалық жасақтаманың сапасына қандай талаптар қойылады?
4. Программалық жасақтаманың функционалдық мақсаттары қандай?
5. Программалардың құқықтық мәртебесінің қандай түрлері бар?

2

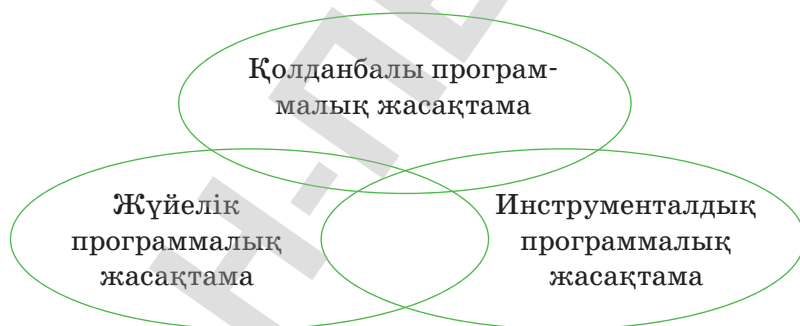
Ойланайық, талқылайық

1. Неге программалық жасақтама қолданушының қажеттілігіне байланысты бөлінеді?
2. Компьютердің операциялық жүйесіне берілетін лицензияның мерзімі қандай?

3

Талдап, салыстырайық

Программалық жасақтама түрлеріне талдау жасаңдар.



4

Дәптерге орындайық

1. Қазіргі уақыттағы программалық жасақтаманың сапа критерийлерін, олардың атқаратын қызметтерін сипаттайтын кесте толтырыңдар.
2. Компаниялар программалық жасақтаманың қандай түрлерін пайдаланушыларға өнімдерін жарнамалау мақсатымен тегін ұсынады?

5

Компьютерде орындайық

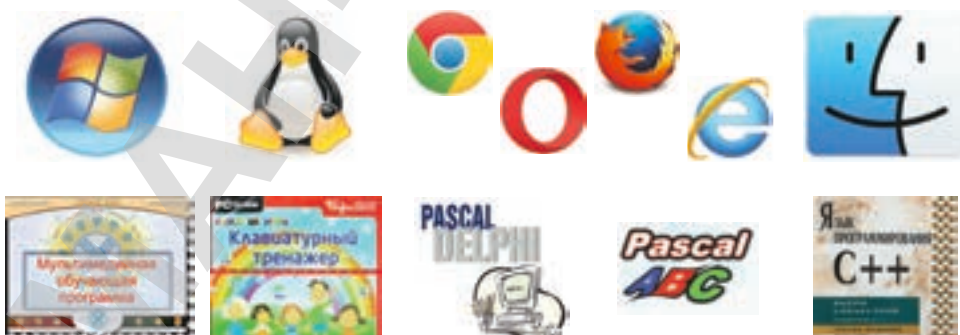
Берілген жағдайларда қандай программалық жасақтама қажет екенін MS Word-та кесте құрып, белгілеңдер:

Жағдай	Жүйелік ПЖ	Қолданбалы ПЖ	Инструменталдық ПЖ
Кәсіби программалаушы ірі фирмдан тапсырыс алып, компьютерлік программаны жазады.			
11-сынып түлектері фотоальбом дайындап, мектептегі өмірлерін еске түсірді.			
Мұғалім өз пәні бойынша компьютерде тест тапсырмасын дайындады.			
Смартфон өндірушілері жаңа телефондардың дизайнын ұсынады.			

6

Ой бөлісейік

1. Сыныптастарыңмен заманауи программалық жасақтаманың сапа критерийлерін талқылаңдар.
2. Программалардың құқықтық мәртебесінің түрлерін сыныптастарыңмен талдаңдар.
3. Берілген суреттер программалық жасақтаманың қандай түріне жататынына талдау жасаңдар.



7

Үй тапсырмасы

Лицензияланған, шартты тегін және еркін таратылатын программаларға мысалдар жазыңдар.

§ 8. Компьютер құнын есептеу

Естеріңізге түсіріңдер:

- компьютерлік программалық жасақтама дегеніміз не? Оның рөлі қандай?
- программалық жасақтаманың қандай түрлері бар?
- программалық жасақтаманың функционалдық мақсаттары қандай?
- программалық жасақтаманың сапа критерийлерін атаңдар.

Менгерілетін білім:

- модель, оның түрлері;
- модельдеу кезеңдері;
- ақпараттық модель.

Физикалық модель –

Физическая модель –

Physical model

Биологиялық модель –

Биологическая модель –

Biological model

Экономикалық модель –

Экономическая модель –

Economical model

Адамзат әрдайым өз қызметінде қоршаған ортаның модельдерін жасаған және пайдаланған.

Модельдер өте күрделі нысандар мен процестерді бейнелеуге мүмкіндік береді. Көрнекі модельдер білім беру процесінде жиі пайдаланылады (глобус, молекулалық үлгілер, кристалдық торлар және т.б.).

Модельдер техникалық жабдықтарды, мәшинелер мен механизмдерді жобалау мен құруда өте маңызды рөл атқарады. Теориялық модельдерсіз (теориялар, заңдар, болжамдар) нақты нысандардың құрылымын, олардың қасиеттері мен сипатын зерттемей, ғылымның дамуын елестету мүмкін емес.

Модель – нысанның, құбылыстың немесе процестің маңызды айырмашылықтарын көрсететін белгілі бір жаңа нұсқа. Модель зерттеушінің модельдейтін нысандарының қасиеттері туралы ақпаратпен қамтамасыз етіледі.

Модельдеу – практикалық мәселелерді шешу үшін модельдерді құру және пайдалану процесі.

Оның негізгі кезеңдері:

1. Есептің қойылуы.
2. Модель құру.
3. Тапсырманың орындалуын тексеру.
4. Нәтижелерді талдау.

Мысал ретінде модельдеудің мына түрлерін қарастырайық:

- экономикалық модель;
- физикалық модель;
- биологиялық модель.

Модельдеу үшін кесте процессоры ортасын тандаймыз. Бұл ортада ақпараттық және математикалық модельдер екі аймақты қамтитын кестеге біріктіріледі:

- бастапқы деректер;
- есептік деректер (нәтижелер).

1. Ғылым ретінде, экономикалық теория өзінің арнайы зерттеу әдістеріне ие. Ең маңызды әдіс – экономикалық модельдерді құру.

Экономикалық модель екі немесе одан да көп айнымалылардан тұрады, мысалы:

- а) айнымалылардың арасындағы қатынасты сипаттайды;
- ә) осы қатынастардың экономикалық нәтижесін көрсетеді;
- б) айнымалы мәндердегі өзгерістердің экономикалық нәтижеге қалай әсер ететінін болжайды.

Экономикалық модельдер күрделі құбылыстардың негізгі сипаттамаларын жинақтап, оларды жеңілдету және талдауды қолжетімді ету үшін пайдаланылады.

Қазіргі заманғы экономист толық математикалық білімге мұқтаж екендігі анық. Бұл экономикалық және математикалық үлгілерді жоспарлау және зерттеу кезінде экономикалық компоненттер арасында байланыс орнату қажет екені расталады. Электрондық кестелер экономистер, бухгалтерлер, инженерлер, тағы басқа мамандар мен ғалымдар үшін ыңғайлы және көптеген сандық деректермен жұмыс жасауға мүмкіндік береді.

Компьютердің құнын есептеу мысалында экономикалық модельді қарастырамыз.

Бүгінгі күні сатуға арналған түрлі компьютерлер (ноутбук, нетбук және т.б.) бар. Сондықтан компьютер сатып алғанда, алдымен қандай мақсатта қолданылатынымызды анықтауымыз керек. Жаңа компьютерді таңдағанда оның ішкі құрылымын, сондай-ақ мониторды, тінтуірді, пернетақтаны және т.б. таңдау қажет. Әрбір құрылғы бөлек таңдалады, тауарлардың сапасы мен бағасын ескеру қажет. Компьютерді таңдау үшін мынадай сипаттамаларды білуіміз керек:

1. Процессор түрі
2. Жедел жады
3. Бейнекарта
4. Аналық тақша
5. Қатқыл диск
6. Операциялық жүйе
7. Салқындатқыш құрылғылар
8. Монитор

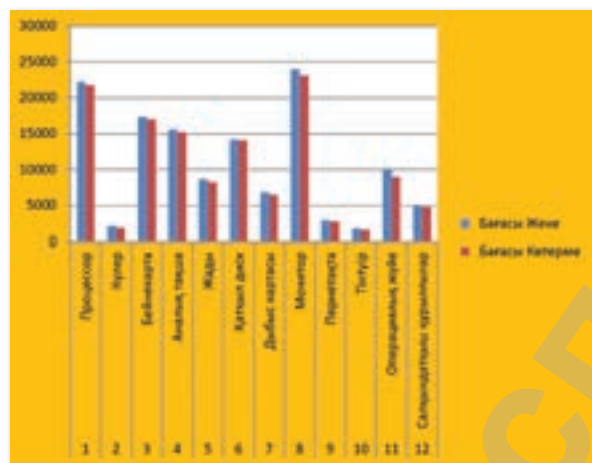
Процессордың түрі мен тактілік жиілігін ескеру қажет. *Мысалы*, егер тактілік жиілігі жоғары болса, онда компьютер өндіруші жұмыс істейді. Сондай-ақ компьютер ядроларының санына назар аудару керек. Бейнекарта жақсырақ болған сайын, суреттер сапасы анық болады. Компьютерді сатып алған кезде ең басты құрылғы – аналық тақшаны таңдау маңызды. Оған компьютердің барлық компоненттері қосылады.

Мысал. Компьютер құрылғыларының тізімі мен бағасы алдын ала беріліп, компьютердің құнын есептеп, өзімізге қажет компьютерге қанша қаражат дайындау керектігін анықтаймыз. Компьютер құнын есептейміз (8-сурет).

Шешім үшін MS Excel көмегімен 4-кестені құрайық:

4-кесте. Компьютердің есептік бағасы

Компьютердің құнын есептеу					
№	Тауар атауы	Бағасы		Кепілдік, жыл	Ескерту
		Жеке	Көтерме		
1	Процессор	22185	21749	1	
2	Кулер	2170	1887	1	
3	Бейнекарта	17330	16975	3	
4	Аналық тақша	15550	15255	3	
5	Жады	8655	8241	1	
6	Қатқыл диск	14175	14045	1	
7	Дыбыс картасы	6825	6501		
8	Монитор	23990	23067		
9	Пернетақта	2926	2787	1	
10	Тінтуір	1800	1600	1	
11	Операциялық жүйе	10000	9000		
12	Салқындатқыш құрылғылар	5046	4806	1	
Барлығы		130652	125913		



8-сурет. Компьютер құнын есептеудегі экономикалық модель

2. Физика-математикалық модельдеу – зерттеудің өте маңызды әдісі.

Мысалы: көкжиекке бұрыш жасай лақтырылған дененің қозғалысының ақпараттық моделін құрастырайық.

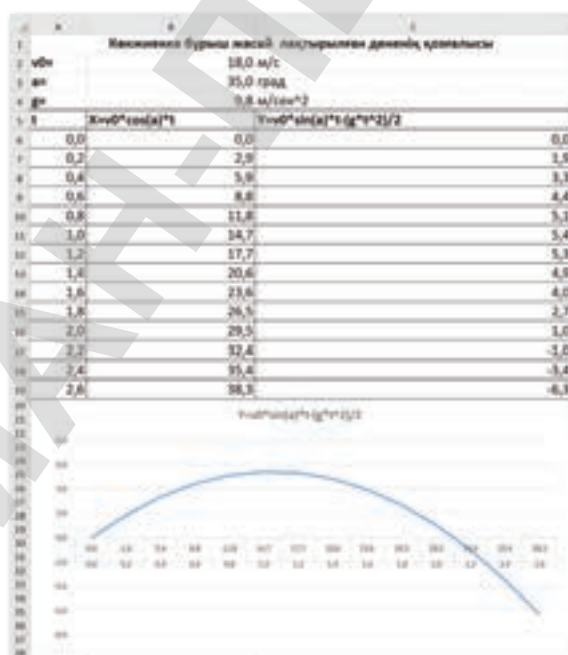
Доп Жерге қарағанда кіші болғандықтан, оны материалдық нүкте деп санауға болады; доптың биіктігінің өзгеруі болмашы ғана, сондықтан еркін түсу үдеуі $g = 9,8 \text{ м/с}^2$ тұрақты мән ретінде қарастырылуы мүмкін және қозғалысты ОУ осі бойынша теңүдемелі деп есептеуге болады.

Денені лақтыру жылдамдығы аз, сондықтан ауа қарсылығын ескермейміз және ОХ осі бойынша қозғалыс бірқалыпты.

Электрондық кестені пайдалану арқылы келесі әрекеттерді орындандар:

- 1) А1-ден С1-ге дейінгі ұяшықтарды біріктіріңдер;
- 2) «Көкжиекке бұрыш жасай лақтырылған дененің қозғалысы» деген мәтінді біріктірілген ұяшыққа жазыңдар;
- 3) мәтін бір жолға сыю үшін, В және С бағандарын ашыңдар;
- 4) А2, А3 және А4 ұяшықтарына, тиісінше, $v_0 =$, $a =$, $g =$ деп енгізіңдер;
- 5) С2, С3 және С4 ұяшықтарында, тиісінше, м/с, град, м/сек² деп енгізіңдер;
- 6) В2, В3 және В4 ұяшықтарына ондық таңбалар санын орнату үшін, пішімді сандық мәнге ауыстырыңдар;
- 7) В2, В3 және В4 ұяшықтарына тиісінше 18; 35; 9,8 мәндерін енгізіңдер;

- 8) А5-ке t , В5-ке $X = v_0 * \cos(\alpha) * t$, С5-ке $Y = v_0 * \sin(\alpha) * t - (g * t^2)/2$ формулаларын сәйкесінше енгізіңдер;
- 9) А6-дан бастап С19 ұяшықтарын таңдап және сан пішімін ондық таңбалар санымен орнатыңдар;
- 10) А6 ұяшығына 0 нөмірін енгізіңдер;
- 11) А6-дан А19-ға дейінгі ұяшықтарды таңдап, оларды 0,2 аралықта толтырыңдар;
- 12) В6 ұяшығына $= \$B\$2 * \cos(\text{РАДИАНЫ}(\$B\$3)) * A6$ формуласын енгізіңдер;
- 13) С6 ұяшығына $= \$B\$2 * \sin(\text{РАДИАНЫ}(\$B\$3)) * A6 - (\$B\$4 * A6^2)/2$ формуласын енгізіңдер;
- 14) формулаларды тиісінше В7: В19 және С7: С19 ұяшықтарына көшіріңдер;
- 15) А5-тен С19-ға дейінгі ұяшықтарды белгілеп, кесте шекараларын орнатыңдар;
- 16) бұл компьютерлік модель көкжиекке бұрышпен тасталған дене қозғалысын зерттеу бойынша қойылған тапсырманы шешуге, бастапқы жылдамдық пен лақтыру бұрышының оңтайлы мәнін таңдауға мүмкіндік береді;
- 17) жұмысты өз бумаларыңа «Физикалық модель» деп сақтаңдар (9-сурет).



9-сурет. Физикалық модель

3. Адам өмірі **биоритм** деп аталатын үш циклдік процестерге ұшырайтындығы туралы болжам бар.

Биоритмдер – тірі ағзадағы биологиялық процестер мен құбылыстардың сипаты мен қарқындылығының жиі (периодты) қайталанатын өзгерістері. Бұл циклдер адам денсаулығының үш жағын сипаттайды: физикалық, эмоционалдық және интеллектуалдық. Биоритмдер біздің көңіл күйіміздің көтерілуін және төмендеуін көрсетеді. Синусоидалық тәуелділікті білдіретін графаның «көтерілуі» неғұрлым қолайлы күндерге сәйкес келеді делінеді. График абсцисса осін қиып өтетін күндер қолайсыз деп саналады. Бұл теория ғылыми түрде дәлелденбеген, бірақ көптеген жандар оған сенеді.

«Үш биоритм» теориясына жүз жылға жуық уақыт болды. Оның авторлары үш адам болған. Эмоционалдық және физикалық биоритмді ашып, бір-біріне тәуелсіз бірдей нәтижелер алғандар – психолог Герман Свобода, отоларинголог Вильгельм Флисс, ал педагог Фридрих Тельчер интеллектуалды ырғақты зерттеді. Тельчердің айтуынша, студенттердің ақпаратты қабылдауы, жүйелеуі және пайдалануы, жаңа идеяларды тудыру қабілеттері құбылмалы. Олардың туған күндерін, емтихан күндерін, нәтижелерін салыстыру арқылы 33 күндік интеллектуалды ритмді анықтады.

Аустриялық психолог Г.Свобода өз пациенттерінің мінез-құлқын талдап, аурулардың пайда болуы мен дамуын зерттеп, жүрек және демікпе ұстамаларының циклдік сипатын байқады. Осы зерттеулердің нәтижесі физикалық (23 күн) және ақыл-ой (28 күн) процестерінің ырғағын ашты.

Неміс докторы В.Флисс ті адам ағзасының ауруларға қарсы тұруы қызықтырды. Неліктен бірдей диагнозы бар балалардың бірінің иммунитеті жақсы, ал екіншісі үнемі ауыра береді? Ғалым аурудың басталуы, температурасы және емдеу нәтижелері туралы деректерді жинап, оларды науқастың туған күнімен байланыстырды. Есептеулер иммунитеттің өзгеруін 23 күндік физикалық және 28 күндік эмоционалдық биоритмдердің көмегімен болжауға болатындығын көрсетті.

Барлық үш биоритм адамның туған күнінде басталады. Адам үшін туған сәт өте күрделі, өйткені осы күні барлық үш биоритм абсцисса осін қиып өтеді. Биология тұрғысынан бұл шындыққа жанасады, себебі дүниеге келген бала суда тіршілік

ету ортасынан ауаға өтеді. Сондықтан сәби ағзасында орасан зор өзгерістер болады. Физикалық биоритм адамның өмірлік күштерін, яғни оның физикалық әл-ауқатын сипаттайды. Оның жиілігі – 23 күн.

Эмоционалдық биоритм адамның ішкі көңіл күйін, қоршаған ортаны эмоциялық жағынан сезіну қабілетін білдіреді. Бұл цикл кезеңінің ұзақтығы – 28 күн.

Үшінші биоритм ойлау қабілетін, адамның интеллектуалдық күйін сипаттайды. Оның айналымы – 33 күн.

Мысалы, модельді одан әрі талдау мақсатында нақты бір адамға биоритмді алдын ала ағымдағы белгілі бір күннен (санақ басталған күн) бір ай мерзімге модельдеу ұсынылады.

Математикалық модель. Математикалық модельді былайша ұсынады:

Физикалық биоритм: $f(x)=\sin(2*\pi*x/23)$;

Эмоционалдық биоритм: $f(x)=\sin(2*\pi*x/28)$;

Интеллектуалдық биоритм: $f(x)=\sin(2*\pi*x/33)$,

мұнда x – күндермен санағандағы адамның жасы.

Кестені үлгі бойынша толтырыңдар (*10-сурет*):

	А	В
1		
2	Биоритмдер	
3	Бастапқы деректер	
4	Туған күні	06.03.2004
5	Анықталған мерзімі	01.04.2019

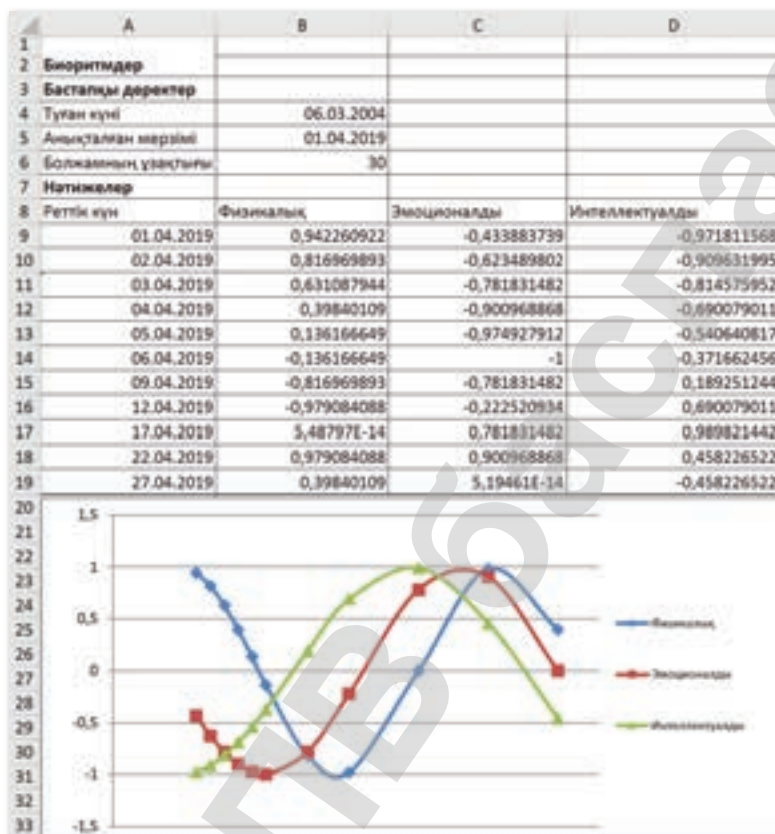
10-сурет. Ақпараттық модель

Модельді үлгі бойынша құрастырыңдар. Есептелген формуладағы бастапқы деректерді ұяшықтарға енгізіңдер (*11-сурет*).

Ұяшық	Формула	
A9	=B\$5	(1)
A10	=A9+1	(2)
B9	=SIN(2*ПИ()*(A9-B\$4)/23)	(3)
C9	=SIN(2*ПИ()*(A9-B\$4)/28)	(4)
D9	=SIN(2*ПИ()*(A9-B\$4)/33)	(5)

11-сурет. Бастапқы деректер

Биоритмнің компьютерлік моделі (12-сурет):



12-сурет. Биологиялық модель

Жұмысты «Биологиялық модель» деп сақтаңдар.

1

Сұрақтарға жауап берейік

1. Физикалық, биологиялық және экономикалық модель жасай аласың ба?
2. Кез келген физикалық есепті электрондық кестеде шығаруға бола ма?
3. Биоритм диаграммалары туралы не айта аласыңдар?
4. Экономикалық есептердің қандай түрлерін білесіңдер?

2

Ойланайық, талқылайық

1. Модельдер не үшін қажет?
2. Биологиялық модель қандай мақсатпен жасалады?

3. Физикалық модель мен биологиялық модельдің айырмашылығы неде?
4. Экономикалық модельдеудің мақсаты неде?

3

Талдап, салыстырайық

Модельдерді құру әдістері туралы айтыңдар. Экономикалық модельді құру әдісіне талдау жасаңдар.

4

Дәптерге орындайық

1. Физикалық модельдер теориясы туралы есептер дайындаңдар.
2. Физикалық, биологиялық, экономикалық модельдеудің тиімді жақтарын жазыңдар.

5

Компьютерде орындайық

1. Егер екі адам бір, екі немесе тіпті барлық үш биоритмі бірдей немесе өте ұқсас кестелер алса, онда біз бұл адамдардан жоғары сәйкестікті көре аламыз. Екі достың физикалық, эмоционалдық және интеллектуалдық сәйкестік моделін құрастырыңдар.
2. 15 қызметкерге сыйақы бөлу керек. Әрбір қызметкерге есептелген соманың белгілі бір проценті төленеді. 150 000 тг сыйақы қорын қанағаттандыру үшін сыйақының қандай процентін белгілеу керектігін анықтаңдар.

6

Ой бөлісейік

Қалай ойлайсыңдар, нысандардың бірнеше моделі болуы мүмкін бе? Адам өмірінің түрлі салаларындағы модельдеу мысалдарын келтіріңдер.

7

Үй тапсырмасы

1. «Электрондық кестеде процестердің физикалық моделін құрудың маңыздылығы» деген тақырыпқа презентация дайындаңдар.
2. CD дискіден осы тақырыпқа сәйкес тапсырманы орындаңдар.

2-БӨЛІМДІ ҚОРЫТЫНДЫЛАУҒА АРНАЛҒАН ТЕСТ ТАПСЫРМАЛАРЫ

1. Жүйелік блок құрамына кірмейтін құрылғы:
А) қатқыл диск.
В) пернетақта.
С) процессор.
D) кэш-жады.
Е) жедел жады.
2. Винчестер – ...
А) ақпаратты шығару құрылғысы.
В) ақпаратты өңдеу және сақтау құрылғысы.
С) ақпаратты сақтау құрылғысы.
D) ақпаратты енгізу құрылғысы.
Е) ақпаратты өңдеу құрылғысы.
3. Ақпаратты шығару құрылғыларына жатпайтын құрал:
А) принтер.
В) дыбыс колонкалары.
С) дисплей.
D) плоттер.
Е) сканер.
4. Ақпаратты өңдейтін компьютердің негізгі құрылғылары:
А) процессор, жады, принтер, HDD, монитор.
В) чипсет пен жедел жады.
С) микропроцессор мен құрылғылар контроллері.
D) орталық процессор мен жедел жады.
Е) орталық процессор мен чипсет.
5. Процессордың негізгі сипаттамалары:
А) модель, тактілік жиілік.
В) модель, тактілік жиілік, адрестік шинаның разрядтылығы.
С) модель, тактілік жиілік, разрядтылығы.
D) тактілік жиілік, разрядтылығы, жедел жадының көлемі.
Е) модель, тактілік жиілік, сыртқы құрылғыларға айналу жылдамдығы.

6. Компьютердің жедел жадын не деп атайды?
 - A) CD-ROM.
 - B) RAM (Random Access Memory).
 - C) ROM (Read Only Memory).
 - D) Hardware.
 - E) Software.
7. Компьютердең «миы» деп қандай құрылғыны айтамыз?
 - A) Принтер.
 - B) Диск-жетек.
 - C) Процессор.
 - D) Пернетақта.
 - E) Модем.
8. Кэш-жады қай құрылғы құрамында орналасқан?
 - A) Жедел жады.
 - B) Дискілік жады.
 - C) Тұрақты есте сақтау жадысы.
 - D) Процессор.
 - E) Сканер.
9. Компьютерді өшіргенде ... барлық ақпарат жойылады.
 - A) қатқыл дискідегі.
 - B) иілгіш дискідегі.
 - C) тұрақты жадыдағы.
 - D) CD-ROM дискідегі.
 - E) жедел жадыдағы.
10. Жедел жады ... арналған.
 - A) өзгертілмейтін ақпаратты сақтауға.
 - B) ақпаратты ұзақ уақыт сақтауға.
 - C) ақпаратты уақытша сақтауға.
 - D) ақпаратты жоюға.
 - E) өзгертілмейтін ақпаратты ұзақ уақыт сақтауға.

11. Программалық жасақтаманың құрамдас негізгі бөлігі:

- A) программалау тілдері мен компьютерлік вирустар.
- B) инструменталды құрылғылар мен жүйелік ПЖ.
- C) программалық компилятор және интерпретатор.
- D) драйвер және операциялық жүйе.
- E) қолданбалы ПЖ, жүйелік ПЖ және инструменталдық ПЖ.

12. Компьютердің өнімділігі ... байланысты.

- A) жүйелік блоктың құрамдас бөліктеріне.
- B) орнатылған бағдарламалық жасақтамаға.
- C) Интернетке қосылу жылдамдығына.
- D) компьютердің құнына.
- E) өндіріс уақытына.

13. Ақпараттың көлемін өлшеудің ең аз өлшем бірлігі:

- A) байт.
- B) килобит.
- C) бит.
- D) нөлдік.
- E) бірлік.

14. Компьютердің аппараттық және программалық құралдарының оңтайлы жұмысын диагностикалау, тестілеу және талдауға арналған программа:

- A) CPU-Z.
- B) Виртуалды мәшине.
- C) EVEREST Ultimate Edition.
- D) Сервистік программалық жасақтама.
- E) CPU-R.

15. Дербес компьютердің компоненттерінің техникалық сипаттамаларын анықтайтын программа:

- A) CPU-Z.
- B) Виртуалды мәшине.
- C) EVEREST Ultimate Edition.
- D) Сервистік программалық жасақтама.
- E) CPU-R.

3-БӨЛІМ

ДЕРЕКТЕР БАЗАСЫ

Күтілетін нәтижелер:

- деректер базасы, жазба, өріс терминдерін түсіндіру;
- электрондық кестеде деректер базасын жасау;
- деректерді іздеу, сұрыптау және сүзгілеуді жүзеге асыру.

§ 9. Деректер базасы: негізгі түсініктер мен анықтамалар

Естеріңе түсіріңдер:

- ақпараттық нысандарға не жатады?
- кестелік процессорларда қандай ақпарат сақталады?
- кестелік процессордың түрлері.

Деректер базасы –
Базы данных –
Database

Меңгерілетін білім:

- деректер базасы; ДББЖ түсінігі;
- деректер базасының қолданылуы;
- деректер базасының функциялары.

Күнделікті өмірде адамдар қоршаған орта туралы ақпарат ағынын үнемі бақылап отырады. Бұл ағыннан мәлімет алу тиімділігін арттыру үшін деректер базасын басқару жүйелерін (ДББЖ) пайдалану өзекті мәселе болып табылады.

Деректер базасын басқару жүйесі (ДББЖ) – деректер базасымен жұмыс істеуге арналған программалық жасақтама. Деректер базасын құру, қолдау және жұмыс істеу үшін пайдаланылатын программалық және тілдік құралдар жиынтығы.

ДББЖ-мен еркін жұмыс істеу үшін деректер базасына қатысты кейбір тұжырымдамалар туралы нақты түсінік алу қажет.

ДББЖ көптеген мүмкіндіктерге ие (*7-сызба*).

Жаңа ДБ құру

Деректерді сақтау
және өңдеу

Деректерді енгізу
кезеңінде басқару

Сұраныстар мен
есептерді дайындау

Деректерді
іздеу

Деректермен ың-
ғайлы жұмыс істеу
үшін форма құру

Бірнеше пайдалану-
шының бір мезгілде
жұмыс істеуі

Деректерді
жазу

7-сызба. ДББЖ мүмкіндіктері

Деректер базасы (ДБ) – компьютер арқылы тез арада іздеу және керек құжатты табу мүмкіндігін арттыру үшін ұйымдас-тырылған деректер мен ақпараттар жиынтығы. *Мысалы*, «Мек-теп» деректер базасы мұғалімдер, оқушылар, ата-аналар тура-лы барлық деректерді: әрқайсысының тегі, аты, телефон нөмірі және т.б. сақтайды. «Кітапхана» деректер базасында кітаптар туралы барлық деректер бар: кітап атауы, авторы, жариялан-ған жылы және т.б. деректер.

Деректер базасы құрылымына байланысты: *реляциялық, иерархиялық және желілік* деп бөлінеді.

Реляциялық ДБ – нақты қатынастармен байланысты кесте немесе бірнеше кесте, кесте жолдары *жазбалар* және бағандар деректер базасының *өрістері*. Реляциялық деректер базасында кестелер реттелген және байланысқан деректер жиынтығы болып табылады.

Иерархиялық ДБ – алдыңғы элементтердің тәуелділігін есепке ала отырып құрылған база. Онда жоғарғы деңгейдегі элементтердің (тораптардың) әрқайсысы төменгі деңгейдегі бір немесе бірнеше тораппен байланысты болуы мүмкін.

Желілік ДБ – өзінің құрылымы бойынша аса күрделі және тораптар арасындағы еркін байланыстары бар деректер базасының графикалық моделі. Деректер базасының иерар-хиялық және желілік модельдерінің негізгі ұғымдарына: *эле-мент (торап), деңгей, байланыс* жатады.

Деректер базасы Excel, Oracle, SQL, DataBase және т.б. программаларда жасалады. Бұлар – деректер базасын-да деректерді өңдеу және сақтауды автоматтандыруға қабілетті өте қуатты және қымбат программалар. Дегенмен пайдаланушылардың басым бөлігі Excel-де қарапайым де-ректер базасын жасауды қалайды. Себебі **Microsoft Excel** құралдар жиынтығымен кез келген пайдаланушы кестелік процессордағы деректерді енгізу, іздеу және өңдеу жұмысын қауіпсіз бастай алады.

Қызықты ақпарат

Исландия XVIII ғасырдан бастап, барлық исландиялықтардың туыс-тықтары туралы ақпаратты қамтитын кешенді генеалогиялық деректер ба-засын **Íslendingabók** құрды. Исландияның әрбір азаматы осы деректер ба-засына қол жеткізе алады, сондай-ақ белгілі тұлғалармен өз қарым-қатынас деңгейін тексере алады.

1

Сұрақтарға жауап берейік

1. Деректер базасын күнделікті өмірде қолданамыз ба?
2. Деректер базасын басқару жүйелерінің мақсаты қандай?
3. Деректер базасының негізгі модельдері мен функцияларын атаңдар.
4. Деректер базасын басқару жүйесінің мүмкіндіктері туралы айтыңдар.
5. Деректер базасының құрылымын қысқаша сипаттаңдар.

2

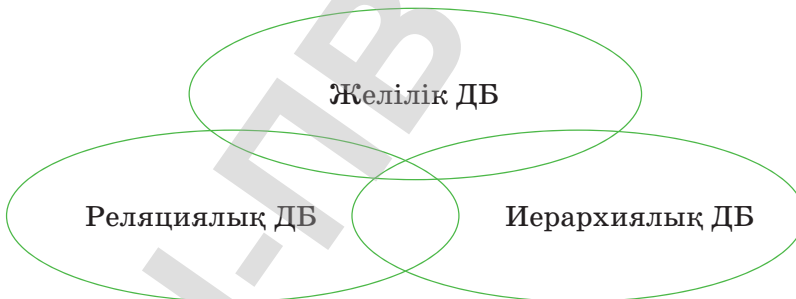
Ойланайық, талқылайық

1. Не себепті деректер базасын құру қажет?
2. Неге көптеген пайдаланушылар Excel-де деректер базасын құруды таңдайды?

3

Талдап, салыстырайық

Венн диаграммасын пайдалану арқылы реляциялық және иерархиялық және желілік деректер базасын салыстырыңдар.



4

Дәптерге орындайық

1. Өмірден деректер базасының әртүрлі үлгілерін келтіріңдер (реляциялық, желілік және т.б.).
2. ДБ нысандарына байланысты сөзжұмбақ немесе басқатырғыш құрастырыңдар: кесте, форма, сұраныстар, есептер және т.б.

5

Компьютерде орындайық

Деректер базасын құру үшін өздеріңе қызықты тақырыпта деректерді дайындаңдар. MS Excel программасында болашақ ДБ құрылымын құрастырыңдар (кесте қандай

бағандар мен жолдардан тұрады, оған не негіз бола алады, қандай деректер жиі қолданылады? және т.б.).

6

Ой бөлісейік

Сыныптастарыңмен деректер базасы қай жерде жиі пайдаланылатынын талқылаңдар. Деректер базасы күнделікті өмірді қаншалықты жеңілдетеді?

7

Үй тапсырмасы

1. Интернеттен әртүрлі мамандықтардың классификациялық деректер базасын табыңдар.
2. CD дискіден осы тақырыпқа сәйкес тапсырманы орындаңдар.

Шығармашылық тапсырма

Интернеттен материалдарды пайдалана отырып, демонстрациялық слайд дайындаңдар: «Әртүрлі деректер базасына (реляциялық, желілік және т.б.) арналған артықшылықтар және оларды қолдану салалары».

§ 10. Кесте. Өріс және жазба туралы түсінік

Естеріңізге түсіріңдер:

- деректер базасы дегеніміз не?
- деректер базасы қандай мақсатта жасалады?
- деректер базасының негізгі функциялары қандай?

Меңгерілетін білім:

- жазба, өріс, кілт;
- деректер базасында деректердің типтері;
- деректерге қолданылатын операциялар;
- MS Excel программасының жаңа мүмкіндіктері.

Өріс – Поле – *Field*

Жазба – Запись – *Record*

Реляциялық деректер базасы

Кестелік деректер базасы екіөлшемді кесте түрінде ұсынылуы мүмкін: оның әрбір жолында жазбалар реттелген түрде орналастырылады және әрбір бағанда нысанның немесе құбылыстың сипатының көрсетілген өріс атауына сәйкес мәні жазылады.

Деректер базасының негізгі компоненттері – кесте, өріс, жазба.

ДБ өрісі – нысанның қасиетінің мәндерін қамтитын кесте бағаны.

ДБ жазбасы – деректер қорының өрісінде орналасқан, белгілі қасиеті жазылған кестенің қатары.

Кілттік өріс – кестедегі әрбір жазбаны бірегей түрде анықтайтын өріс.

Кестенің әрбір өрісі нысанның бір сипаттамасын және көрсетілген деректердің типін (мәтін, сандық немесе күн/уақыт) қамтиды.

ДБ деректер типі:

- **мәтіндік** – қарапайым мәтін жолы.
- **сандық** – кез келген түрдегі нақты, бүтін сан және т.б.
- **күні / уақыты** – күні және / немесе уақыты.

«Мектеп» ДБ кестесіндегі жазба мен өрістер байланысына мысал келтіреміз (5-кесте):

5-кесте. «Мектеп» ДБ

Жазба	Кілттік өріс	Өріс 1	Өріс 2	Өріс 3	Өріс 4
	Реттік нөмірі	Тегі, аты-жөні	Туған кезі	Телефон нөмірі	Ата-анасы
Жазба 1	1	Абылай Венера Берікқызы	22.03.2004	8***	Абзалов Б.А.
Жазба 2	2	Жапаров Қайрат Абайұлы	02.12.2004	8***	Көкенова А.А.
Жазба 3	3	Раева Жанна Жакейқызы	13.04.2003	8***	Маликов Ж.Р.

5-кестеде «Мектеп» ДБ кестесінде өрістер мен жазбалар арасындағы байланыс көрсетілген. Мұнда **қою қаріппен** Оқушылардың аты-жөні, ал **курсивтік қаріппен** Туған кезі, **Телефон нөмірі**, **Ата-анасы** туралы мәліметтер берілген. Оқушыға қатысты (Реттік нөмірі, Тегі, аты-жөні, Туған кезі, Телефон нөмірі, Ата-анасы) кілттік өріс **Реттік нөмірі** болады.

Кестенің әр жолында белгілі бір нысан туралы деректер бар (мысалы, компьютер, тауар), ал кесте бағанында сол нысан туралы сипаттама (мысалы, сауалнамалық деректер, процессордың сериясы мен маркасы, жеткізушілердің немесе клиенттердің байланыстары т.б.) берілген.

1

Сұрақтарға жауап берейік

1. Реляциялық ДБ өмірде қандай салаларда қолданылады?
2. Өріс, жазба, кілттік өріс деген не?
3. MS Excel деректер типін қалай тізімдейміз?
4. ДБ-да деректер типі деген не?

2

Ойланайық, талқылайық

Бұл мысалда өріс, жазба деген не? Деректердің қандай типтері өрістерде мәндерді қабылдайды?

№	Планета	Орташа температурасы, °C	Ауданы, мың км ²
1	Жер	14,8	510 072
2	Марс	-63,1	144 370

№	Планета	Орташа температурасы, °C	Ауданы, мың км ²
3	Меркурий	67	74 800
4	Шолпан	463	460 000

3

Талдап, салыстырайық

ДБ-ның жаңа терминдері: өріс, жазба, кілт, деректер түрлері және т.б. қолданып, сөзжұмбақ немесе басқатырғыш жасаңдар. Терминдерді талдап, салыстырыңдар.

4

Дәптерге орындайық

Жақшада ассоциативті мысалдарды жалғастырыңдар.
Деректер базасында жиі пайдаланатын өрістер типі:

- Мәтін (тегі, аты, ...);
- Сандық (температура, шығын, ...);
- Күн/уақыт (күн / ай / жыл, ...).

5

Компьютерде орындайық

Берілген кестедегі деректерді MS Excel-ге енгізіп, «Менің достарым» атты деректер базасын құрыңдар.

№	Тегі	Аты	Әкесінің аты	Туған кезі	Қаласы
1	Абылай	Венера	Берікқызы	22.03.2004	Нұр-Сұлтан
2	Жапаров	Қайрат	Абайұлы	02.12.2004	Алматы
3	Раева	Жанна	Жакейқызы	13.04.2003	Алматы

6

Ой бөлісейік

Сыныптастарыңмен талқылаңдар, сабақта қандай жаңа ұғымдармен таныстыңдар? MS Excel-де деректердің қандай типтері бар?

7

Үй тапсырмасы

Хабарлама дайындаңдар: «MS Excel-дің түрлі нұсқаларындағы деректер типі».

Шығармашылық тапсырма

«MS Excel-де деректер базасын құрудың артықшылықтары мен мүмкіндіктері» тақырыбына презентация дайындаңдар.

§ 11. Деректер базасы. Электрондық кестеде деректер базасын құру

Естеріңізге түсіріңдер!

- реляциялық деректер базасы дегеніміз не?
- өріс түсінігі, жазба, кілт деген не?
- MS Excel-дегі деректер типі қандай?

Меңгерілетін білім:

- электрондық кестедегі деректер базасы;
- MS Excel жұмысының принциптері;
- диапазон, тізім түсінігі.

Электрондық кесте –
Электронная таблица –
Spreadsheet

Бұл тақырыпта ДБ-мен MS Excel-де жұмыс жасау бойынша практикалық дағдылар қалыптастырамыз. Программаны іске қосу командасы: **Іске қосу (Пуск) ⇒ Программалар (Программы) ⇒ Microsoft Excel.**

MS Excel программасында **Файл ⇒ Құру (Создать)** командасын орындағаннан кейін, біз «Күнделік» атты файлды құрып, оны электрондық тасымалдаушы құралдарға немесе «Менің деректер базам» атты бумаға сақтаңдар.

ДБ құрудың тәртібі:

1. Екі кестенің құрылымын төмендегідей қойыңдар:

1) Тегі, аты, әкесінің аты, мекенжайы, телефон нөмірі, қызығатын ісі (оқушылар). Алынған кесте – «Оқушылар» (6-кесте).

6-кесте. «Оқушылар» ДБ

№	Тегі	Аты	Әкесінің аты	Мекенжайы	Телефон нөмірі	Қызығатын ісі
1	Қасымов	Ерлан	Болатұлы	Абай, 50	8***	Футбол
2	Абылай	Венера	Берік-қызы	Әйтеке би, 1	8***	Суда жүзу
3	Жапаров	Қайрат	Абайұлы	Әуезов, 15	8***	Программалау
4	Раева	Жанна	Жакей-қызы	Қазыбек би, 5	8***	Фотоға түсіру
5	Дархан	Айя	Серікқызы	Төле би, 38	8***	Билеу
6						

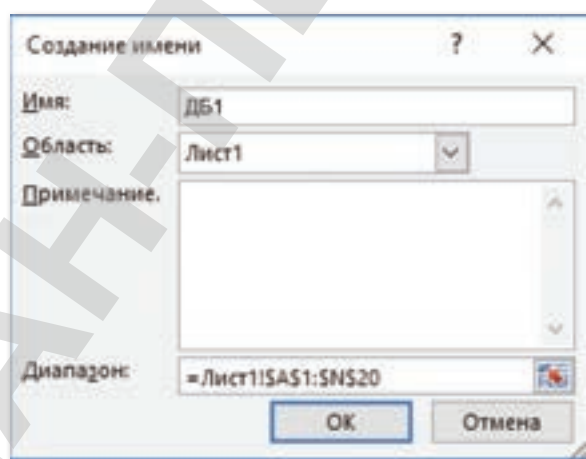
- 2) Тегі, аты, әкесінің аты, мекенжайы, телефон нөмірі, пәннің аты (мұғалім), алынған кесте – «Мұғалімдер» (7-кесте).

7-кесте. «Мұғалімдер» ДБ

№	Тегі	Аты	Әкесінің аты	Мекенжайы	Телефоны	Пәннің аты
1	Әбенова	Лаура	Исақызы	Төле би, 70	8***	Информатика
2	Вагнер	Вера	Петровна	Гагарин, 4	8***	Орыс тілі
3						

- 3) Кестедегі өрістерді сыныптастарың мен мұғалімдердің деректерімен толтырыңдар. *Мысалы*, жеке парақтарда, Бет 1, Бет 2.
2. Деректер базасы диапазонының атауын тағайындаңдар. Ол үшін A2 ... G20 баған тақырыптары бар деректер аймағын таңдаңдар; негізгі мәзірде **Формула** ⇒ **Атауын тағайындау** (Присвоить имя) командасын орындаңдар. Пайда болған **Атауын құру** (Создание имени) терезесінде атын жазып (мысалы, ДБ1), **ОК** батырмасын басыңдар – диапазонға атау беріледі (13-сурет).

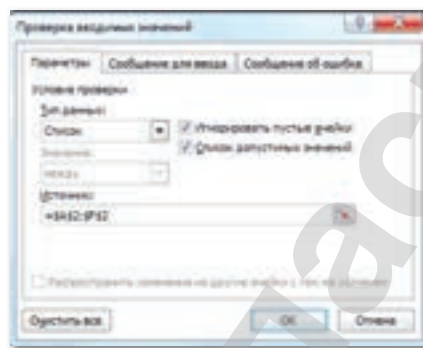
Екінші кесте үшін де дәл осылай жасаңдар.



13-сурет. Деректер базасы диапазонының атауын тағайындау

3. Деректер базасының диапазонын тізім ретінде жариялау. Енгізілетін мәндерді тексеру үшін, берілген ұяшықты немесе бағанды белгілеп, **Деректер** (Данные) ⇒ **Деректерді тексеру** (Проверка данных) командасын басыңдар.

Пайда болған **Енгізілетін мәндерді тексеру** (Проверка вводимых значений) терезесінде **Баптаулар** (Параметры) командасының **Деректер типі** (Тип данных) өрісінен **Тізім** (Список) тармағын таңдаңдар. **Дерек көзін** (Источник) басып, тізім диапазонын таңдаңдар. ОҚ батырмасын басамыз – тізім жасалады (14-сурет).



14-сурет. Тізімді құру тармағы

4. MS Excel-де деректер базасы жасалуы аяқталды.

Маңызды мәлімет

Әрбір бағанда бір ғана деректер типі, мәтін, сандар немесе күндер жазылады.

1

Сұрақтарға жауап берейік

1. Деректер базасын құрудың жұмыс принципін сипаттаңдар.
2. Диапазон деген не? Тізім дегеніміз не?
3. Жеке ұяшықтарды пішімдеу мүмкін бе?
4. Түрлі типтегі деректер бір бағанда сақталуы мүмкін бе?

2

Ойланайық, талқылайық

Кілттік өрісті таңдаңдар және кестені толықтырыңдар.

Реттік нөмірі	Автор	Шығарманың атауы	Бағасы	Басылған жылы	Баспа
1	М.О.Әуезов	Абай жолы	3500	2008	Жазушы

- Реттік нөмірі
- Автор
- Шығарманың атауы

- Бағасы
- Басылған жылы
- Баспа

3

Талдап, салыстырайық

MS Excel программасында деректер базасын жасау қадамдарын талдаңдар.

4

Дәптерге орындайық

Кесте өрістерінде деректер типін анықтап, дәптерлеріңе жазыңдар.

Реттік номері	Автор	Шығарманың атауы	Бағасы	Басылған жылы	Баспа
1	М.О.Әуезов	Абай жолы	3500	2008	Жазушы
2	Л.Н.Толстой	Соғыс пен бейбітшілік	1800	2004	Азбука
3	Э.М.Ремарк	Үш дос	2300	2017	АСТ
4	Ш.Т.Айтматов	Қызыл алма	1000	2005	Атамұра

5

Компьютерде орындайық

Сабақта оқылған материалды бекіту үшін «Күнделік» деректер базасын қажетті ақпаратпен толықтырыңдар:

Реттік номері	Тегі	Аты	Әкесінің аты	Туған кезі	Қаласы
1	Қасымов	Ерлан	Болатұлы	13.11.2004	Нұр-Сұлтан
2	Абылай	Венера	Берікқызы	22.03.2004	Нұр-Сұлтан
3	Жапаров	Қайрат	Абайұлы	02.12.2004	Алматы
4	Раева	Жанна	Жакейқызы	13.04.2003	Алматы
5	Дархан	Айя	Серікқызы	23.03.2004	Нұр-Сұлтан

6

Ой бөлісейік

Сабақта жаңадан не үйрендіңдер? Пайдаланушылар арасында деректер базасын құруда MS Excel-ді пайдалану қаншалықты тиімді деп ойлайсыңдар?

7

Үй тапсырмасы

MS Excel программасында өздерің оқыған кітаптарыңның деректер базасының үлгісін жасаңдар.

§ 12. Ақпаратты іздеу әдістері

Естеріңе түсіріңдер:

- деректер базасы қалай құрылады?
- диапазон, тізім дегеніміз не?
- ДБ құруға қойылатын талаптар қандай?

Меңгерілетін білім:

- ақпаратты ұсынудың алуан түрлері;
- деректер базасында ақпаратты іздеу әдістері;
- зерттелген материалды практикада қолдану.

Форма – Форма – *Form*

Іздеу – Поиск – *Search*

Деректер базасында ақпарат ұсыну

Ақпараттың едәуір мөлшерімен жұмыс істегенде, қандай программалық жасақтама қолданылғанына қарамастан, мынадай әрекеттер орындалады:

1. Деректер базасының кестелері құрылады.
2. Кестелердегі деректерді енгізу үшін формалар жасалады.
3. Деректер базасы кестелеріне қажетті сұраныстар жасалады.
4. Баспаға басу үшін сұраныс негізінде есептер жазылады.

ДББЖ қолдану арқылы деректер базасын қалыптастыру ДБ құрылымын анықтаудан басталады: деректер базасының өрістерін, олардың типтерін анықтау және атауларын енгізу. Әрі қарай кестеге тікелей енгізу әдісі бойынша немесе форма арқылы деректер базасы жазбаларын енгізу, қарау, жөндеу жүргізіледі. Содан кейін деректер базасында деректерді сұрыптауға және іздеуге болады.

Форманы пайдаланып деректер базасының жазбаларын ұсыну

Форма – деректер базасына жазбаларды енгізу, қарау және өңдеуге арналған нысан. Формалар кестеден бір жазбаны көрсетеді және онда бір жазбадан екіншісіне жылжыту батырмалары орналасады.

Форманың міндеті – ашылмалы терезе түрінде, пайдаланушыға қарапайым, түсінікті және ыңғайлы интерфейспен

қамтамасыз ету, ол программаға деректерді тиімді енгізуге және программаның нәтижелерін диалог режимінде оқуға мүмкіндік береді.

MS Excel программасында жазбаларды кесте режимінде де, форма режимінде де көруге болады. Форма жасау барысында оның терезесіне деректер базасының қандай өрістерін қосуға болатынын, сондай-ақ өрістердің орналасқан жерін көрсетуге болады.

Форманы шақыру үшін негізгі кестедегі кез келген ұяшықты белсенді етіңдер (тінтуірмен шертіндер). Программаның бас мәзірінде орналасқан **Форма** белгішесін басыңдар. Пайда болған **ДБ1** терезесінен (Excel жұмыс кітабындағы парақтың аты бойынша) **Қосу** (Добавить) батырмасын басыңдар. Терезені жаңа жазбаның мәліметтерімен толтырамыз. Терезеден терезеге пернетақтадағы **Tab** пернесін басу арқылы жылжыңдар.

15-суреттегідей деректер базасы жазбасы бар форма пайда болады:

	A	B	C	D	E	F	G
2	№	Тегі	Аты	Әкесінің аты	Мекенжайы	Телефон нөмірі	Қызығатын ісі
3	1	Қасымов	Ерлан	Болатұлы	Абай, 50	8***	Футбол
4	2	Абылай	Венера	Берікқызы	Әйтеке би, 1	8***	Суда жүзу
5	3	Жапаров	Қайрат	Абайұлы	Әуезов, 15	8***	Программалау
6	4	Раева	Жанна	Жакейқызы	Қазыбек би, 5	8***	Фотоға түсіру
7	5	Дархан	Айя	Серікқызы	Төле би, 38	8***	Билеу
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							

ДБ1 ? X

№:

Тегі:

Аты:

Әкесінің аты:

Мекенжайы:

Телефон нөмірі:

Қызығатын ісі:

1 из 2

Добавить

Удалить

Вернуть

Назад

Далее

Критерии

Заккрыть

15-сурет. Деректер жазбаларын форма ретінде көрсету

Деректер базасында форманы пайдаланудың басты артықшылығы – бір уақытта экранда берілген жазбадағы барлық өрістерді көру мүмкіндігі.

Сүзгілерді пайдаланып ақпарат іздеу

Деректер базасының негізгі мақсатының бірі – қажетті ақпаратты іздеу. Ақпаратты іздеу сүзгілер арқылы жүзеге асырылады.

Деректер базасы кестелерінде *сүзгі* деп аталатын белгілі бір шарттарды қанағаттандыратын жазбаларды іздей аламыз.

Сүзгілер – белгілі бір шартқа немесе шарттар жиынына сәйкес келетін жазбалардың таңдауы. Жазбаларды іздеу шарттары салыстыру операторының көмегімен құрылады: сандық деректер үшін (=, >, <, >=, <=, <>) және мәтіндік деректер үшін (құрамында жоқ, басталады ..., басталмайды ..., аяқталады ..., аяқталмайды ...).

Сүзгілер мынадай түрлерге бөлінеді:

- 1) қарапайым (тек бір өріске арналған жазбаларды таңдау шарттары);
- 2) күрделі (өртүрлі өрістердің бірнеше шарты).

Күрделі сүзгіні қолдану нәтижесінде бір мезгілде барлық шартты қанағаттандыратын жазбалар көрсетіледі.

MS Excel программасындағы сүзгілер: бір кестедегі жазбалар бойынша деректерді іздеу. Олар бөлек нысан ретінде сақталмайды және оларда есептелетін өрістер жасалмайды.

Функцияларды пайдаланатын ақпаратты іздеу

MS Excel программасында іздеу функцияларын қолданып, қажетті деректерді тез және тиімді табуға болады. Функциялардың комбинациясын іздеу және көрсету проблемаларын шешу үшін ВПР, ИНДЕКС және ПОИСКПОЗ функциялары қолданылады.

Кестедегі жолдар немесе диапазон бойынша деректерді табу қажет болғанда, іздеу функцияларының бірі ВПР-ды қолданамыз. Мысалы, автокөлік бөлшектерінің бағасын оның нөмірімен таба аламыз.

ВПР-дың ең қарапайым функциясы мынадай:

= **ВПР** (іздеу мәні; мәнді іздестіру диапазоны; қайтару мәні мен диапазондағы бағандардың саны; дәл немесе шамамен сәйкестігі 0 / FALSE немесе 1 / TRUE ретінде көрсетілген).

ПОИСКПОЗ функциясы ұяшықтар диапазонындағы элементтерді іздейді және диапазондағы сол элементтің салыстырмалы орнын қайтарады. Мысалы, егер A1: A3 диапазоны 5, 25 және 38 мәндерін қамтыса, онда = **ПОИСКПОЗ (25; A1: A3; 0)** формуласы 2 мәнін қайтарады, себебі 25-элемент екінші диапозонда тұр.

1

Сұрақтарға жауап берейік

1. MS Excel программасында форма не үшін қолданылады?
2. Сүзгі деген не?
3. Мәтіндік және сандық сүзгілер қалай қолданылады?
4. Деректер базасында жұмыс істеу кезінде сүзгілер қалай көмектеседі?
5. Іздеу функцияларын қашан қолданған дұрыс?

2

Ойланайық, талқылайық

Деректерді іздеу және сүзгілеу жұмысымен байланысты MS Excel программалық құралдарының қысқаша сипаттамасын беріңдер. «Күнделік» деректер базасында сүзгілерді пайдалану бойынша әрекеттердің ретін талқылаңдар.

3

Талдап, салыстырайық

Оқушылар топқа бөлінеді. Әр топқа үлестірмелі қағаздарда жазылған жаңа терминдер таратылады: формалар, сүзгілер, диапазондар, тізімдер және т.б. Содан кейін топтардың арасында ақпарат алмасуға уақыт бөліну керек. Топтарға арналған тапсырма шарты: ақиқат немесе жалған ақпарат болатындай, жаңа терминдерге сөйлем құрастырып, оларды талдап, салыстырады.

4

Дәптерге орындайық

Интернеттен әртүрлі деректер базасы негізінде ВПР, ИНДЕКС, ПОИСКПОЗ іздеу функцияларын қолдану арқылы бірқатар мысалдар дайындаңдар.

5

Компьютерде орындайық

«Әлемдегі қазақтардың саны» деректер базасы кестесі берілген. Осы кестеге іздеуді қолданып, «Бұл сан 1 миллион

адамнан асады» деген шартты қанағаттандыратын жазбалардың санын анықтаңдар.

№	Елдер	Саны	Жылы
1	Қазақстан Республикасы	12 000 000	2017
2	Қытай Халық Республикасы	1 462 588	2010
3	Ресей Федерациясы	647 732	2010
4	Өзбекстан Республикасы	803 000	2017
5	Қырғыз Республикасы	34 122	2015

6

Ой бөлісейік

Бүгінгі сабақта не білдіңдер? Деректер базасында ақпаратты іздеу әдістері жұмысты қаншалықты жеңілдетеді? Практикада қолдану мысалдарын келтіріңдер.

7

Үй тапсырмасы

1. Қолданыстағы «Күнделік» деректер базасын дайындаңдар. «Оқушылар» кестесінде «Мекенжай» жолағында іздеу функцияларын қолдана отырып, Алматы қаласындағы барлық оқушыларды сақтаңдар. Алынған кестені электрондық тасымалдауышта немесе «Бұлтта» сақтаңдар.
2. CD дискіден осы тақырыпқа сәйкес тапсырманы орындаңдар.

Шығармашылық тапсырма

Ақпаратты іздеу үшін оқыған әдістерді қолданудың MS Excel деректер базасында жеке моделін дайындаңдар.

§ 13. Деректерді сұрыптау және сүзгілеу

Естеріңізге түсіріңдер:

- деректердің ақпараттық жүйелері;
- ДБ-да ақпараттың берілуі;
- ДБ-да ақпаратты іздеу тәсілдері.

Меңгерілетін білім:

- сұрыптау және сүзгілеу жолдары;
- зерттелген материалды практикада қолдану.

Сұрыптау – Сортировка –
Sorting

Сүзгі – Фильтр – Filter

Деректерді сұрыптау. MS Excel бізге деректерді сұрыптауға мүмкіндік береді, яғни деректерді белгілі бір тәртіпте орналастыра аламыз (өсуі немесе кему ретімен). Мысалы, өсу тәртібінде сұрыптағанда: 1) сандар ең кішісінен ең үлкеніне дейін; 2) мәтін алфавиттік тәртіппен; 3) күн және уақыт хронологиялық тәртіппен жүреді. Тиісінше кему тәртібінде керісінше орындалады.

Деректерді сүзгілеу. Белгілі бір шарттар мен салыстыру операторларына сәйкес келетін жазбаларды таңдағымыз келсе, сүзгілеуді пайдаланамыз. Мысалы, көрсетілген жазбалардың санын азайту үшін, мәні таңдау шарты ретінде алынатын өрісті тінтуірдің оң жақ батырмасымен шертіп, контекстік мәзірдің төменгі жағында **Тең** (Равно), **Тең емес** (Не равно), **Қабылдау немесе Қабылдамау** (Содержит или Не содержит) параметрін орнатамыз.

Сүзгілеудің Іздеуден айырмашылығы – тек берілген шартқа сәйкес шектеулі жазбаларды көрсетуінде. Сүзгілерді қосу немесе өшіру арқылы тез арада кез келген деректерді сүзгіленгеннен сүзгіленбеген түрлеріне ауыстыруға болады.

Басты (Главная) қосымшасының **Сұрыптау және сүзгі** (Сортировка и фильтр) топтары опцияларынан **Сүзгілеу** (Фильтровать) батырмасын басып, қажетті сүзгіні таңдаңдар. Осыған байланысты сұрыптау **Өсу** (По возрастанию) немесе **Кему** (По убыванию) бағыттары бойынша орындалады.

Ақпаратпен жұмыс істеудің практикалық мысалдары

Деректер базасындағы деректерді сұрыптау және сүзгілеу мысалдарын қарастырамыз.

1. Деректер базасында 8-кесте төмендегідей берілген:

8-кесте. «Қысқы олимпиада-2018» ДБ

Орын	Мемлекет	Алтын	Күміс	Қола	Барлығы
1	Норвегия	14	14	11	39
2	Германия	14	10	7	31
3	Канада	11	8	10	29
4	АҚШ	9	8	6	23
5	Нидерланд	8	6	6	20
6	Швеция	7	6	1	14
7	Оңтүстік Корея	5	8	4	17
8	Швейцария	5	6	4	15
9	Франция	5	4	6	15
10	Аустрия	5	3	6	14

Теориялық материалды іс жүзінде бекіту үшін мына тапсырмаларды орындаңдар:

- 1) «Алтын» өрісіндегі жазбаларды өсу ретімен сұрыптау;
 - 2) шартты қанағаттандыратын жазбаларды таңдау «Күміс ≤ 5 »;
 - 3) шартқа сәйкес келетін жазбаларды таңдау: «Қола ≥ 10 ».
2. Деректер базасында 9-кесте төмендегідей берілген:

9-кесте. «Оқушылар» ДБ

№	Тегі	Аты	Әкесінің аты	Туған кезі	Қаласы
1	Абылай	Венера	Берікқызы	22.03.2004	Нұр-Сұлтан
2	Жапаров	Қайрат	Абайұлы	02.12.2004	Алматы
3	Раева	Жанна	Жакейқызы	13.04.2003	Алматы
4	Дархан	Айя	Серікқызы	23.03.2004	Нұр-Сұлтан

- 1) «Оқушылар» жазбасынан «Нұр-Сұлтан» қаласы бар жазбаларды табыңдар;
- 2) тегі «А»-дан басталатын жазбаларды табыңдар.

3. «Ғалам» деректер базасының кестесі мынадай түрде берілген (10-кесте):

10-кесте. «Ғалам» ДБ

№	Планета	Орташа температура, °С	Аудан, мың км ²
1	Жер	14,8	510 072
2	Марс	-63,1	144 370
3	Меркурий	67	74 800
4	Шолпан	463	460 000

- 1) *орташа температура* өрісіндегі жазбаларды Кему реті (По убыванию) бойынша сұрыптаңдар;
- 2) «Ауданы 500 млн км² артық» шартына сәйкес келетін жазбалардың санын көрсету.

Барлық тапсырмаларды орындау нәтижесінде тек іздестіру немесе сүзгілеу шарттары көрсетілетін жазбалар ғана табылуы және белгіленуі керек.

1

Сұрақтарға жауап берейік

1. Деректерді сұрыптау қандай қадамдардан тұрады?
2. Кеңейтілген сүзгілеуді пайдалану реті қандай?

2

Ойланайық, талқылайық

1. Деректер базасында сұрыптау және сүзгілеу не үшін қолданылады?
2. Деректер базасымен жұмыс істеу кезінде MS Excel сұрыптау және сүзгілеу әдістеріне қалай көмектеседі?

3

Талдап, салыстырайық

Деректер базасындағы сұрыптау және сүзгілеуден өткізу түрлеріне, олардың қызметтеріне талдау жасаңдар.

4

Дәптерге орындайық

Жаңа терминдерді пайдаланып, шағын кроссворд жасаңдар: сүзгі, сұрыптау, деректерді іздеу және т.б.

5

Компьютерде орындайық

«Әлемдегі қазақтардың сандық бейнесі» кестесіне сүзгілеуді пайдалана отырып, анықтаңдар:

№	Мемлекеттер	Саны	Жылы
1	Қазақстан Республикасы	12 000 000	2017
2	Қытай Халық Республикасы	1 462 588	2010
3	Ресей Федерациясы	647 732	2010
4	Өзбекстан Республикасы	803 000	2017
5	Қырғыз Республикасы	34 122	2015

Осы кестедегі «саны 1 миллионнан асатын» шартын қанағаттандыратын жазбалардың саны: 1) 1; 2) 2; 3) 3; 4) 4.

6

Ой бөлісейік

Бұл сабақта не үйрендіңдер? Сыныптастарыңмен деректер базасымен жұмыс істеу кезінде деректерді сұрыптау және сүзгілеу әдістері қаншалықты жеңілдетілгенін талқылаңдар.

7

Үй тапсырмасы

Қолданыстағы «Күнделік» деректер базасын дайындаңдар. «Оқушылар» кестесін алфавит бойынша өсу/ кему ретімен сұрыптаңдар. Сүзгілеуді пайдаланып, «Аты-жөні» өрісіндегі барлық қыздарды / ұлдарды таңдаңдар. Алынған кестені сыртқы тасымалдауышта немесе өз бумаларыңда сақтаңдар.

Шығармашылық тапсырма

MS Excel программасында деректер базасының жеке моделіне сүзгілеу әдістерін қолданыңдар.

§ 14. Практикум. Деректер базасымен жұмыс. «Пәндік олимпиада» деректер базасын құру

Деректер базасын жасау кезеңдері:

1) Міндетті айқындау.

Деректер базасының құрылымы мен құрамы, оны құрудың мақсаты, деректер базасында орындалатын жұмыстардың түрлері (деректерді енгізу, өңдеу, деректерді шығару немесе есеп беру және т.б.).

2) Нысанды талдау.

Деректер базасында жасалатын нысандар мен олардың қасиеттерін анықтау. Деректер базасының деректер типтерін зерттеу.

3) Модельді жинақтау.

Деректер базасы моделін таңдау. Кестелер арасындағы байланыстары бар деректер базасы моделінің сызбалық көрінісі.

4) Ақпаратты ұсыну тәсілдері.

- форманы пайдалану;
- формасыз.

5) Нысан моделін синтездеу және оны құру технологиясы.

1-қадам. Microsoft Excel программасын іске қосу.

2-қадам. Бастапқы кестені немесе кестелерді жасау.

3-қадам. Форманы жасау.

4-қадам. Ақпаратты деректер базасымен толтыру.

6) Жасалған деректер базасымен жұмыс істеу.

- қажетті деректерді іздеу;
- баспаға жіберу;
- деректерді сұрыптау;
- деректерді өңдеу және өзгерту.
- деректерді сүзгілеу;

Орындалатын жұмыстың тәртібі:

ДБ-мен жұмыс істеу үшін бөлек бума жасаңдар.

Жұмыс дискісінде STUDY каталогын құрыңдар. STUDY каталогында жеке каталог жасаңдар, мысалы: MYWORK.

«Пәндік олимпиада» деректер базасын құру.

1) Microsoft Excel программасын іске қосыңдар.

2) «Пәндік олимпиада» ДБ-дағы Қатысушылар кестесін жасаңдар және толтырыңдар.

3) *Реттік нөмірі, Аты-жөні, Пән аты, Қаласы* өрістерін құрыңдар.

4) *Реттік нөмірі/№* негізгі өріс ретінде құрылады.

5) Жасалған кестені «Қатысушылар» ретінде сақтаңдар.

6) Кестеге мына жазбаларды толтырыңдар:

11-кесте. «Қатысушылар» ДБ

Реттік нөмірі	Аты-жөні	Пән аты	Қаласы
1	Иса Назерке Қайратқызы	Қазақстан тарихы	Нұр-Сұлтан
2	Ахметова Анар Қыдырханқызы	Математика	Алматы
3	Қараш Айым Әуезқызы	Физика	Алматы
4	Айтбай Елжас Ерболатұлы	Шет тілі	Қызылорда
5	Жақсылық Динара Ардаққызы	Қазақ тілі мен әдебиеті	Павлодар
6	Еркін Зилола Еркінқызы	Орыс тілі мен әдебиеті	Шымкент
7	Ерболатова Ақтоты Шаханқызы	Химия	Орал
8	Қайрат Аслан Темірханұлы	Биология	Шымкент
9	Батырова Ажар Ерболқызы	Математика	Тараз

- 7) **Аты-жөні** өрісін алфавит реті бойынша **сұрыптаңдар**.
 8) Ол үшін кестені белгілеп, Саймандар тақтасындағы **Деректер** (Данные) қосымшасынан **Сұрыптау** (Сортировка) батырмасын шертіндер. Содан кейін **Сұрыптау** терезесінен қажетті **Аты-жөні** бағанын және сұрыптау тәртібін таңдаңдар.
 9) Формадағы кестені мынадай жазбалармен толықтырыңдар:

12-кесте. Толықтырулар

Реттік нөмірі	Аты-жөні	Пән аты	Қаласы
10	Созақбаева Құралай Маратқызы	Қазақстан тарихы	Қызылорда
11	Марат Аяулым Жақсылыққызы	Химия	Ақтөбе
12	Омаров Алмас Сабыржанұлы	Физика	Нұр-Сұлтан
13	Абдуллаева Адия Рахатқызы	География	Қарағанды

- 10) Форманы пайдаланып А.Е. Батырованың пәнін «Қазақ тілі мен әдебиетіне» өзгертiңдер.
- 11) Кестеге тағы бiр **Оқу тiлi** өрiсiн қосыңдар және тиiстi деректердi енгiзiңдер.
- 12) **Қатысушылар** формасын жасаңдар.
- 13) Осы нысандар бойынша мәлiметтердi берiңдер.
- 14) «Пәндiк олимпиада» деректер базасын сақтаңдар.
- 15) Нәтижесiнде ДБ-дағы кесте мынадай көрiнiс табады:

13-кесте. «Пәндiк олимпиада» ДБ

Реттік нөмірі	Аты-жөні	Пән аты	Оқу тілі	Қаласы
1	Иса Назерке Қайратқызы	Қазақстан тарихы	қазақ	Нұр-Сұлтан
2	Ахметова Анар Қыдырханқызы	Математика	қазақ	Алматы
3	Қараш Айым Әуезқызы	Физика	қазақ	Алматы
4	Айтбай Елжас Ерболатұлы	Шет тілі	орыс	Қызылорда
5	Жақсылық Динара Ардаққызы	Қазақ тілі мен әдебиеті	қазақ	Павлодар
6	Еркін Зилола Еркінқызы	Орыс тілі мен әдебиеті	орыс	Шымкент
7	Ерболатова Ақтоты Шаханқызы	Химия	қазақ	Орал
8	Қайрат Аслан Темірханұлы	Биология	орыс	Шымкент
9	Батырова Ажар Ерболқызы	Қазақ тілі мен әдебиеті	қазақ	Тараз
10	Созақбаева Құралай Маратқызы	Қазақстан тарихы	қазақ	Қызылорда
11	Марат Аяулым Жақсылыққызы	Химия	қазақ	Ақтөбе
12	Омаров Алмас Сабыржанұлы	Физика	қазақ	Нұр-Сұлтан
13	Абдуллаева Адия Рахатқызы	География	орыс	Қарағанды

§ 15. Практикум. Деректер базасында іздеу және сүзгілеу бойынша жұмыс

Тармақ 1. «Пәндік олимпиада» деректер базасын MYWORK бумасынан ашыңдар.

Тармақ 2. «Пәндік олимпиада» деректер базасында деректерді іздеу және сүзгілеу.

- 1) Microsoft Excel программасын іске қосыңдар.
- 2) **Қатысушылар** кестесіне жаңа **Нәтижелер** өрісін қосыңдар.
- 3) Мына үлгідегідей жазбалармен толтырыңдар:

14-кесте. Жаңа өрісті қосу

Реттік нөмірі	Аты-жөні	Нәтижелер (балл)
1	Иса Назерке Қайратқызы	78
2	Ахметова Анар Қыдырханқызы	118
3	Қараш Айым Өуезқызы	109
4	Айтбай Елжас Ерболатұлы	89
5	Жақсылық Динара Ардаққызы	114
6	Еркін Зилола Еркінқызы	121
7	Ерболатова Ақтоты Шаханқызы	96
8	Қайрат Ақерке Темірханқызы	84
9	Батырова Ажар Ерболқызы	69
10	Созақбаева Құралай Маратқызы	101
11	Марат Аяулым Жақсылыққызы	92
12	Омаров Алмас Сабыржанұлы	77
13	Абдуллаева Адия Рахатқызы	105

- 4) **Диаграмма** қосымшасында **Аты-жөні** және **Нәтижелер** өрістерін пайдаланып, гистограмма құрыңдар. Гистограмманың типін және түсін өзгертіндер.
- 5) **Пән** және **Нәтижелер** өрістері үшін 4 пунктті қайта жасаңдар.
- 6) 90-нан жоғары балды іздеу үшін сүзгіні пайдаланыңдар.
- 7) Қатысушылар жазбасын жеке-жеке таңдап алыңдар:
 - а) қазақ тілінде оқыту; ө) орыс тілінде оқытуды таңдау.
- 8) **Физика** пәні бойынша барлық қатысушыларды іздеңдер (... басталады (Начинается с ...) сүзгісін пайдалану).
- 9) Жұмыстың нәтижелерін ұсыныңдар. ДБ-сын сақтандар.

3-БӨЛІМДІ ҚОРЫТЫНДЫЛАУҒА АРНАЛҒАН ТЕСТ ТАПСЫРМАЛАРЫ

1. Белгілі бір пән аймағы бойынша реттелген деректер жинағы:

- A) электрондық кесте.
- B) нақты тізім.
- C) деректер базасы.
- D) бөлінген деректер.
- E) иерархиялық жүйе.

2. Excel программасы – ...

- A) мәтіндік редактор.
- B) электрондық кесте.
- C) деректер базасын басқару жүйесі.
- D) кестелік редактор.
- E) қосымшаларды жобалау программасы.

3. Сұрыптау – ...

- A) массивтің ең үлкен және ең кіші элементтерін іздеу.
- B) кейбір жиындардың үздіксіз реттеу.
- C) кейбір жиындардың элементтерін ауыстыру.
- D) кейбір жиындардың сызықтық реттелуі.
- E) берілген шартты қанағаттандыратын жиын элементтерін таңдау.

4. Реляциялық деректер базасында жазба дегеніміз ...

- A) кесте ұяшығы.
- B) кесте жолы.
- C) кестенің бағаны.
- D) өріс атауы.
- E) кесте.

5. MS Excel-де деректердің бір типтес бағаны:

- A) өріс.
- B) жазба.
- C) форма.
- D) есеп.
- E) кілт.

6. Деректер базасы – ...
- А) белгілі бір зат туралы өзара байланысты деректер жиынтығы.
 - В) ақпаратты жинау және сақтау программасы.
 - С) Excel пішіміндегі деректерге ие кесте.
 - Д) еркін ақпараттың жиынтығы.
 - Е) деректерді толтыруды қолдайтын интерфейс.
7. Кестенің өрістерінде орналасқан элементтің қасиеттері үшін мәндердің жиынтығы ...
- А) вектор.
 - В) атрибут.
 - С) жазба.
 - Д) кесте.
 - Е) кілт.
8. Реляциялық деректер базасында ақпаратты сақтаудың басты нысаны:
- А) кесте.
 - В) сұраныс.
 - С) форма.
 - Д) есеп.
 - Е) модуль.
9. Деректер базасы кестесі элементінің қасиетін сипаттайтын жол:
- А) өріс.
 - В) бланк.
 - С) жазба.
 - Д) байланыс.
 - Е) кілт.
10. Реляциялық деректер базасының өріс типі ... анықталады.
- А) өріс атауымен.
 - В) деректер типімен.
 - С) ұяшықтың атауымен.
 - Д) кестенің түрімен.
 - Е) негізгі түрімен.

11. MS Excel кестесіндегі күн/уақыт өрісінің типі ...

- A) 65535 таңбаға дейінгі көпжолдық мәтін.
- B) кез келген түрдегі сан (заттық, бүтін және т.б.).
- C) 255 символға дейінгі мәтіннің әдеттегі жолы.
- D) деректердің логикалық мәндері (ақиқат/жалған).
- E) деректер форматы күні немесе уақыты.

12. Төмендегілердің қайсысы Excel нысаны болып табылмайды?

- A) Кестелер.
- B) Кілттер.
- C) Формалар.
- D) Есептер.
- E) Сұраныстар.

13. Сүзгілеуді қолдану үшін берілген әдісті таңдаңдар:

- A) Деректер $\Rightarrow$ Сұрыптау және сүзгі $\Rightarrow$ Сүзгілеу.
- B) Формулалар $\Rightarrow$ Сұрыптау және сүзгі $\Rightarrow$ Сүзгілеу.
- C) Кірістіру $\Rightarrow$ Сұрыптау және сүзгі $\Rightarrow$ Сүзгілеу.
- D) Басты $\Rightarrow$ Сұрыптау және сүзгі $\Rightarrow$ Сүзгілеу.
- E) Сүзгі $\Rightarrow$ Сұрыптау және сүзгі $\Rightarrow$ Сүзгілеу.

14. ДББЖ ... қызмет етеді.

- A) деректер базасын шебер басқару мүмкіндігін ұсыну үшін.
- B) деректерді пайдаланушылар арасында бөлу үшін.
- C) үлкен деректерді сақтау үшін.
- D) деректерді сипаттау үшін.
- E) инженерлік және ғылыми есептерді орындау үшін.

15. Деректер базасының ең көп тараған моделі:

- A) үлестірілген.
- B) иерархиялық.
- C) желілік.
- D) реляциялық.
- E) постреляциондық.

4-БӨЛІМ

PYTHON ПРОГРАММАЛАУ ТІЛІНДЕ АЛГОРИТМДЕРДІ ПРОГРАММАЛАУ

Күтілетін нәтижелер:

- бірөлшемді, екіөлшемді массивтерді пайдаланып, Python программалау тілінде программаларды жасау;
- сұрыптау алгоритмдерін қолдану.

§ 16–17. Бірөлшемді массив

Естеріңізге түсіріңдер:

- Python-дағы деректер типі;
- Python-дағы алгоритмдердің түрлері;
- Python-дағы таңдау операторлары;
- Python-дағы циклдік операторлар.

Массив – Массив – *Array*

Деректер типі – Типы данных – *Data Type*

Индекс – Индекс – *Index*

Меңгерілетін білім:

- массив ұғымы;
- индекстер;
- индекс пен массивтің қабылдайтын деректер типі.

«Массив» деген жаңа ұғымды оқып-үйренуді бастайық. Бұған дейін біз айнымалылармен, деректер типімен, яғни `int` – бүтін, `float` – нақты бөлшектер, `str` – символдық және т.б. мәндерді қабылдайтынымен танысқанбыз.

Әрбір айнымалы типі белгілі бір типтің деректерін сақтайды және компьютер жадынан өзіне тиіс көлемде орын алады. Мысалы, `int` жадыдан 1-ден 4 байтқа дейін орын алуы мүмкін. Бірөлшемді және көпөлшемді массивтер бар.

Массив – бір атаумен біріктірілген, бір типтегі деректерден тұратын айнымалылардың жиынтығы. Әрбір алынған жеке айнымалы **массив элементі** деп аталады. Массив элементтері индекстер арқылы белгіленеді, индекстер тік жақшаның ішінде жазылады және олар нөлден бастап нөмірленеді. Сондықтан да массивтерді кейде *индексі бар айнымалы* деп атайды.

Бірөлшемді массивтер

Бірөлшемді массив – бір атауға біріктірілген бірнеше бір типтес элементтерден тұратын жиын. Әрбір элементтің өзінің реттік нөмірі болады.

Реттік нөмір – массив элементінің индексі. Массив элементтері бірінен соң бірі ретпен орналасады.

Массивті жариялаған кезде массивтің атауын, массив элементтерінің типін көрсетеміз. Массивтің өлшемі көрсетілмейді, ол кез келген уақытта жаңа элементтерді қосу арқылы

кеңейтілуі мүмкін. Массивті Python-да пайдалану үшін array стандартты модулін импорттау қажет.

Python программасындағы қызметтік сөздер:

'and', 'or', 'in', 'is', 'not', 'if', 'else', 'elif', 'assert', 'break', 'continue', 'def', 'del', 'for', 'from', 'while', 'yield', 'import', 'print', 'return'.

Модуль – функцияларды, нысандарды, кластарды қамтитын дайын программалар. *Мысалы:*

```
import array as arr
a = arr.array('i'),
```

мұндағы arr – array модулін әрі қарай пайдалану үшін өзіміз қойған атау;

a – қазіргі жағдайдағы массив атауы;

i – массив элементінің типі (integer).

Массивтің әрбір элементі массив атауы мен оның элементінің индексін көрсету арқылы шақыра аламыз (16-сурет).

Массив элементінің мәні					Массив элементтерінің индексі				
0	1	2	3	4	5	6	7	8	9
7	-5	9	1	0	-2	4	3	6	8

16-сурет. Бірөлшемді массив

Бірөлшемді массивтермен жұмыс істеу соншалықты қиын емес.

```
# массивті жариялай firstArray
firstArray = arr.array('i')
```

Массив оның элементтерінің типін көрсету арқылы жарияланады, біздің жағдайда бұл – бүтін (int), содан кейін оның аты көрсетіледі, бұл – firstArray. Массивтің 8 элементтен тұрғаны ыңғайлы. Мәндерді өзгерту арқылы массивті инициализациялау керек. Оны массивтің барлық элементтеріне дәйекті түрде өтетін цикл арқылы орындау оңай. $i * 2 - 1$ формуласын қолдана отырып инициализациялаймыз, мұндағы i – массив индексінің мәні.

```
# деректер массивін шығару
for i in range(0, 8):
    firstArray.append(i * 2 - 1)
```

Маңызды мәлімет

Python-да программаны дұрыс рәсімдеу өте маңызды. Жол және оның арасындағы табуляция мен бос орындар маңызды қызмет атқарады. Шегіністе (артқа қарай) 4 бос орынды қалдыру ұсынылады.

Бірөлшемді массивті енгізу және шығару

Массив элементтерінің мәнін енгізу және шығару үшін параметрі бар цикл қолданылады.

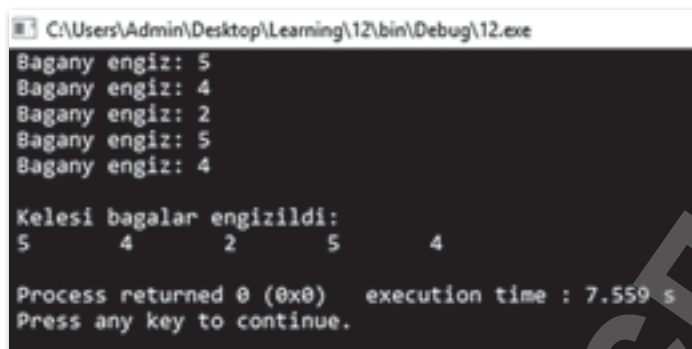
```
#циклде массивтің барлық элементтерін толтыру
for i in range(0,10):
    testArray.append(i)
#массивтің мазмұнын шығару циклі
for i in range(0,10):
    print(testArray[i])
```

Массивтер бір-екі айнымалымен жұмыс істеуге болмайтын жағдайларда, яғни бір типті элементтердің белгілі бір тізбегі болған кезде қолданылады. *Мысалы*, пайдаланушы белгілі бір орынмен, ретпен, басында немесе ортасында ауысуға, олардың мәндерін орындарымен ауыстыруға, өсу/кему және т.б. бойынша сұрыптауға тиіс.

Барлығын жақсы игеру үшін төмендегі мысалдың программасын қарастырайық:

```
#Программа пән бойынша қойылған бағаларды сұрайды
point1 = input("Bagany engiz: ")
point2 = input("Bagany engiz: ")
point3 = input("Bagany engiz: ")
point4 = input("Bagany engiz: ")
point5 = input("Bagany engiz: ")
print("Kelesi bagalar engizildi: ")
print(point1+'\t',point2+'\t',point3+'\t',
point4+'\t',point5+'\t')
```

Орындалған программаның нәтижесі (17-сурет):



17-сурет. Программа нәтижесі

Жоғарыда келтірілген программа пайдаланушыға бес айнымалы бойынша сақталатын бес бағаның нәтижесін сұрайды:

```
point1, point2, point3, point4, point5;
```

Енді осы программаның нұсқасын массивті пайдалану арқылы жасап көрейік:

```
#Программа пән бойынша қойылған бағаларды сұрайды
#Оларды сақтайды және басып шығарады
```

```
import array as arr
point = arr.array('i')
for i in range(0,5):
    point.append(int(input("Bagany engiz: ")))
# input әрқашан str типтеріне деректерді қайта-
рады, сондықтан int() функциясы санға түрлендіру үшін
пайдаланылады.
print("Kelesi bagalar engizildi: ")
for i in range(0,5):
    print(point[i], end='\t')
```

Берілген программа да алғашқы нұсқамен бірдей нәтиже береді, бірақ бұл жерде қысқаша келтірілген. Программалау нұсқасы массивтің қолданылуына байланысты *ыңғайлы* және *ықшамды* болып табылады.

Массивтің типтері

Енді программалауда пайдаланылатын массивтердің негізгі түрлерін қарастырамыз. Массивтер өлшемінің төмендегідей түрлері бар:

- бірөлшемді;
- екіөлшемді немесе матрица;
- үшөлшемді және т.б.

Практикада үшөлшемді массивтер сирек қолданылады. Үшөлшемді массивтерге оқулық немесе кітап мысал бола алады (1-ші индекс – кітаптың беттері десек, 2-ші индекс – беттегі жолдар, ал 3-ші индекс – жолдағы әріптер болады). Бірден көп өлшемі бар массивтерді **көпөлшемді** (екіөлшемді, үшөлшемді массивтер және т.б.) деп атайды.

Python программалау тілінде көпөлшемді классикалық массивтер жоқ. Дегенмен осы платформаның негізгі ерекшеліктері көпөлшемді тізімді (*list*) құруды жеңілдетеді.

Маңызды мәлімет

Массивтің элементтерін нөмірлеу нөлден басталады. Мысалы, егер бізде бес элементтен тұратын массив болса, онда оларға мынадай жолмен қол жеткізуге болады:

```
A[0]; A[1]; A[2]; A[3]; A[4].
```

Python программалау ортасында массивті жариялау

Массивті жариялау үшін, массивтің атауын және айнымалылардың типін көрсету керек.

Мысалы, массивті `float` типімен жариялайық:

```
import array as arr
a = arr.array('f')
```

Python программалау ортасында массивті инициализациялау

Біз массивті жариялағаннан кейін оны өз программamızда пайдалана аламыз, егер оны инициализацияламаған болсақ (оның элементтеріне бастапқы мән бермесек), оның ешқандай пайдасы болмайды. Массивті инициализациялауды бірнеше жолмен орындауға болады. Мысалы, `int` деп жариялағанда:

```
numbers_array = arr.array('i', [7, -5, 9, 1, 0,
-2, 4, 3, 6, 8])
```

Numbers_array массивінде 10 элемент бар. Массивтің өлшемін len () функциясын қолдану арқылы табуға болады:

```
len(numbers_array)
```

Егер бастапқыда 9 өлшемді массивтің барлық элементтері бірлік болса, онда циклді қолданамыз:

```
import array as arr
b = arr.array('i')
for i in range(0,9):
    b.append(1)
```

Массивті былай да беруге болады:

```
b = arr.array('i', [10, 20, 42])
```

Python программасындағы жүйелілік индексі нөлден басталады. Теріс индекстер элементті кезектің соңынан бастап көрсетеді (-1 – соңғы элемент). *Мысалы:*

```
>>> s = [1, 2, 3, 4, 5]
>>> print s[0], s[2], s[-1]
1 3 5
```

Қызықты ақпарат

Python тілін 1990 жылы Гвидо Ван Россум ойлап тапқан. Өлемдегі ең ірі компаниялар: NASA, Google, Yahoo, Microsoft және т.б. Python тілін өздерінің жобалары үшін пайдаланады.

1

Сұрақтарға жауап берейік

1. Массив деген не?
2. Массив қалай жарияланады? Массив қалай форматталады?
3. Массив қандай деректер типінен тұруы мүмкін?

2

Ойланайық, талқылайық

1. Индекс пен массив элементінің арасындағы айырмашылықты түсіндіріңдер.
2. Массивтерді пайдаланудың артықшылықтары туралы айтыңдар. Массивтердің қалай берілетіндігі туралы қысқаша сипаттама беріңдер: олар қалай жарияланады, массив элементтерін қалай оқи аламыз, массив элементтерін енгізу және шығару қалай жүргізіледі?

3

Талдап, салыстырайық

Python-дағы деректердің типтерін атаңдар, осы деректердің типтері бар массивтерге мысалдар келтіріп, талдаңдар.

4

Дәптерге орындайық

- Array массивінде бірнеше операцияны орындаңдар:
 - 1) 10-элементтің саны бар float типтегі массивті жариялап және оған нөлдік бастапқы мәндерді тағайындаңдар.
 - 2) Array = 10 массивінің 3-ші элементін шығарыңдар.
 - 3) Массивтің 8-ші элементіне 3,14 мәнін меншіктеңдер.
 - 4) Массивтің 1 элементінің мәнін 1 мәнге арттырыңдар.
- Жаңа терминдермен сөзжұмбақ немесе ребус құрастырыңдар: *массив, индекс, массив типі және т.б.*

5

Компьютерде орындайық

Есептердің нәтижелерін тексеріңдер.

```
import array as arr
Array = arr.array('f')
for i in range(0,10):
    Array.append(0)
print("Array[2]: ", Array[2])
Array[7] = 3.14
Array[0] += 1
```

6

Ой бөлісейік

Сабақта не үйрендіңдер? Сыныптастарыңмен программалауда массивтерді қалай пайдалануға болатынын талқылаңдар. Массивтерді пайдалану қаншалықты қажет?

7

Үй тапсырмасы

Array массивімен бірнеше амалды орындаңдар.

- 20 элементтері бар int типіндегі массивті жариялаңдар және оның бастапқы мәндеріне нөлді меншіктеңдер.
- Массив элементтеріне өздерінің индекстерін меншіктеңдер.

Шығармашылық тапсырма

Интернеттен алынған материалдарды пайдалана отырып, «Массивтерді тәжірибеде қолдану» атты презентация дайындаңдар.

§ 18–19. Берілген сипаттары бар элементтерді іздеу

Естеріңізге түсіріңдер:

- массив деген не?
- индекс не үшін қолданылады?

Меңгерілетін білім:

- шарттармен стандартты мәселелерді шешу;
- қосындының жинақталу жұмысы;
- көбейтіндінің жинақталу жұмысы.

Оң элементтер –

Положительные элементы –
Positive elements

Теріс элементтер –

Отрицательные элементы –
Negative elements

Жұп индекстер – Четные индексы – Even indexes

Тақ индекстер – Нечетные индексы – Odd indexes

Бірөлшемді массивтермен жұмыс істегенде, белгілі бір шарттармен элементтерді іздеу қажет болады. Бұл элементтердің жиынтығы арасында ең үлкен немесе ең аз элементті (максимум, минимум) табу, оң немесе теріс, жұп немесе тақ сандардың қосындысын есептеу бар.

Бірөлшемді массивтерге мысалдар

Массивтерді қолданудың мысалдарын практикалық түрде қарастырайық. $[1, 100]$ аралығынан кездейсоқ алынған 20 элементтен тұратын бірөлшемді массивті көрсетіңдер.

Нәтижесін есептептеңдер және шығарыңдар:

1. Массивтің барлық жұп элементтерінің қосындысы (S1).
2. Массивтің барлық тақ элементтерінің қосындысы (S2).
3. Массивтің ең үлкен элементінің мәні (max).
4. Массивтің ең кіші элементінің мәні (min).
5. Массивтің саны 30-дан кем (Sum) элементтерінің қосындысы.
6. 2-ден 9-ға дейінгі индекстермен берілген массив элементтерінің көбейтіндісі (P).

Берілген есептердің программасын тиянақты түрде қарастырып, талдаңдар. Алдымен массив элементтерімен жасалған жұмыс нәтижелерін сақтау үшін пайдаланылатын массив пен айнымалыларды анықтаймыз.

```

size = 20
a = arr.array('i', )
S1 = 0
S2 = 0
Sum = 0
P = 1

```

Біз массивті жарияладық, енді оған инициализациялауды орындауымыз қажет. Тапсырма бойынша массивті 1-ден 100-ге дейінгі кездейсоқ мәндермен толтыруымыз керек. Енді біздің бірөлшемді массивімізге кездейсоқ мәндерді берейік:

```

for i in range(size):
    a.append(random.randint(1,100))

```

Әрине, массивтің барлық элементтерін біртіндеп қарастыру үшін және әрқайсысына өзінің мәнін меншіктеуді цикл арқылы жасау өте ыңғайлы. `Randomint()` функциясының 1-ден 100-ге дейінгі аралықта кездейсоқ өлшемдерді реттеу үшін бастапқы және соңғы мәндерін беруіміз керек. Енді қажетті қосынды мен көбейтіндіні есептейтін негізгі мәселемізге көшейік. Тағы да массив бойынша қарастыру үшін `for` циклін пайдаланамыз.

```

for i in range(size):
    if a[i]%2 == 0:
        S1 += a[i]
    if a[i]%2 != 0:
        S2 += a[i]
    if a[i] < 30:
        Sum += a[i]
    if i == 2 or i == 9:
        P*=a[i]

```

Әдеттегідей барлық 0 мен `size` аралығындағы бірөлшемді массив элементтерін қарастыру үшін біз `for` циклін пайдаланамыз, мұндағы `size` – массив өлшемі. Шарт орындалғанда `size` қолданылмайды, себебі массив элементі 0-ден басталады. Демек, егер массив 20 элементтен тұратын болса, онда оның индекстері 0-ден 19-ға [19-ды қоса алғанда] дейін болады.

- 1) Одан кейін шарт бойынша массивтің жұп элементтерінің қосындысын табуымыз керек. Оны жасау үшін біз тиісті таңдау шарттарын қолданамыз.

```
if a[i]%2 == 0:
    S1 += a[i]
```

Мұнда бәрі анық: егер элементтің мәнін 2-ге бөлгенде қалған бөлігі нөлге тең болса, массивтің осы элементінің мәні тең және сәйкесінше ол жалпы қосындыға қосылуы мүмкін. Өйткені біз $S1$ айнымалысын жалпы соманы жинақтау үшін пайдаланамыз, сонда біз оны дереу инициализацияланған 0 мәнімен жариялаймыз. Бұл $S1 += a[i]$ қысқартылған өрнек екендігін және оның толық формасы $S1 = S1 + a[i]$ болатынын есімізге түсірейік.

- 2) Әрі қарай шарт бойынша массив элементтерінің барлық тақ мәндерінің қосындысын табу керек. Бұл қосындыны табу үшін кері шарт қолданылады, демек, бөліндінің қалған бөлігі 0-ге тең болмаса, онда қосындының айнымалысы бойынша жиынтықты орындаймыз.

```
if a[i]%2 != 0:
    S2 += a[i]
```

- 3–4) Енді ең үлкен элемент (max) және массивтің ең кіші элементін (min) табу мәселелеріне көшеміз.

Алгоритм мынадай: 1) цикл алдында массивтің бірінші элементін minimum және maximum айнымалыларына меншіктеуіміз керек, олар бастапқы ең аз және ең көп мән ретінде сақталады; 2) цикл кезінде бастапқы minimum-нан аз немесе maximum-нан көп мән табылса, онда оларды ауыстырып отырады; 3) максималды және минималды элементтері бар айнымалыларды шығару.

Массивтің әрбір элементін салыстыру керек: егер элементтің ең үлкен мәні айнымалы мәннен үлкен болса, онда мән maximum айнымалыға жазылады; егер элементтің ең аз мәні айнымалы мәннен аз болса, онда мән minimum айнымалыға жазылады:

```

    maximum = a[0] // Массивтің 1-ші элементінің
    мәнін айнымалыға орналастырамыз
    minimum = a[0]
    for i in range(size):
        #егер элементтің мәні max айнымалының мәнінен
        үлкен болса, онда бұл мәнді айнымалыға жазамыз
        if a[i] > maximum:
            maximum = a[i]
        #min-ге дәл солай
        if a[i] < minimum:
            minimum = a[i]

```

Сондай-ақ Python-да кірістірілген `max()`, `min()` функциялары бар:

```

maximum_1 = max(a)
minimum_1 = min(a)

```

- 5) 30-дан аз болатын барлық элементтің қосындысын табу үшін тиісті іріктеу шартын пайдаланыңдар.

```

if a[i] < 30:
    Sum += a[i]

```

- 6) Соңында біз 2-ші немесе 9-шы индекстегі массив элементтерінің көбейтіндісін табуымыз керек. Егер дәл осы уақытта біз массивтің 2-ші немесе (`or`) 9-шы элементтерінде тұрған болсақ, онда көбейтіндіні орындаңдар деген шарт береміз.

```

if i == 2 or i == 9:
    P *= a[i]

```

Төменде осы программаның толық коды көрсетілген:

```

#Бірөлшемді массив элементтерімен жұмыс істеу
мысалы
import array as arr
import random
random.seed(None)

#қажетті айнымалыларды жариялау
size = 20

```

```

a = arr.array('i', )
S1 = 0
S2 = 0
Sum = 0
P = 1

#массивті экранға шығару функциясы
def print Array(array):
    for i in range(len(array)):
        print(array[i], end=" ")
    print('\n')

#массивті кездейсоқ айнымалы мәндермен толтыру
for i in range(20):
    a.append(random.randint(1,100))

#массивтің мазмұнын экранда басып шығару
print Array(a)

#циклде қажетті есептеулерді орындау:
for i in range(size):
    if a[i]%2 == 0:
        S1 += a[i]
    if a[i]%2 != 0:
        S2 += a[i]
    if a[i] < 30:
        Sum += a[i]
    if i == 2 or i==9:
        P *= a[i]

maximum_1 = max(a)
minimum_1 = min(a)

maximum = a[0]
minimum = a[0]

for i in range(len(a)):
    if a[i] > maximum:
        maximum = a[i]

```

```

if a[i] < minimum:
    minimum = a[i]
#программаның нәжіжелерін көрсету:
print("zhup elementter kosyndysy: ", S1)
print("tak elemetter kosyndysy: ", S2)
print("Min: ", minimum, maximum_1)
print("Max: ", maximum, maximum_1)
print("30 deyingi elementter kosyndysy: ", Sum)
print("2 nemese 9 indexti elementter kobeitin-
disi: ", P)

```

1

Сұрақтарға жауап берейік

1. Массив элементтерін белгілі бір параметрлермен қалай іздеуге болады?
2. Max және min массивін табу алгоритмі қалай орындалады?

2

Ойланайық, талқылайық

Массив элементтерін белгілі бір параметрлермен іздеу әдісі туралы қысқаша сипаттама беріндер. Осы мәселелерді шешудің мәні неде? Массив элементтеріне қол жеткізу қалай орындалады, массив элементтерінің қосындысы мен көбейтіндісі қалай есептеледі?

3

Талдап, салыстырайық

Бірөлшемді массивтің барлық элементінің қосындысы оңай есептеледі. Ешқандай шартсыз, барлық элементі рет-ретімен, кезекпен орналасады.

```
sum += massiv[i];
```

Массивтің барлық элементтерінің көбейтіндісін есептеу формуласы қандай болады? Салыстырыңдар.

4

Дәптерге орындайық

Array массивімен бірнеше амалды орындаңдар:

1. Массивтің барлық элементтерінің қосындысын (S) табыңдар.
2. Массивтің барлық элементтерінің көбейтіндісін (P) табыңдар.

5

Компьютерде орындайық

1. Есептеулердің нәтижелерін тексеріңдер.

```
for i in range(size):
    S += Array[i]
for i in range(size):
    P1 *= Array[i];
```

2. CD дискіден осы тақырыпқа сәйкес тапсырманы орындандар.

6

Ой бөлісейік

Сабақта не білдіңдер? Не үйрендіңдер? Өз ойларыңды достарыңмен бөлісіңдер, алған жаңа білімдеріңді күнделікті өмірде қандай жағдайда қолдануға болады? Неге назар аудару қажет?

7

Үй тапсырмасы

Array қызметші сөзін пайдаланып, массив элементтерімен бірнеше амал орындандар:

1. Массивтің оң элементтерінің (S) қосындысын табыңдар.
2. Массивтің теріс элементтерінің (P) көбейтіндісін табыңдар.

§ 20. Практикум. Элементтердің орнын ауыстыру

Массив элементтерінің орнын ауыстыру алгоритмін қарастырайық. Ол үшін бірінші айнымалы мәнін уақытша жылжытатын айнымалы ұяшықты (буфер) пайдалану керек. Содан кейін бірінші айнымалы мен екінші айнымалы мәнін араластырамыз. Ал аралықтағы бірінші айнымалының сақталған мәні екінші айнымалыға ауысады. Бұл рәсім кесте ретінде мынадай түрде жазылады:

0	1	2	3	4	5	6	7	8	9
7	-5	9	1	0	-2	4	3	6	8

Buf

```
buf=a[0]
a[0]=a[1]
a[1]=buf
```

0	1	2	3	4	5	6	7	8	9
-5	7	9	1	0	-2	4	3	6	8

1-тапсырма. X массивіндегі жұп индексті элемент пен тақ индексті элементтің орындары ауысатындай етіп, оларды керісінше орналастырып, кейін у массивіне әкеліп қойыңдар.

```
import array as arr

x = arr.array('i')
print("x:", end="")
n = 10
x = input().split()[:n]
y=x

for i in range(10):
    if i % 2 == 0:
        c = x[i]
        x[i]=x[i+1]
        x[i+1]=c
    y[i] = x[i]
```

```
print("y:", end="")
for i in range(len(y)):
    print(y[i], end=" ")
```

Программа нәтижесі (18-сурет):

```
===== RESTART: C:\Users\Admin\Documents\123.py =====
x: 7 1 8 2 9 3 6 4 9 5
y: 1 7 2 8 3 9 4 6 5 9
>>>|
```

18-сурет. Программа нәтижесі

2-тапсырма. Берілген массивтегі ең үлкен және ең кіші (max және min) мәндерді ауыстырыңдар.

```
import array as arr

x = arr.array('i')
print("x=", end="")
n = 10
x = input().split()[:n]
max = x[0]
min = x[0]
for i in range(len(x)):
    if x[i] > max:
        max = x[i]
    elif x[i] < min:
        min = x[i]
imax=x.index(max)
imin=x.index(min)
print("Max: ", max, "Index of max element:", imax)
print("Min: ", min, "Index of min element:", imin)

b = x[imin]
x[imin] = x[imax]
x[imax] = b

for i in range(len(x)):
    print(x[i],end=' ')
print()
```

Программа нәтижесі (19-сурет):

```
===== RESTART: C:\Users\Admin\Documents\max min.py =====
x=7 8 9 -4 5 6 1 2 3 0
Max: 9 Index of max element: 2
Min: -4 Index of min element: 3
7 8 -4 9 5 6 1 2 3 0
>>>|
```

19-сурет. Программа нәтижесі

3-тапсырма. n элементтері бар сандық мәндердің бірөлшемді массиві берілген. Массив элементтерін айналдыра оңға қарай жылжыту керек:

$A[1] \rightarrow A[2]; A[2] \rightarrow A[3]; \dots A[n] \rightarrow A[1]$

```
import random

n=random.randint(1,20)
arr=[random.randint(0,100) for i in range(n)]
print(arr)

for i in range(n):
    arr[-i], arr[-i-1]=arr[-i-1], arr[-i]
arr[0], arr[1]=arr[1], arr[0]

print(arr)
```

§ 21. Практикум. Массив элементтерін сұрыптау

Массив элементтерін сұрыптауға массивтің элементтерінің өсу немесе кему тәртібімен орналасу жатады. Сұрыптау массивтің элементтерін түрлі әдістермен ауыстыру арқылы жүзеге асырылады.

Көпіршікті сұрыптау деп аталатын сұрыптау әдісін қарастырайық (айырбастау әдісі деп те аталады). Python тілінде жазылу алгоритмі мен программалау түрі көрсетіледі.

Реттелген массив бастапқы тізбек тұрған жадыда құралады. Бұл әдістің мақсаты – көршілес тұрған элементтерді бір-бірімен салыстыру.

Әрбір қадам жаңа жүйеліліктен басталады. Массивтің бірінші элементі екіншісімен салыстырылады, егер олардың арасындағы реттілік бұзылса, орындарын ауыстырамыз. Содан кейін екінші мен үшіншісі, үшінші мен төртіншісі және т.с.с. массивтің соңына дейін салыстырылып, жұптағы реттелмеген элементтердің орындары ауыстырылады.

Нәтижесінде бірінші қадамнан кейін ең үлкен (ең кіші элемент сұрыптау түріне қарай өсу/кему) элемент массивтің ең соңына жазылады. Әрбір элемент «қалқып» жоғары шығып отырады, сол себепті бұл сұрыптау *көпіршікті* деп аталады. Келесі қадамда 1-ден $N-1$ -ге, одан кейін 1-ден $N-2$ -ге, осылай соңына дейін тізбек қарастырылады. Әрбір қадамнан соң элементтердің орын ауысқандығын тексеруге болады. Егер жоқ болса, сұрыптау аяқталды.

1-тапсырма. Python программалау тілінде сұрыптау алгоритмі енгізілген:

```
import array as arr
#n – элементтер саны
n = int(input("Number of elements: "))
mass = arr.array('i')
#массивті толтырамыз
for i in range(n):
    print (i+1, end="-")
    mass.append(int(input("element: ")))
#бастапқы массивті шығару
print ("Initial array: ", end=" ")
```

```

for i in range (len(mass)):
    print (mass[i], end=" ")

print ("\n")
#массивті кему ретімен сұрыптау
for i in range(n):
    for j in range(n-1):
        if mass[j] < mass[j+1]:
#орын ауыстыру
            temp = mass[j]
            mass[j] = mass[j+1]
            mass[j+1] = temp
#сұрыпталған массивті шығару
print ("Final array: ", end=" ")
for i in range(len(mass)):
    print(mass[i], end=" ")

```

Программа нәтижесі (20-сурет):



```

C:\Windows\System32\cmd.exe
Number of elements: 7
1-element: 2
2-element: 6
3-element: 9
4-element: 1
5-element: 3
6-element: 7
7-element: 4
Initial array: 2 6 9 1 3 7 4
Final array: 9 7 6 4 3 2 1
C:\Users\Gulnur\Desktop\Informatics\source>

```

20-сурет. Программа нәтижесі

Python программалау тілінде өсу реті бойынша сұрыптау `sorted()` функциясы арқылы жүзеге асырылады. Тізбектің элементтерін кері тәртіпте көрсету үшін `reverse` пайдаланылады.

Өсу реті бойынша:

```

my_list = [2,6,9,1,3,7,4]
a = sorted(my_list)
print(a) # [1,2,3,4,6,7,9]

```

Кему реті бойынша:

```
my_list = [2,6,9,1,3,7,4]
a = sorted(my_list, reverse = True)
print(a) # [9,7,6,4,3,2,1]
```

2-тапсырма. Python программалау тілінің консолдық программасындағы массивті есептеп, нәтижесін көпіршіктеп сұрыптау әдісімен экранға шығарудың үлгісі.

```
import array as arr

n = int (input("Number of elements: "))
mass = arr.array('i')

for i in range(n):
    print (i+1, end="-i")
    mass.append(int(input("element: ")))

print("Initial array: ", end=" ")
for i in range(n):
    print(mass[i], end=" ")
print("\n")

noSwap = True
passnum = n-1
while passnum > 0 and noSwap:
    noSwap = False
    for i in range(passnum):
        if mass[i] > mass[i+1]:
            noSwap = True
            temp = mass[i]
            mass[i] = mass[i+1]
            mass[i+1] = temp
    passnum = passnum - 1

print("Final array: ", end=" ")
for i in range(n):
    print(mass[i], end=" ")
```

Программаның кодын талдап көрейік. Қажет кітапхана-ларды (1-жол) қосып, N массивтегі элементтер санын енгізіп және N-ді (3-жол) санап, массив құрамыз (4-жол), пайдалану-шыға массивтің элементтерін енгізіп, массивтің элементтеріне (6–8-жолдар) көпіршікті сұрыптау әдісін орындаймыз (16–26-жолдар), экранда сұрыпталған массивті көрсетеміз (29–31). Программа төмендегі суретте көрсетілген (21-сурет):

```

C:\Windows\System32\cmd.exe
Number of elements: 5
1-i element: 9
2-i element: -6
3-i element: 1
4-i element: 8
5-i element: -7
Initial array: 9 -6 1 8 -7
Final array: -7 -6 1 8 9
C:\Users\Gulnur\Desktop\Informatics\source>

```

21-сурет. Программа нәтижесі

3-тапсырма. Кестеде Нұр-Сұлтаннан Қазақстанның кейбір қалаларына дейінгі арақашықтық жазылған. Қалаларға дейінгі қашықтық берілген рет бойынша шығатындай, программа құрындар:

$S = \{1224, 196, 1505, 426, 276\}$

ҚР қалалары	Алматы	Қарағанды	Шымкент	Павлодар	Көкшетау
S, км	1224 км	196 км	1505 км	426 км	276 км

§ 22. Практикум. Элементті жою және кірістіру

Бірөлшемді массивтен элементті жою деген не? Элементті жою дегеніміз – элементтің массивтен «өшірілуі», элементтердің жалпы санының азаюы. Элементті массивке кірістіру кезінде элементтердің жалпы саны көбеюі керек.

Бастапқы тізбегі:

-5	7	9	1	0	-2	4	3	6	8
----	---	---	---	---	----	---	---	---	---

Массив элементін жойғаннан кейінгі тізбек:

-5	7	9	1	0	-2	4	3	6
----	---	---	---	---	----	---	---	---

Массив элементі кірістірілгеннен кейінгі тізбек:

-5	7	9	1	0	-2	4	3	6	8	-1
----	---	---	---	---	----	---	---	---	---	----

Python программалау тілінде жазылған элементтерді жою және массивтерге қою үшін пайдалануға болатын функциялар бар.

Элементтерді кірістіру үшін `insert()` және `append()` әдістерін, ал бірнеше элементті енгізу қажет болса, `extend()` әдісін қолданамыз.

`x.insert(i, n)` функциясы `x` массивінде `n` элементін көрсетілген `i` индексі бойынша кірістіруге мүмкіндік береді.

Біз `del` операторын пайдаланып, массивтің элементтерін жоя аламыз. Сондай-ақ элементті жоюға `remove()` немесе көрсетілген индексі бойынша элементті жою үшін `pop()` әдісін қолдануға болады.

`x.remove(n)` функциясы `n` элементін `x` массивінен жоюға мүмкіндік береді.

Мысал үшін осы алгоритмді қарастырайық:

```
import array as arr

#элементтер санын енгізу
n = int(input("Number of elements: "))
mass = arr.array('i')
```

```

#массивті экранға шығару функциясы
def print Array(a):
for i in range(len(a)):
    print(a[i], end=" ")
    print('\n')

#массивті инициализациялау
for i in range(n):
    print (i+1, end="-i")
    mass.append(int(input("element: ")))
print("Initial array: ", end=" ")
print Array(mass)

#жойғымыз келетін элемент нөмірін енгізу
m = int(input("Enter the number of element for
delete: "))

#элементті массивтен жою
del mass[m-1]
#элементті массивтің басына кірістіру
mass.insert(0,1)

#элементті массивтің соңына кірістіру
mass.append(9)
print ("Array: ", end=" ")
print Array(mass)

print("extend([5, 6, 7]) appends iterable to the
end of the array")

#элементтерді массив соңына кірістіру 5, 6, 7
mass.extend([5, 6, 7])
print ("Extended array: ", end=" ")
print Array(mass)

#3-ші элементті өшіру
print("Remove 3d element")
mass.pop(2)

```

```
#массивтегі 7-ге тең элементті жою
mass.remove(7)
print ("Final Array: ", end=" ")
print Array(mass)
```

Программа нәтижесі (22-сурет):

```
===== RESTART: C:\Users\Admin\Documents\delete_insert.py =====
Number of elements: 5
1-i element: 1
2-i element: 2
3-i element: 3
4-i element: 4
5-i element: 5
Initial array: 1 2 3 4 5

Enter the number of element for delete: 2
Array: 1 1 3 4 5 9

extend ([5, 6, 7]) appends iterable to the end of the array
Extended array: 1 1 3 4 5 9 5 6 7

Remove 3d element
Final Array: 1 1 4 5 9 5 6
```

22-сурет. Программа нәтижесі

1-тапсырма. Еркін берілген массивте элементтердің барлық жұп мәнін жойыңдар.

```
import random

a = []
for i in range(20):
    a.append(random.randint(1,100))
print("A =",a)

b = []
i = 0
while i < len(a):
    if a[i] %2 == 0:
        b.append(a[i])
        del a[i]
    else:
        i += 1
print("A =",a)
print("B =",b)
```

2-тапсырма. n өлшемді массив және k ($1 \leq k \leq n$) бүтін сан берілген. k реттік нөмірі бар массив элементінің алдына, нөлдік мәнімен жаңа элементті енгізіңдер.

```
import array as arr

n = int(input("Number of elements: "))
k = int(input("Number of elements: "))
mass = arr.array('i')

def printArray(a):
    for i in range(len(a)):
        print(a[i], end=" ")
    print('\n')

for i in range(n):
    print(i+1, end="-i ")
    mass.append(int(input("element: ")))

print("Initial array: ", end = "")
printArray(mass)

mass.insert(k,0)

print("Array: ", end = "")
printArray(mass)

print("Final Array: ", end = "")
printArray(mass)
```

§ 23. Екіөлшемді массивтер

Естеріңізге түсіріңдер:

- деректер типтерінің қандай түрлерін білесіңдер?
- *while, for* цикл операторлары қандай қызмет атқарады?
- бірөлшемді массивтерді енгізу және шығару қалай жүзеге асады?

Меңгерілетін білім:

- көпөлшемді массивтермен жұмыс;
- *тізім, тізім өлшемі;*
- *тізімдерді енгізу және шығару.*

Тізім – Список – List

Есептерде деректерді кесте түрінде сақтау жиі кездеседі. Мұндай деректер **матрица** немесе **екіөлшемді массив** деп аталады. Python программалау тілінде көпөлшемді массивтер жоқ. Алайда кез келген кесте тізімдер тізімімен ұсынылуы мүмкін. Яғни тізімнің әрбір элементі тізім болып табылады.

Тізім негіздері (List)

Python-дағы тізім кез келген типтегі элементтердің реттелген жиынтығы болуы мүмкін. Массив – бір типтегі элементтердің реттелген жиынтығы, сондықтан тізім массивке қарағанда икемді, бірақ бұл икемділік қарапайым құрылыммен жұмыс істейтін кезде қиындық туғызады. Тізім – динамикалық өзгертін тип, бұл кез келген уақытта тізімге элементтерді қосуға және тізімдегі элементтерді жоюға болады дегенді білдіреді.

Екіөлшемді массивті жазу үлгісін қарастырайық:

```
my_list1 = [[11, 12, 5, 2], [15, 6, 10, 5], [10, 8, 12, 5], [12, 15]]
```

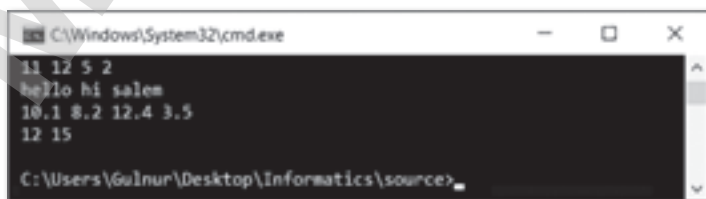
Көпөлшемді массивте әртүрлі бағандар мен жолдар болуы мүмкін екенін ескеру керек.

Екіөлшемді массивтерде әртүрлі типтер болуы мүмкін.

```
my_list2 = [[11, 12, 5, 2], ['hello', 'hi', 'salem'], [10.1, 8.2, 12.4, 3.5], [12, 15]]
```

Бұл массив `my_list2`: `int`, `string`, `float` сияқты түрлі деректер типіндегі кірістірілген тізімдерінен тұрады.

`my_list2` нәтижесін шығару (23-сурет):



23-сурет. Программа нәтижесі

Екіөлшемді массивтердегі мәндерге қол жеткізу

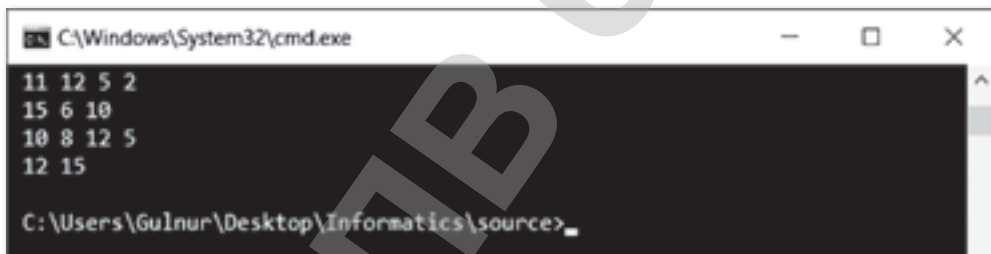
Екіөлшемді массивтердегі деректер элементтеріне екі индекс арқылы қол жеткізуге болады. Бір индекс негізгі массивке, екіншісі ішкі массивте деректер элементінің орналасуына қатысты. Егер тек бір индексті алатын болсақ, онда осы массивтің барлығы индекстің сол позициясы үшін ғана жазылады. Мысал қарастырайық:

```
print(my_list1[0]) - экранға шығарады [11, 12, 5, 2]
print(my_list1 [1][2]) - 10-ды экранға шығарады.
```

Толық екіөлшемді массивті басып шығару үшін, біз for циклін пайдалана аламыз:

```
for i in myList1:
    for j in i:
        print (j, end=" ")
    print ()
```

Келтірілген код мынадай нәтижені береді (24-сурет):



24-сурет. Программа нәтижесі

1

Сұрақтарға жауап берейік

1. Тізім деген не?
2. Тізімдер қандай деректер типінен тұруы мүмкін?
3. Екіөлшемді массивтің енгізу және шығару элементтері қалай жазылады?
4. Екіөлшемді массив элементтерімен қалай байланыс жасауға болады?

2

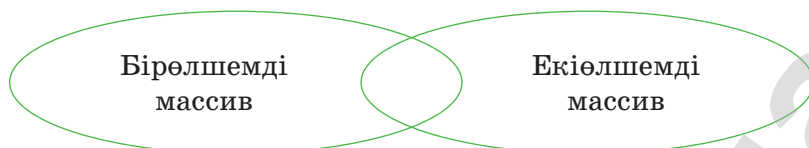
Ойланайық, талқылайық

Тізімдермен жұмыс істеудің артықшылықтары туралы айтыңдар. Кірістірілген тізімдер массивтерде қалай анықталады? Екіөлшемді массив элементтеріне қалай қол жеткізуге болады?

3

Талдап, салыстырайық

Венн диаграммасы көмегімен бірөлшемді және екіөлшемді массивті салыстырыңдар.



4

Дәптерге орындайық

Төмендегі кодты орындағаннан кейін қандай нәтиже аламыз? Дәптерлеріңе жазыңдар.

```
my_list = [[7, 12, 4], [9, 50, 3], [45, 0, -1]]
print(my_list[1])
print(my_list[2])
print(my_list[-1])
```

5

Компьютерде орындайық

Берілген программаны компьютерде орындаңдар.

```
my_list1 = [[7, 12, 4], [9, 50, 3], [45, 0, -1]]
my_list2 = [[7, 18, 6], [11, 3]]
my_list3 = [['Almaty', 'Dubai', 'London'],
['Kazakhstan', 'UAE', 'England']]
print (my_list1[1])
my_list2[1] = [11, 3, 5]
print (my_list2)
print (my_list3[1][0])
```

6

Ой бөлісейік

Сабақта жаңа не үйрендіңдер? Бірөлшемді және екіөлшемді массивтер қалай ерекшеленеді?

7

Үй тапсырмасы

Екіөлшемді массивті төмендегідей түрде көрсететін Python программасын жазыңдар:

```
1
2 3
4 5 6
7 8 9 10
```

§ 24. Практикум. Екіөлшемді массивте элементті жою және кірістіру

Екіөлшемді массивте мәндерді кірістіру

Біз `insert ()` әдісін қолданып, анықталған позицияларға жаңа элементтердің индексін көрсете отырып, кірістіре аламыз.

1-тапсырма. Келтірілген мысалда жаңа деректердің элементі 2 индексті позицияға кірістіріледі.

```
from array import *
my_list = [[11, 12, 5, 2], [15, 6, 10], [10, 8, 12, 5], [12, 15]]

my_list.insert(2, [0, 5, 11, 13, 6])

for i in my_list:
    for j in i:
        print (j, end=" ")
        print ()
```

Программа нәтижесі мынадай болады (25-сурет):



25-сурет. Программа нәтижесі

Екіөлшемді массивтегі мәндерді жою

`Del ()` әдісін пайдалана отырып, индекс көмегімен мәндерді қайта анықтау арқылы ішкі массивті немесе ішкі массивтің кейбір нақты элементтерін жоюға болады. Бірақ белгілі бір деректер элементтерін ішкі массивтердің бірінде жою қажет болса, төмендегі сипатталған жаңарту процесін пайдаланамыз.

2-тапсырма. Бұл мысалда массивтегі элементтерді жоюды қарастырамыз.

```
from array import *
myList = [[11, 12, 5, 2], [15, 6, 10], [10, 8, 12,
5], [12, 15]]

del myList[3]

for i in myList:
    for j in i:
        print (j, end=" ")
    print()
```

Программа нәтижесі былай болады (26-сурет):



26-сурет. Программа нәтижесі

3-тапсырма. 5x5 матрицасы түрінде екіөлшемді массив берілсін. Жұп бағандағы барлық элемент мәндерін 0 мәнімен, ал тақ бағандағы элемент мәндерін 1 санына ауыстырындар.

§ 25. Практикум. Екіөлшемді массивтерді шешу

1-тапсырма. Екі санды m (жол) және n (баған) тұратын кіріс деректер ретінде қабылдайтын және екіөлшемді массив жасайтын Python программасын жазыңдар. i жолда және j бағанында элементтің мәні $i * j$ болуы тиіс.

Программа листингі:

```
row = int(input("Input number of rows: "))
column = int(input("Input number of columns: "))
my_list = [[0 for col in range(column)] for r in range(row)]

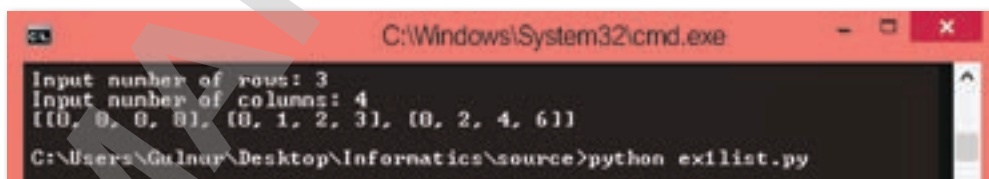
for i in range(row):
    for j in range(column):
        my_list[i][j] = i*j

print (my_list)
```

Жолдар мен бағандар санын сұраймыз. Екіөлшемді массивтерді $row * column$ нөлдермен толтырамыз. Әр элементті жолдар мен бағандар индекстерінің қиылысында олардың көбейтіндісімен ауыстырамыз.

Мысалы, 2-жолда және 3-бағанда матрица элементінің мәні мынадай түрде шығарылады: 2- жолдың индексі 1-ге тең, себебі массивтегі индекстеу 0-ден басталады; тиісінше 3-бағанның индексі 2-ге тең. Элементтің мәні $my_list[1][2] = 1 * 2 = 2$. Массивті экранға шығарамыз.

Программа нәтижесі мынадай болады (27-сурет):



27-сурет. Программа нәтижесі

2-тапсырма. Екіөлшемді массивтің өлшемін есептейтін және оны 0 мен 50 арасындағы кездейсоқ сандармен толтыратын программаны жазыңдар. Әр бағанның қосындысын және әр жолдың көбейтіндісін экранға шығарыңдар.

Программа листингі

```

import random
random.seed(None)
row = int(input("Input number of rows: "))
column = int(input("Input number of columns: "))
my_list = [[0 for col in range(column)] for r in
range(row)]

sum_column = [0 for col in range(column)]
product_row = [1 for row in range(row)]

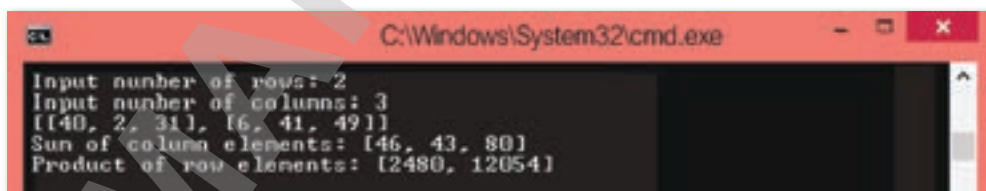
for i in range(row):
    for j in range(column):
        my_list[i][j] = random.randint(0,50)
print (my_list)

for i in range(column):
    for j in range(row):
        sum_column[i] += my_list[j][i]

for i in range(row):
    for j in range(column):
        product_row[i] *= my_list[i][j]

print("Sum of column elements: ", sum_column)
print("Product of row elements: ", product_row)

```

Программа нәтижесі мынадай болады (28-сурет):**28-сурет. Программа нәтижесі**

3-тапсырма. Көбейту кестесін бейнелейтін программа жасаңдар. Алынған нәтиже тізім түрінде көрсетілуі керек. Программаны «5 * 5 көбейтіндісі неге тең?» деген нәтиже шығатындай толтырыңдар.

4-БӨЛІМДІ ҚОРЫТЫНДЫЛАУҒА АРНАЛҒАН ТЕСТ ТАПСЫРМАЛАРЫ

1. Қай жазбада массивті жариялау көрсетілген?

- A) `var A: array [1..10] of integer.`
- B) `for i in range(10): if A[i]<min: min=A[i].`
- C) `S=0 for i in range(10): S+=array[i].`
- D) `for i in range(10): A[i]=random.randint(1,100).`
- E) `a = arr. array('i').`

2. Қай жазбада массив элементтерін енгізу көрсетілген?

- A) `a = arr.array('i').`
- B) `for i in range(10): if A[i]<min: min=A[i].`
- C) `S=0 for i in range(10): S+=array[i].`
- D) `for i in range(10): A[i]=random.randint(1,100).`
- E) `var A: array [1..10] of integer.`

3. Қай жазбада массив элементтерін шығару көрсетілген?

- A) `for i in range(10): print(A[i]).`
- B) `for i in range(10): if A[i]<min: min=A[i].`
- C) `S=0 for i in range(10): S+=array[i].`
- D) `for i in range(10): A[i]=random.randint(1,100).`
- E) `var A: array [1..10] of integer.`

4. Қай жазбада оң сандар көрсетілген?

- A) `B[i]%2!=0.`
- B) `B[i]%2==0.`
- C) `B[i]>0.`
- D) `B[i]<0.`
- E) `B[i]<>0.`

5. Қай жазбада теріс сандар көрсетілген?

- A) `B[i]>0.`
- B) `B[i]%2==0.`
- C) `B[i]%2!=0.`
- D) `B[i]<0.`
- E) `B[i]<>0.`

6. Қай жазбада жұп сандар көрсетілген?

- A) `B[i]>0.`

- B) $B[i] \% 2 == 0$.
- C) $B[i] \% 2 != 0$.
- D) $B[i] < 0$.
- E) $B[i] = 0$.

7. Қай жазбада тақ сандар көрсетілген?

- A) $B[i] \% 2 != 0$.
- B) $B[i] > 0$.
- C) $B[i] \% 2 == 0$.
- D) $B[i] < 0$.
- E) $B[i] = 0$.

8. Массивтің 4-элементі 7-ге тең болуы қай жазбаға сәйкес?

- A) $B[3] = 7$.
- B) $B[4] = 7$.
- C) $B[7] = 4$.
- D) $B[7] = 3$.
- E) $B[i] = 7$.

9. Массивтің әрбір элементінің екі есеге өсуі қай жазбада дұрыс жазылған?

- A) `for i in range(n): A[2*i]=A[i].`
- B) `for i in range(n): A[i]=A[i]/2.`
- C) `for i in range(n): A[i]=A[i]+2.`
- D) `for for i in range(1,n): A[i]=A[i]+2.`
- E) `for i in range(n): A[i]=A[i]*2.`

10. Массивтің әрбір элементінің екі есеге азаюы қай жазбаға сәйкес?

- A) `for i in range(1,n): A[i]=A[i]+2.`
- B) `for i in range(n): A[i]=A[i]*2.`
- C) `for i in range(n): A[i]=A[i]+2.`
- D) `for i in range(n): A[i]=A[i]/2.`
- E) `for i in range(n): A[2*i]=A[i].`

11. Массив элементтерінің көбейтіндісін тап.

- A) `P=1 for i in range(0,n): P*=A[i].`
- B) `P=0 for i in range(0,n) P*=A[i].`
- C) `P=1 for i in range(1,n) P*=A[i].`
- D) `P=0 for i in range(1,n) P*=A[i].`
- E) `P=n for i in range(0,n) P*=A[i].`

12. Массивтің теріс элементтерінің қосындысын тап.

- A) `S=0 for i in range(0,n): if A[i]<0: S+=A[i].`
- B) `S=0 for i in range(0,n): if A[i]>0: S+=A[i].`
- C) `S=1 for i in range(0,n): if A[i]<0: S+=A[i].`
- D) `S=0 for i in range(1,n): if A[i]<0: S+=A[i].`
- E) `S=0 for i in range(0,n): if A[i]<>0: S+=A[i].`

13. Массивтің оң элементтерінің көбейтіндісін тап.

- A) `P=1 for i in range(0,n): if A[i]>0: P*=A[i].`
- B) `P=1 for i in range(0,n): if A[i]<0: P*=A[i].`
- C) `P=0 for i in range(0,n): if A[i]<0: P*=A[i].`
- D) `P=1 for i in range(1,n): if A[i]>0: P*=A[i].`
- E) `P=1 for i in range(0,n): if A[i]<>0: P*=A[i].`

14. Массивтің жұп элементтерінің қосындысын тап.

- A) `S=0 for i in range(0,n): if A[i]%2 == 0: S+=A[i].`
- B) `S=0 for i in range(0,n): if A[i]%2 = 0: S+=A[i].`
- C) `S=1 for i in range(0,n): if A[i]%2 == 0: S+=A[i].`
- D) `S=0 for i in range(1,n): if A[i]%2 == 0: S+=A[i].`
- E) `S=0 for i in range(0,n): if A[2*i] == 0: S+=A[i].`

15. Массивтің тақ элементтерінің көбейтіндісін тап.

- A) `P=1 for i in range(0,n): if A[i]%2 != 0:
P*=A[i].`
- B) `P=1 for i in range(0,n): if A[i]%2 != 0:
P*=A[i].`
- C) `P=0 for i in range(0,n): if A[i]%2 != 0:
P*=A[i].`
- D) `P=1 for i in range(0,n): if A[i]%2 == 0: P*=A[i].`
- E) `P=1 for i in range(0,n): if A[i]%2 <> 0: P*=A[i].`

5-БӨЛІМ

PYTHON ПРОГРАММАЛАУ ТІЛІНДЕ 2D ОЙЫНЫН ҚҰРУ

Күтілетін нәтижелер:

- PyGame кітапханасын қосу;
- ойын терезесін жасау үшін PyGame кітапханасының дайын модульдерін пайдалану;
- ойынның артқы фонын құру;
- ойынға арналған дайын кейіпкерлерді жүктеу;
- кейіпкер қозғалысын программалау;
- пернетақтадан кейіпкерді басқару;
- дайын сценарий бойынша ойын әзірлеу;
- ойынның нәтижелерін есептеу алгоритмін іске асыру.

§ 26. PyGame кітапханасы. Кітапхананы қосу

Естеріңізге түсіріңдер:

- Python-дағы алгоритмдердің қандай типтерін білесіңдер?
- тармақталу және циклдік операторлары деген не?
- Python-дағы деректер типін айтыңдар.

Меңгерілетін білім:

- PyGame кітапханасы;
- PyGame кітапханасын іске қосу;
- PyGame модульдерінің қызметі.

PyGame кітапханасы –
Библиотека PyGame –
PyGame Library

Қосу – Запустить – *Initiate*

Оқиға – Событие – *Event*

Кітапхана – функция мен кластар кодтарының жиынтығы. Python-да код кітапханасын сипаттау үшін қолданылатын термин – *модуль (module)*. Мысалы, **pygame** модулінде ішкі модульдер ретінде *pygame.draw*, *pygame.image*, *pygame.mouse* және т.б. қолданылады.

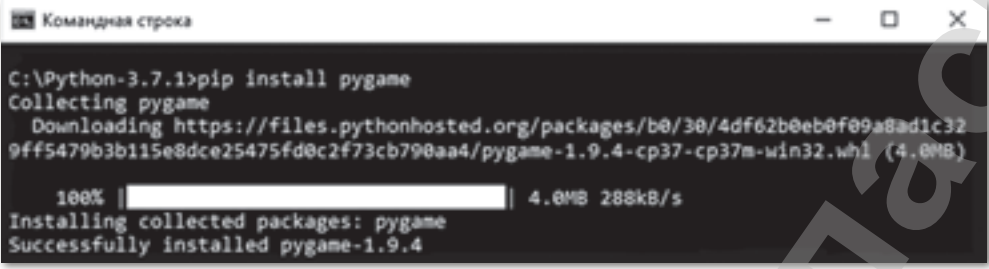
PyGame кітапханасы – 2D ойындарын және басқа мультимедиялық қосымшаларды әзірлеуге арналған Python программалау тілінің модулі. PyGame көптеген платформалар мен операциялық жүйелерде жұмыс істеуге мүмкіндік беретін SDL (Simple DirectMedia Layer) ойындарын жасау үшін басқа кітапхана негізінде жасалды. PyGame – Python программалаушыларына осы қуатты кітапхананы пайдалануға мүмкіндік беретін SDL байланысы.

PyGame өте ыңғайлы болғанымен Python-ның стандартты кітапханасына кірмейді, яғни ол орнату бумасымен бірге жеткізілмейді, оны бөлек орнату қажет. Ол үшін өзіміз қолданып отырған Python 3.4. нұсқасына сәйкес PyGame кітапханасын www.pygame.org сайтынан жүктей аламыз немесе командалық жолдан

```
Python - m pip install - U pygame - user
немесе
```

```
Python - m pip install pygame
```

командасын орындаймыз (29-сурет).



```

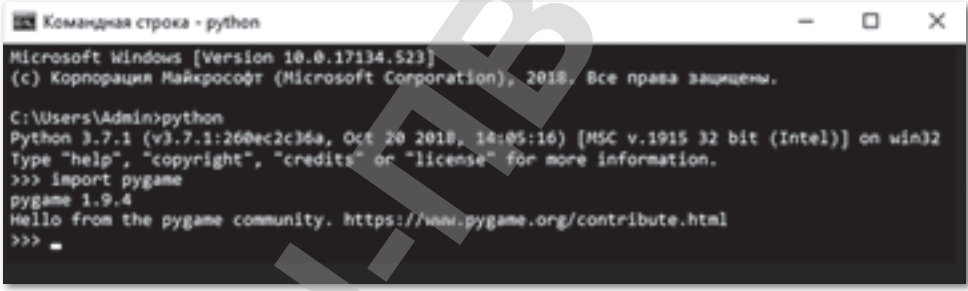
C:\Python-3.7.1>pip install pygame
Collecting pygame
  Downloading https://files.pythonhosted.org/packages/b0/30/4df62b0eb0f09a8ad1c329ff5479b3b115e8dce25475fd0c2f73cb790aa4/pygame-1.9.4-cp37-cp37m-win32.whl (4.0MB)
    100% |████████████████████████████████████████| 4.0MB 288kB/s
Installing collected packages: pygame
Successfully installed pygame-1.9.4
  
```

29-сурет. PyGame орнату

Тиісті буманы орнатқаннан кейін Python интерпретаторын ашып, мына команданы енгізу арқылы оны тексеруге болады:

```
import pygame.
```

Егер кітапхана дұрыс орнатылса, сендер PyGame-нің 30-суреттегі үлгісін көресіңдер.



```

Microsoft Windows [Version 10.0.17134.523]
(c) Корпорация Майкрософт (Microsoft Corporation), 2018. Все права защищены.

C:\Users\Admin>python
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import pygame
pygame 1.9.4
Hello from the pygame community. https://www.pygame.org/contribute.html
>>> _
  
```

30-сурет. PyGame орнатылғандығын тексеру

PyGame кітапханасын импорттағаннан кейін оны инициализациялау керек:

`pygame.init()` – аппараттық жасақтаманы сұрайтын PyGame пигменттік пакеттегі әрбір ішкі модульді инициализациялайды, драйверлерді жүктеп, PyGame компьютердің барлық құрылғыларын пайдалануға дайын болады.

PyGame кітапханасы бірнеше модульден тұрады, олар жеке-жеке қолданылуы мүмкін.

Ойындарды жасаған кезде қолдануға болатын әрбір құрылғы үшін модульдер бар.

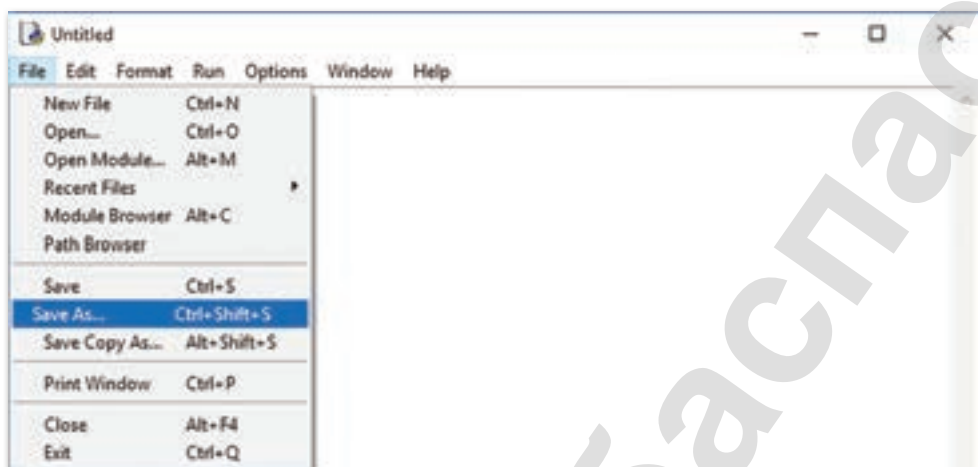
15-кестеде кейбір модульдердің тізімі және мақсаты көрсетілген. Толық тізімді және құжатты мына жерден табуға болады www.pygame.org/docs/. Бұл модульдерге қол жеткізу үшін, PyGame арқылы шақыру қолданылады.

15-кесте. PyGame-нің негізгі модульдері

Модуль атауы	Мақсаты
pygame.cdrom	CD-ROM жетектерін басқару және оған кіру
pygame.cursors	Меңзер бейнелерін жүктеу
pygame.display	Экранға кіру
pygame.draw	Фигураларды, сызықтарды және нүктелерді сызу
pygame.event	Сыртқы оқиғаларды басқару
pygame.font	Жүйелік қаріптерді қолдану
pygame.image	Кескінді жүктеу және сақтау
pygame.key	Пернетақтада басқан пернені оқу
pygame.mixer	Дыбыстарды жүктеу және ойнату
pygame.mouse	Тінтуірмен басқару
pygame.movie	Кинофильм файлдарын ойнату
pygame.music	Музыка және аудиолармен жұмыс
pygame.rect	Төртбұрышты аймақтарды басқару
pygame.sndarray	Дыбыс деректерін басқару
pygame.surface	Суреттер мен экранды басқару
pygame.transform	Суреттерді өзгерту және жылжыту
pygame.time	Уақыт пен кадр жиілігін басқару

Python программасын PyGame кітапханасымен пайдалануда құжатты құру және сақтау әдеттегідей орындалады. Ол үшін IDLE интеграцияланған әзірлеу ортасында **File** мәзірін ашып және **New File** элементін таңдаймыз. Кейін қосымша терезе ашылады, бұл терезеде **File** мәзірін ашып, ... **ретінде сақтау** (Save As...) командасын орындаймыз.

Содан соң орынды таңдап, «*.ру» кеңейтілімімен файлды сақтаймыз (31-сурет).



31-сурет. Python-да құжатты сақтау

1

Сұрақтарға жауап берейік

1. Python кітапханалары деген не?
2. PyGame кітапханасын қалай іске қосуға болады?
3. PyGame кітапханасының қызметінің мәні неде?
4. PyGame модульдерін атаңдар.

2

Ойланайық, талқылайық

1. PyGame Python-ның стандартты кітапханасы ма?
2. Не себепті `import pygame` қолданылады?
3. Here PyGame модульдері бірнеше түрге бөлінеді?

3

Талдап, салыстырайық

Python нұсқасы мен PyGame модулінің нұсқаларының салыстырмалы талдауын жүргізіңдер.

4

Дәптерге орындайық

PyGame кітапханасын орнату алгоритмін дәптерлеріңе жазыңдар.

5

Компьютерде орындайық

1. Сыныптағы компьютерлерге PyGame орнатындар.
2. PyGame программасының орнатылғандығын тексеріңдер.
3. PyGame кітапханасында жасалған программаны сақтау жолдарын көрсетіңдер.

6

Ой бөлісейік

Сабақта сендер не үйрендіңдер? Қандай сайттарда немесе онлайн ресурстарда PyGame-де жасалған программаларды көруге болады?

7

Үй тапсырмасы

1. Python мен PyGame тақырыптарының терминдеріне қатысты сөзжұмбақ құрастырыңдар.
2. CD дискіден осы тақырыпқа сәйкес тапсырманы орындаңдар.

§ 27. Ойын терезесін жасау үшін PyGame кітапханасының дайын модульдерін пайдалану

Естеріңізге түсіріңдер:

- Python-дағы кітапхана, модуль дегеніміз не?
- PyGame модульдерінің типтері қандай?
- PyGame модульдері қалай тағайындалады?

Экран – Экран – Screen

Режим қою – Задать режим – Set mode

Шығу – Выход – Quit

Менгерілетін білім:

- `pygame.display` модулін тағайындау;
- `pygame.display` модулінің қызметі;
- терезені құру үшін программаны жазу.

Ойын әзірлеу кезінде бірінші қадам – терезені құру, яғни ойын алаңы болып табылады. PyGame кітапханасында дисплей терезесі мен экранды басқаруға арналған дайын `pygame.display` модулі бар.

16-кестеде осы модульдің кейбір ерекшеліктері көрсетілген:

16-кесте. PyGame.display модулінің қызметі

Қызметтер	Сипаттамалары
<code>pygame.display.init</code>	Дисплей модулін баптау
<code>pygame.display.quit</code>	Дисплей модулін жабу
<code>pygame.display.flip</code>	Дисплей бетін толығымен жаңарту
<code>pygame.display.set_mode</code>	Бейнені көрсету үшін терезені немесе экранды инициализациялау
<code>pygame.display.get_surface</code>	Ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу
<code>pygame.display.toggle_fullscreen</code>	Толық экран мен терезе дисплей арасындағы ауысу
<code>pygame.display.set_caption</code>	Терезе атауын тағайындау
<code>pygame.display.get_caption</code>	Ағымдағы терезенің тақырыбын қайтару

PyGame терезесі бір көрініс қабатынан тұрады, ол терезенің ішінде немесе толық терезе режимінде жұмыс жасайды. $x = 0$ және $y = 0$ -дегі кескіннің басы – экранның жоғары сол жақ бұрышы. Екі ось экранның оң жақ төменгі жағында оң бағытта өседі.

PyGame-де тек бір дисплей белсенді болуы мүмкін. `pygame.display.set_mode()` функциясын пайдаланып, жаңа элемент жасау арқылы алдыңғы экранды жаба аламыз. *Мысалы*, ойынға арналған өлшемі 600x500 болатын `screen` терезесін құрамыз (бұл өзіміз берген терезе атауы). Ол үшін біз төмендегі функцияны қолданамыз:

```
pygame.display.set_mode(resolution = (x, y), flags = 0, depth = 0)
```

Resolution аргументі – ені мен биіктігін білдіретін сандар жұбы. **Flags аргументі** – қосымша опциялар жиынтығы. **Depth аргументі** түс үшін пайдаланылатын биттердің санын білдіреді.

Ойынымыздың атауы «My First Game». Төменде сипатталған функцияны пайдаланып, терезенің атауын орнатамыз:

```
pygame.display.set_caption("Some title")
```

Программа коды:

```
import pygame
pygame.init()
screen = pygame.display.set_mode((600, 500), 0, 32)
pygame.display.set_caption("My First Game")
```

Негізгі программа циклі ойын логикасы жазылған блок болып табылады:

```
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            exit()
    screen.blit(background, (0,0))

    pygame.display.update()
```

Шарт ретінде while циклі True мәнін қабылдайды, біз одан шыққанша немесе оны басқа жолмен тоқтатқанға дейін, үздіксіз қайталанатын болады.

Фондық суреттің терезесін толығымен жабу үшін біз келесі әрекеттерді қолдана аламыз:

```
screen.blit (фон, (0,0,0))
```

Соңында, жаңа суретті алу үшін экранды жаңартамыз:

```
pygame.display.update ()
```

Программа циклі 3 негізгі бөлімнен тұрады:

- пайдаланушымен байланыс: пайдаланушының кіруін және оларға жауапты тұрақты «тыңдау» үшін пайдаланылады. Бұл пайдаланушы пернетақта немесе тінтуірді пайдаланған кезде болуы мүмкін;
- ойын логикасын енгізу;
- көріністі және кейіпкерлерді қайта жасау арқылы экранды жаңарту.

Біздің листингтен шығу кезінде белгілі бір өлшеммен және «My First Game» атымен терезе аламыз.

Осы модульдің кейбір функцияларын қолдану үлгісін қарастырайық (32-сурет):

- 1) print("get_caption: ", pygame.display.get_caption())
- 2) print("get_surface: ", pygame.display.get_surface())
1. PyGame функциясы display.get_caption () дисплей бетінің атауы мен белгішесінің атауын қайтарады. Олар жиі бірдей мәнге ие болады.
2. pygame.display.get_surface () функциясы ағымдағы көрсетілген дисплей бетіне сілтеме береді. Егер дисплей режимі орнатылмаған болса, None қайтарылады.

```
===== RESTART: C:\Users\Admin\Documents\display.get.py =====
pygame 1.9.4
Hello from the pygame community. https://www.pygame.org/
contribute.html
get_caption: ('My First Game', 'My First Game')
get_surface: <Surface (600x500x32 SW) >
>>>
```

32-сурет. Модульдің қосымша функцияларын қолдану

1

Сұрақтарға жауап берейік

1. `Pygame.display` модулінің негізгі мақсаты қандай?
2. `Pygame.display.set_mode()` қандай қызмет атқарады?
3. `Pygame.display.flip()` қандай қызмет атқарады?
4. `Pygame.quit()` қандай қызмет атқарады?

2

Ойланайық, талқылайық

`Pygame.display.set_mode()` функциясын пайдалану үшін қандай екі операция орындалуы керек?

3

Талдап, салыстырайық

Компьютерлік экрандардың координата жүйесі мен стандартты тікбұрышты координата жүйесі арасындағы айырмашылықты талдап, түсіндіріңдер.

4

Дәптерге орындайық

`for event in pygame.event.get()` циклдік команда-сын дәптерге жазыңдар.

```
while True:
```

```
    for event in pygame.event.get():
```

```
        if event.type == QUIT:
```

```
            exit()
```

5

Компьютерде орындайық

Сабақта талқыланған мысалда экран параметрлерін (300,200) немесе (200,300): `screen = pygame.display.set_mode((x, y), 0.32)` өзгертіп көріңдер. Қандай айырмашылықтар байқадыңдар?

6

Ой бөлісейік

Сабақта не үйрендіңдер? Өз сыныптастарыңмен `pygame.display` модулінің қолданылатын функциялары мен маңыздылығын талдаңдар.

7

Үй тапсырмасы

«`PyGame.display` функциялар кестесін тағы қандай командалармен толықтыруға болады?» деген сұраққа толық жауап дайындаңдар.

§ 28. PyGame кітапханасының басқа да дайын модульдерімен танысу

Естеріңізге түсіріңдер:

- `pygame.display` қалай қосылады?
- `pygame.display` модулінің негізгі мақсаты қандай?
- `pygame.display` модулінің функцияларын айтыңдар.

Меңгерілетін білім:

- PyGame-нің дайын модульдері;
- `pygame.surface`, `pygame.image` модульдерінің функциясы;
- `pygame.draw` модулінің функциясы және т.б.

Кескін – Изображение – *Image*

Жүктеу – Загрузить – *Load*

Сурет салу – Рисовать – *Draw*

Бұл бөлімде біз ең көп таралған PyGame-де де бар модульдермен танысамыз. Олар: `pygame.surface`, `pygame.image`, `pygame.event`, `pygame.draw`, `pygame.mouse`.

Surface модулі – кескіндерді көрсетуге арналған PyGame нысаны.

`Surface.blit (source, dest, area=None, special_flags = 0)` – бір кескіннің үстіне екіншісін салады. Негізгі қабаттың бетіне (*source*) берілген қабатты сызады, үстіңгі жағына (*Surface*) сурет салады, мұнда *dest* – кортеж (*x, y*), *source* қабаты, *area* – (*width, height*) бастапқы беттің өлшемдері.

`Surface.get_rect()` – көріністі қайтару (*x, y, width, height*), мұнда *x, y* – беттің жоғарғы сол жақ бұрышының координаталары, (*Surface*), *width, height* – өлшеміне сәйкестігі.

Image модулі – кескіндерді жүктеу және сақтау функциялары бар нысан. Кескін *Surface* нысаны ретінде жүктеледі. *Surface* класы оларды басқаруға мүмкіндік береді (сызықтар салу, пиксельдерді орнату, аймақтарды түсіру және т.б.).

`Pygame.image.load ()` – файлдан жаңа кескінді жүктейді.

Event модулі – PyGame нысаны, оқиғалар және кезектермен өрекеттесу үшін қолданылады.

PyGame барлық оқиғаның хабарламаларын оқиғалар кезегі арқылы өңдейді. Осы модульдегі кіші бағдарламалар оқиғалардың кезегін басқаруға көмектеседі.

PyGame-де кез келген оқиға, *мысалы*, пернені басу, Event objects нысандарынан тұратын тізімге орналастырылады. Барлық «нысан оқиғалары» event.type әдісі арқылы қатынасуға болатын түрге ие.

pygame.event.get () – get () әдісін қолданып, кезекпен оқиғаларды алуға болады.

Mouse модулі – PyGame нысаны, тінтуірмен жұмыс істейді.

pygame.mouse.set_pos – тінтуірдегі меңзер қызметін орнатады.

pygame.mouse.set_cursor – тінтуірдің меңзер жүйесі үшін кескін орнатады.

Draw модулі – PyGame фигураларды сызуға арналған нысан.

Осы модуль мен оның функцияларына толық тоқталып өтейік (17-кесте):

17-кесте. Pygame.draw көмегімен негізгі кескіндерді сызу

Қызметтер	Сипаттамалары
pygame.draw.rect	Төртбұрышты салу
pygame.draw.polygon	Көпбұрышты салу
pygame.draw.circle	Шеңбер салу
pygame.draw.ellipse	Эллипс салу
pygame.draw.arc	Эллипстің доғасын салу
pygame.draw.line	Түзу сызықтың қиындысын салу
pygame.draw.lines	Бірнеше іргелес сегменттерді сызу
pygame.draw.aaline	Тегіс сызықтарды сызу
pygame.draw.aalines	Тізбегі тегістеліп сызылған сызықтар

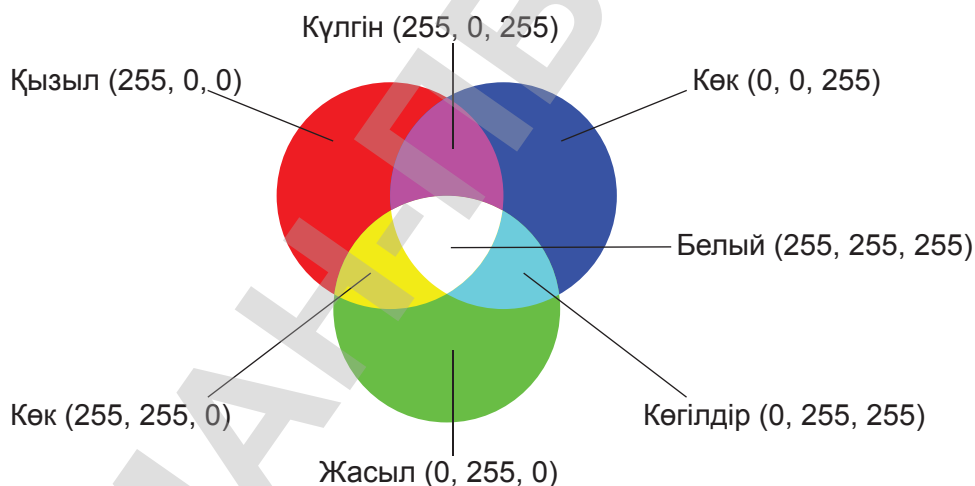
Сурет салуға арналған модульді түстермен толықтыруға болады. Компьютерде түстер қалай пайда болады? Компьютер экраны үш түсті пиксельдерден тұрады: қызыл (Red), жасыл (Green), көк (Blue). RGB – бұл үш негізгі компонентті 0 мен 255 аралығындағы араластыру арқылы барлық түстер мен олардың реңктерін алу әдісін ұсынатын түс үлгісі.

Міне, біз одан әрі қолдана алатын түстер кестесі (18-кесте).

18-кесте. Қолданатын түстер кестесі

Түс атауы	RGB түс форматы	Түстер
BLACK	(0, 0, 0)	
WHITE	(255, 255, 255)	
RED	(255, 0, 0)	
GREEN	(0, 255, 0)	
BLUE	(0, 0, 255)	
CYAN	(0, 255, 255)	
MAGENTA	(255, 0, 255)	
YELLOW	(255, 255, 0)	

8-сызбада түстерді біріктіріп, RGB форматында жаңа түс алу жолы көрсетілген.



8-сызба. RGB түстік моделі

Мысалы, `pygame.draw` сурет функцияларын пайдалануды қарастырайық:

Программа листингі:

```
import pygame
from math import pi
```

```

pygame.init()
# RGB форматындағы түстерді анықтау
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)
GREEN = (0, 255, 0)
RED = (255, 0, 0)

# экранның биіктігі мен енін белгілеу
size = [400, 300]
screen = pygame.display.set_mode(size)

# терезе атауын белгілеу
pygame.display.set_caption("Сурет салу мысалы")

# оқиғаларды өңдеу
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            exit()

    # Экранды тазалау және экранның ақ фонын орнату
    screen.fill(WHITE)

    # Экранға (0,0) – (40,40) нүктелері аралығында,
    қалыңдығы 4 пиксель болатын қара сызық салу
    pygame.draw.line(screen, BLACK, [0,0], [40,40], 4)

    # Экранға қалыңдығы 4 пиксель болатын бірнеше
    аралас бөліктерді салу
    pygame.draw.lines(screen, GREEN, False, [[60,20],
    [100,20], [140,5], [180,20], [220,20]], 4)

    # Экранға (0,50) – (40,90) нүктелері аралығында,
    қалыңдығы 2 пиксель болатын тұтас сызықты салу
    pygame.draw.aaline(screen, BLACK, [0,50],
    [40,90], 2)

    # Ұзындығы мен ені 60 пиксельден тұратын, нүктенің
    қалыңдығы 1 пиксель болатын, (60,60) нүктелері
    аралығындағы тік төртбұрыш салу

```

```

pygame.draw.rect(screen, BLUE, [60,60,60,60], 1)

# Ұзындығы мен ені 60 пиксель болатын, (140,60)
нүктелері аралығында іші толық боялған тік төртбұ-
рышты салу
pygame.draw.rect(screen, BLUE, [140,60,60,60])

# Центрі (90,17) нүктелері аралығында, радиусы 30
пиксель болатын шеңбер салу
pygame.draw.circle(screen, BLACK, [90,170], 30,1)

# Центрі (170,170) нүктелері аралығында, радиусы
30 пиксель болатын іші толық боялған шеңбер салу
pygame.draw.circle(screen, BLACK, [170,170], 30)
#
Тіктөртбұрышты қолдана отырып, эллипсті салу
pygame.draw.ellipse(screen, RED, [60,220,60,30], 1)

# Тіктөртбұрышты қолдана отырып, іші толық боялған
эллипс салу
pygame.draw.ellipse(screen, RED, [140,220,60,30])
# Көпбұрышты салу функциясын қолданып, үшбұрыш салу
pygame.draw.polygon(screen, BLACK, [[170,25],
[140,60], [200,60]], 5)

# Жарық бұрышын өлшеу арқылы эллипс доғасын салу
pygame.draw.arc(screen, RED, [210,75,150,125], 0,
pi/2, 3)
pygame.draw.arc(screen, GREEN, [210,75,150,125],
pi/2, pi, 3)
pygame.draw.arc(screen, BLUE, [210,75,150,125],
pi, 3*pi/2, 3)
pygame.draw.arc(screen, BLACK, [210,75,150, 125],
3*pi/2, 2*pi, 3)

# Функцияларды қолданғаннан кейін, экранды жаңарту
pygame.display.flip()

# Жұмысты аяқтау
pygame.quit()

```

Программа нәтижесі мынадай болды (33-сурет):



33-сурет. Программа нәтижесі

1

Сұрақтарға жауап берейік

1. PyGame кітапханасының қандай дайын модульдері бар?
2. `pygame.surface`, `pygame.image`, `pygame.event`, `pygame.mouse` модулінің функциялары қандай?
3. `pygame.draw` модулінің басты мақсаты қандай?
4. `pygame.draw` қандай функцияларды атқарады?
5. RGB түс форматы қайда қолданылады?

2

Ойланайық, талқылайық

- Берілген код `pygame.draw.line(screen, green, [x0, y0], [x1, y1], 5` нені білдіреді? Сұрақтарға жауап беріңдер.
- `Green` нені білдіреді?
 - `[x0, y0]` және `[x1, y1]` не үшін қажет?
 - `5` саны қандай өлшем бірлігін білдіреді?

3

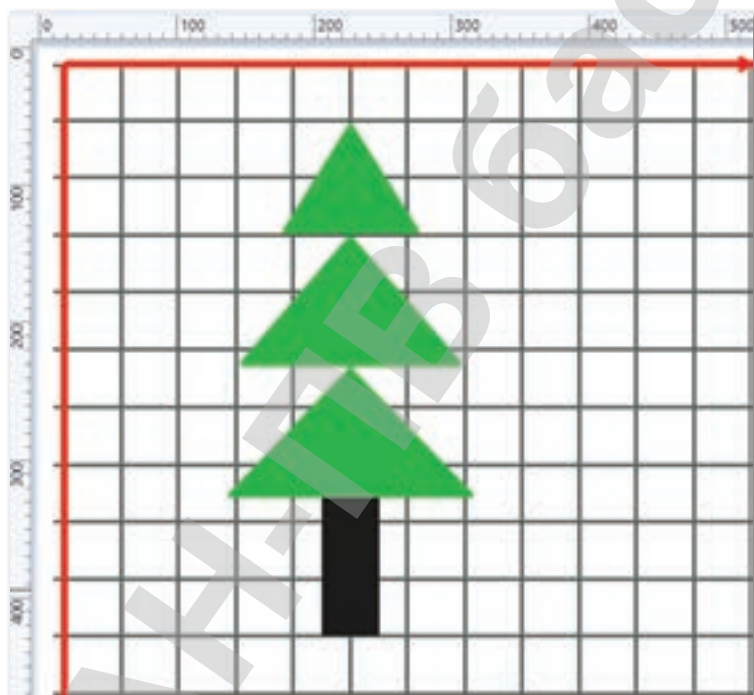
Талдап, салыстырайық

Тік төртбұрыш, шеңбер немесе эллипс сызу барысында сызықтың берілген ені 0-ге тең болса, жауабы қандай болады?

4

Дәптерге орындайық

Компьютерде еркін сурет салу үшін, алдымен мынадай амалдарды істеу керек. Белгілі бір масштабта координаталық жазықтықты дәптерге түсіріп, суреттердің бірнеше моделін (шырша, қоршау және т.б.) салу үшін геометриялық фигураларды пайдаланыңдар.



5

Компьютерде орындайық

1. Берілген программа кодын қолданып, сурет салыңдар.

```
pygame.draw.polygon(screen, GREEN, [[50, 100], [0, 200], [150, 200], [100, 100]], 3)
```

 Сенде қандай фигура пайда болды?

2. Берілген программа кодының үзіндісін қолданып, эллипс салыңдар.

```
pygame.draw.ellipse(screen, BLACK,
[30, 30, 200, 100], 1)
```

Қолданатын тік төртбұрыштың басты координаталарын, ұзындығын және енін көрсетіңдер.

6

Ой бөлісейік

Сабақта жаңа не үйрендіңдер? Сыныптастарыңмен Python-да қосымшалар жасау үшін қандай дайын модульдер қажет екенін талқылыңдар. PyGame графикасымен қалай жұмыс істеуге болады? Сурет модулі функциясының мүмкіндіктері қандай?

7

Үй тапсырмасы

1. draw модулінің функцияларын және RGB түстер моделін пайдаланып, бағдаршамның суретін салыңдар.
2. CD дискіден осы тақырыпқа сәйкес тапсырманы орындаңдар.

§ 29. Артқы фон мен ойын кейіпкерлері

Естеріңізге түсіріңдер:

- PyGame модульдері қандай?
- `pygame.surface`, `pygame.image` модульдерінің қызметі қандай?
- `pygame.draw` модульінің қызметі қандай?

Толтыру – Заполнить – *Fill*

Түс – Цвет – *Color*

Фон – Фон – *Background*

Меңгерілетін білім:

- артқы фонды түспен толтыру;
- кескінді артқы қабатқа жүктеу;
- PyGame-де `time.clock` модулімен жұмыс істеу.

Бұл сабақта ойынның артқы фонын жасауды қарастырайық. Артқы фон жасау үшін фонды кез келген түспен толтыруға немесе суретті жүктеуге болатынын білеміз. Мысалы, фонды қандай да бір түске толтыру үшін мынадай функция қолданылады:

```
pygame.Surface.fill(color, rect=None, special_flags = 0)
```

Функция бетті тұтас түспен толтыруға мүмкіндік береді. Егер `rect` үшін тікелей аргумент орнатылмаса, барлық беті толтырылады. `Rect` аргументі белгілі бір аймаққа толтыруды шектейді. Толтыру, сондай-ақ бетті кесу аймағында болады. `Color` аргументі RGB, RGBA дәйекті немесе түс индексі болуы мүмкін.

PyGame-нің нысаны `Surface` суреттерді көрсету үшін қолданылатынын еске түсіріңдер.

Суретті артқы фонға жүктеу үлгісі:

```
background_image = 'green.jpg'
import pygame
from pygame.locals import *
from sys import exit
pygame.init()
screen = pygame.display.set_mode((600,500), 0, 32)
pygame.display.set_caption("My First Game")
```

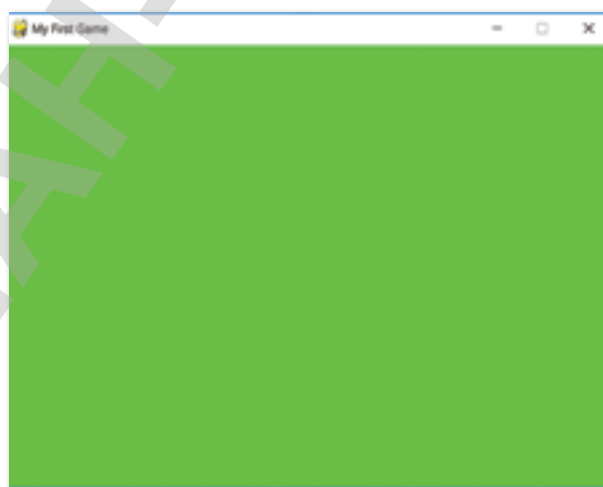
```

background = pygame.image.load(background_image).
convert()
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            exit()
    screen.blit(background, (0,0))
    pygame.display.update()

```

Біздің ойынның артқы фоны біркелкі жасыл түсті болады. Ол үшін өлшемі 600x500-ден кем емес жасыл сурет қажет. Листингте оған `background_image` атауын бердік. Бұл суретті жүктеу үшін `pygame` модулін қолданамыз. `pygame.image` суреттерді тасымалдауға арналған. Атап айтқанда, `pygame.image.load(filename)` файлдан жаңа суретті жүктейді. PyGame автоматты түрде типін анықтайды және деректерден жаңа Surface нысанын жасайды.

Қайтарылатын қабат (бет) ол алынған файл сияқты түс пішімін, түс кілтін және альфа-мөлдірлікті қамтиды. Экранда жылдам сызылатын көшірмені жасау үшін аргументсіз `convert()` функциясы пайда болады. `Convert()` `pygame` түрлендіру үшін қолданылады. Surface бірдей пиксель форматы, `pygame.display.set_mode()` -тан құрылды. Егер сендер оны шақырмасаңдар, онда әр жолы процесті баяулататын пиксельді түрлендіру қажет болады (34-сурет).



34-сурет. Листинг нәтижесі

Ойынның артқы қабатына кез келген суретті орнатуға болады.

Surface нысандары үшін fill () әдісі:

```
# бетіне ақ фонды салу
windowSurface.fill (WHITE)
```

Біз windowSurface сақталған барлық бетті ақ түспен толтырғымыз келеді. Fill () функциясы параметр ретінде берілетін Surface бетін түспен толық жабады.

Fill () әдісін немесе кез келген басқа сурет функциясын шақырғанда, PyGame экранындағы терезе өзгермейтініне назар аударыңдар. Олар Surface нысанын өзгертеді. PyGame функциясы арқылы экранда жаңа Surface нысанын көрсету үшін, display.update () қолдану керек.

Компьютердегі Surface нысанын өзгерту, экрандағы суретті өзгертуден әлдеқайда жылдамырақ болады. Сондықтан суреттің барлық функциялары Surface нысанында салынғаннан кейін ғана экранда сурет салу әлдеқайда тиімді.

Fill () әдісі – функция емес, pygame.Surface нысанының әдісі. Ол түс параметрі үшін берілетін кез келген түс мәнімен барлық Surface нысанын толық толтырады.

Кітапханамен жұмыс істеуді жақсы түсіну үшін төмендегі қарапайым кодты қарастыруды ұсынамыз. Бұл кезеңде жалпы мақсаттар үшін осы кодты іске қосып, алынған нәтижені талдап көрейік. Төменде кодтың жалпы түсіндірмесі берілген.

Программа листингi HelloWorld.py:

```
background_image = 'plate.png'
mouse_image = 'apple.jpg'
import pygame
from pygame.locals import *
from sys import exit
pygame.init()
screen = pygame.display.set_mode((640,480), 0, 32)
pygame.display.set_caption("Hello, World!")
background = pygame.image.load(background_image).
convert()
mouse_cursor = pygame.image.load(mouse_image).
convert_alpha()
while True:
    for event in pygame.event.get():
```

```

if event.type == QUIT:
    screen.blit(background, (0,0))
x, y = pygame.mouse.get_pos()
x-= mouse_cursor.get_width() / 2
y-= mouse_cursor.get_height() / 2
screen.blit(mouse_cursor, (x, y))
pygame.display.update()

```

Программа нәтижесі мынадай болды (35-сурет):



35-сурет. Программа нәтижесі

Берілген мысалды іске қосу үшін бізге екі кескін керек: біреуі фон ретінде, ал екіншісі сурет салудағы тінтуірдің меңзері ретінде. Бұл жағдайда фон – тарелка суреті, тінтуірдің қозғалысынан кейінгі меңзер – алма бейнесі. Листингтің екінші жолы кескінді сипаттайды. Ал 3–6-жолдары `import pygame` импорттайды және `pygame.init()` модулінің қажетті бөлігін баптайды. Инициализациядан кейін терезені құру қажет. 7–8-жолда терезе өлшемі беріледі. Листингтің алғашқы екі жолы сурет деректерін орнатады. 9–10-жолда өлшемі 640x480 болатын терезе жасалады. Терезенің атауы – “Hello, world!”.

Сонымен қатар `load ()` функциясы фонға және тінтуірге кескінді жүктеу үшін пайдаланылады:

```
background = pygame.image.load(background_image).
convert()
mouse_cursor=pygame.image.load(mouse_image).
convert_alpha()
```

Кескінді тінтуір меңзерімен жылжыту үшін бастапқы координаталарын жойып, соңғы деректерді есептейміз:

```
x, y = pygame.mouse.get_pos()
x-= mouse_cursor.get_width() / 2
y-= mouse_cursor.get_height() / 2
screen.blit(mouse_cursor, (x, y))
```

Сондай-ақ `screen.blit ()` функциясын қолданамыз. Соңында жаңа суретті алу үшін экранды жаңартыңдар.

```
pygame.display.update()
```

1

Сұрақтарға жауап берейік

1. Ойынның артқы фонын толтыру қалай жүзеге асырылады?
2. Ойынның фонына кескіндерді жүктеу тәртібін сипаттаңдар.
3. `Fill ()` әдісін пайдаланғанда терезе өлшемі өзгере ме?

2

Ойланайық, талқылайық

1. Неліктен `fill (color)` функция емес, PyGame әдісі болып табылады?
2. `Surface` нысандары үшін `fill ()` әдісінің маңыздылығы неде?

3

Талдап, салыстырайық

Артқы фонды толтыру ерекшеліктерін салыстырып, талдау жасаңдар:

- 1) түспен;
- 2) суретпен.

4

Дәптерге орындайық

Қарастырылған әдістер мен келтірілген кеңестерді дәптерлеріңе жазыңдар.

5

Компьютерде орындайық 

1. `pygame.display` модулінде `set_caption()` функциясы бар. Терезенің тақырыбы ретінде белгіленетін жол беріледі. Терезенің атауын әр секунд сайын өзгертіндей жасаңдар.
2. CD дискіден осы тақырыпқа сәйкес тапсырманы орындаңдар.

6

Ой бөлісейік

Сабақта жаңа не үйрендіңдер? Өз сыныптастарыңмен ойынның артқы фоны үшін суретті жүктеудің маңыздылығы туралы талқылаңдар.

7

Үй тапсырмасы 

Ойынның фоны қызықты болу үшін, тұсқағаз ретінде түрлі кескіндерді қолданыңдар. Фондық тұсқағаздың бірнеше нұсқасын жасаңдар және фон суретін өзгертіндер.

§ 30. Кейіпкерлерді анимациялау. Спрайттарды жасау. Ойынға дайын кейіпкерлерді жүктеу

Естеріңізге түсіріңдер:

- артқы фонды түспен қалай толтыру керек?
- кескінді артқы фонға қалай жүктеуге болады?

Меңгерілетін білім:

- спрайттарды жасау;
- ойынға дайын кейіпкерді жүктеу;
- ойын жасауға арналған программаны жазу.

Класс – Класс – *Class*

Уақыт – Время – *Time*

Көлік – Автомобиль – *Car*

Спрайт – ойындағы графикалық нысандар, көбінесе кейіпкерлер. Әдетте спрайттардың программалық кодтағы графикалық примитивтер сияқты суреті салынбайды, олар алдын ала жасалады (яғни ойынды іске қосу кезіндегі дайын сурет). Оларды бір тұтас экран ретінде манипуляциялауға және жылжытуға болады. Спрайт әртүрлі сипаттары және әдістері бар нысан ретінде қарастырылады (*мысалы*: ені, биіктігі, түсі, т.б.).

Ойынның басты класынан бастайық (оны Game деп атаймыз), ол басқа кодты басқарады. Game класында `__init__` ойынды бастау функциясы, сондай-ақ ойын анимациясының басты циклі болады.

Мысал үшін қарапайым жарыс ойынын көрейік.

Ойыншы көлікті бақылап, солға және оңға қарай жылжуы мүмкін. Сондай-ақ жоғары және төмен пернелердің көмегімен жылдамырақ және баяу жүре алады.

`Car.py` класын құрыңдар. Бұл класс *көліктің* сипаттамасын ұсынады. Ол PyGame-дегі Sprite класынан шыққан. Көлікті біз тік төртбұрыш ретінде саламыз. `image.load()` функциясын пайдалана отырып, көліктің суретін жүктеуге де болады.

```
import pygame

WHITE = (255, 255, 255)
class Car(pygame.sprite.Sprite):
    def __init__(self, color, width, height):
        super().__init__()
```

```

self.image = pygame.Surface([width, height])
self.image.fill(WHITE)
self.image.set_colorkey(WHITE)

pygame.draw.rect(self.image, color, [0, 0,
width, height])
self.rect = self.image.get_rect()

```

600x500 өлшемінде терезе жасаңдар. «My First Game» атауын беріңдер. Фондық суретке жасыл фонды жүктеңдер. **20x30** өлшемді мәшинені салып, координаталарды (200; 450) қойыңдар. Бастапқы координаталары (100, 0) белгілеп, **200x450** өлшемді жол сызыңдар. Сондай-ақ жолдың ортасына бөлу жолағын қосыңдар.

myFirstGame.py листингі мен нәтижесі (36-сурет):

```

background_image = 'green.jpg'

import pygame, random
from Car import Car
from pygame.locals import *
from sys import exit
pygame.init()

RED = (255, 0, 0)
GREEN = (20, 255, 140)
GREY = (210, 210, 210)

screen = pygame.display.set_mode((400,500), 0, 32)
pygame.display.set_caption("My First Game")
background = pygame.image.load(background_image).
convert()

cars_list = pygame.sprite.Group()
playerCar = Car(RED, 20, 30)
playerCar.rect.x = 200
playerCar.rect.y = 450

cars_list.add(playerCar)

```

```

check = True
clock=pygame.time.Clock()

while check:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            check=False

    #Экранда саламыз
    screen.fill(GREEN)
    #Жолды саламыз
    pygame.draw.rect(screen, GREY, [100,0,
200,450])
    #Draw Line painting on the road
    pygame.draw.line(screen, WHITE, [200,0],
[200,450], 5)

    cars_list.draw(screen)

    #Экранды жаңарту
    pygame.display.flip()
    clock.tick(60)

pygame.quit()

```



36-сурет. Программа нәтижесі

1

Сұрақтарға жауап берейік

1. Спрайт қайда қолданылады?
2. Фонға кейіпкер қою қалай жүзеге асырылады?
3. PyGame-дегі Sprite класы қандай мақсатта қолданылады?

2

Ойланайық, талқылайық

1. Ойын құру барысында фонға кейіпкерді қоюдың қажеттілігі қандай?
2. Game класында `__init__` ойынды бастау функциясы не үшін маңызды?

3

Талдап, салыстырайық

1. Ойынға спрайт арқылы кейіпкер қосудың тиімділігі неде?
2. Жасалып жатқан ойынға дайын кейіпкер қою жұмысына арналған программаның артықшылықтары мен кемшіліктерін талдаңдар.

4

Дәптерге орындайық

Қарапайым жарыс ойынының алгоритмін дәптерге жазыңдар.

5

Компьютерде орындайық

PyGame анимациясының үлгісін программаға жазыңдар. Нәтижені талдаңдар.

```
import sys, pygame
pygame.init()
size = [600, 500]
speed = [0, 2]
white = (255, 255, 255)

screen = pygame.display.set_mode(size)

ball = pygame.image.load("ball.bmp")
ballrect = ball.get_rect()
while 1:
    for event in pygame.event.get():
        if event.type == pygame.QUIT: sys.
exit()
```

```

    ballrect = ballrect.move(speed)
    if ballrect.left < 0 or ballrect.right >
width:
        speed[0] = -speed[0]
    if ballrect.top < 0 or ballrect.bottom > height:
        speed[1] = -speed[1]

    screen.fill(white)
    screen.blit(ball, ballrect)
    pygame.display.flip()

```

6

Ой бөлісейік

Сабақта не үйрендіңдер? Сыныптастарыңмен жаңа сабақта өткендеріңді күнделікті өмірде қалай қолдануға болатынын талқылаңдар. Мысал келтіріңдер.

7

Үй тапсырмасы

`pygame.time.clock` арқылы түстерін өзгертіп тұратын бағдаршам жасаңдар.

§ 31–32. Спрайттарды пернетақтамен қозғалту және басқару

Естеріңізге түсіріңдер:

- *спрайттарды қалай пайдалануға болады?*
- *ойынға кейіпкерді қою дегеніміз не?*
- *PyGame-де Sprite класы қандай жағдайға арналған?*

Меңгерілетін білім:

- *пернені басу оқиғасы арқылы спрайттарды басқару;*
- *moveRight әдісі;*
- *спрайттың басқа спрайттармен кездесуінің алдын алу.*

Ойын – Игра – *Game*

Жылжыту – Переместить – *Move*

Пернетақта – Клавиатура – *Keyboard*

Көлікті солға, оңға, алға және артқа жылжыту үшін бізге Car класына әдістерді қосу керек. Содан кейін пернелерді басу оқиғасына жауап қату үшін программаның негізгі цикліне event handlers-ті қосамыз. Ойыншы пернетақтада көрсеткіш пернелерін пайдаланған кезде, біз көлікті экранда жылжыту үшін өзіміздің әдістерді шақыратын боламыз.

MoveRight () әдісі оңға жылжуға жауап береді және екі аргумент қабылдайды. Біріншісі белгісіз және self деп аталады. Бұл ағымдағы нысанға жатады. Екіншісі pixels деп аталады және көлікті жылжыту үшін пайдаланатын пиксельдер санына қатысты. Нысанның ағымдағы x координатасына пиксельдерді қосу:

```
def moveRight(self, pixels):
    self.rect.x += pixels
```

Сол принцип бойынша *moveLeft* () әдісін қосамыз:

```
def moveRight(self, pixels):
    self.rect.x -= pixels
```

Басты циклді жаңартамыз:

```
while check:
    for event in pygame.event.get():
```

```

if event.type == pygame.QUIT:
    check = False
elif event.type == pygame.KEYDOWN:
    if event.key == pygame.K_x:
        check = False

keys = pygame.key.get_pressed()
if keys[pygame.K_LEFT]:
    playerCar.moveLeft(5)
if keys[pygame.K_RIGHT]:
    playerCar.moveRight(5)

```

Перне ОҢҒА және СОЛҒА қарай басылғанда, мәшине 5 пиксельге x осі бойымен жылжиды.

Көлік жарыстары үшін біз тағы бірнеше көлікті қосамыз. Әрбір көлік – мәшинелердің класынан жасалатын нысан.

Car класына бірнеше қасиет пен әдістерді қосамыз. Бұл келесі мәшинелердің түрлі өлшемдеріне (ені мен биіктігіне), түстерге және әртүрлі жылдамдықтарға ие болуын қамтамасыз ету үшін жасалады.

Содан кейін негізгі программада төрт қосымша көлік жасап, оларды `all_comming_cars` деп аталатын тізімге қосамыз. Әрбір мәшинеге өз түсі, жылдамдығы және басқа бастапқы жағдайы (x және y координаталары) тағайындалады.

Ойынның жалпы жылдамдығын `speed` деп аталатын айнымалы сақтайды. Жаңа оқиғаларды жасаушылар пайдаланушының пернетақтаның **ЖОҒАРЫ** немесе **ТӨМЕН** пернесін қашан басатындығын анықтайды. Олар айнымалы `speed`-ті 0,05-ке ұлғайтады немесе азайтады.

Программа листингі:

```

import pygame, random
#Let's import the Car Class
from Car_new import Car
pygame.init()

GREEN = (20, 255, 140)
GREY = (210, 210, 210)
WHITE = (255, 255, 255)
RED = (255, 0, 0)

```

```
MAGENTA = (255, 0, 255)
YELLOW = (255, 255, 0)
CYAN = (0, 255, 255)
BLUE = (100, 100, 255)
speed = 1
colorList = (RED, GREEN, MAGENTA, YELLOW, CYAN, BLUE)
SCREENWIDTH = 600
SCREENHEIGHT = 600
size = (SCREENWIDTH, SCREENHEIGHT)
screen = pygame.display.set_mode(size)
pygame.display.set_caption("My First Game")
cars_list = pygame.sprite.Group()
playerCar = Car(RED, 60, 80, 70)
playerCar.rect.x = 160
playerCar.rect.y = SCREENHEIGHT - 100
car1 = Car(PURPLE, 60, 80, random.randint(50,100))
car1.rect.x = 60
car1.rect.y = -100
car2 = Car(YELLOW, 60, 80, random.randint(50,100))
car2.rect.x = 160
car2.rect.y = -600
car3 = Car(CYAN, 60, 80, random.randint(50,100))
car3.rect.x = 260
car3.rect.y = -300
car4 = Car(BLUE, 60, 80, random.randint(50,100))
car4.rect.x = 360
car4.rect.y = -900
cars_list.add(playerCar)
cars_list.add(car1)
cars_list.add(car2)
cars_list.add(car3)
cars_list.add(car4)
all_coming_cars = pygame.sprite.Group()
all_coming_cars.add(car1)
all_coming_cars.add(car2)
all_coming_cars.add(car3)
all_coming_cars.add(car4)
check = True
clock = pygame.time.Clock()
```

```

while check:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            check = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_x:
                playerCar.moveRight(10)
    keys = pygame.key.get_pressed()
    if keys[pygame.K_LEFT]:
        playerCar.moveLeft(5)
    if keys[pygame.K_RIGHT]:
        playerCar.moveRight(5)
    if keys[pygame.K_UP]:
        speed += 0.05
    if keys[pygame.K_DOWN]:
        speed -= 0.05
    for car in all_coming_cars:
        car.moveForward(speed)
        if car.rect.y > SCREENHEIGHT:
            car.changeSpeed(random.
randint(50,100))
            car.repaint(random.choice(colorList))
            car.rect.y = -200
    car_collision_list = pygame.sprite.
spritecollide(playerCar, all_coming_cars, False)
    for car in car_collision_list:
        print("Car crash!")
        check=False
    cars_list.update()
    screen.fill(GREEN)
    pygame.draw.rect(screen, GREY, [40, 0, 400,
SCREENHEIGHT])
    pygame.draw.line(screen, WHITE, [140, 0],
[140, SCREENHEIGHT], 5)
    pygame.draw.line(screen, WHITE, [240, 0],
[240, SCREENHEIGHT], 5)
    pygame.draw.line(screen, WHITE, [340, 0],
[340, SCREENHEIGHT], 5)
    cars_list.draw(screen)

```

```
pygame.display.flip()
clock.tick(60)

pygame.quit()

Car_new.py
import pygame
WHITE = (255, 255, 255)

class Car(pygame.sprite.Sprite):

    def __init__(self, color, width, height,
speed):

        super().__init__()

        self.image = pygame.Surface([width, height])
        self.image.fill(WHITE)
        self.image.set_colorkey(WHITE)
        self.width=width
        self.height=height
        self.color = color
        self.speed = speed

        pygame.draw.rect(self.image, self.color,
[0, 0, self.width, self.height])

        self.rect = self.image.get_rect()

    def moveRight(self, pixels):
        self.rect.x += pixels

    def moveLeft(self, pixels):
        self.rect.x -= pixels

    def moveForward(self, speed):
        self.rect.y += self.speed * speed / 20

    def moveBackward(self, speed):
        self.rect.y -= self.speed * speed / 20
```

```

def changeSpeed(self, speed):
    self.speed = speed

def repaint(self, color):
    self.color = color
    pygame.draw.rect(self.image, self.color, [0,
0, self.width, self.height])

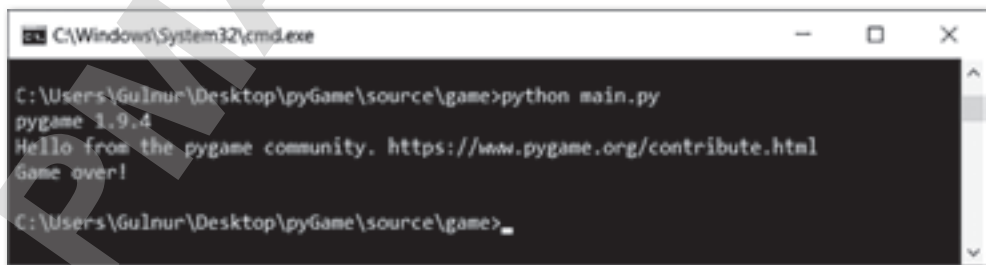
```

Нәтижесі былай болады (37-сурет):



37-сурет. Программа нәтижесі

Келесі қосымша екіөлшемді ойындардың көпшілігінде қолданылады: мақсат – спрайттың басқа спрайттармен соқтығысатынын тексеру (бұл жағдайда `all_coming_cars` тізімінде сақталған мәшинелер). Соқтығысқан жағдайда, ойын аяқталғаны туралы жазба көрсетіледі (38-сурет).



38-сурет. Программа нәтижесі

1

Сұрақтарға жауап берейік

1. Пернені басу оқиғасы арқылы спрайттарды басқару дегеніміз не?
2. Автокөлікті солға, оңға, алға және артқа жылжытуға арналған программаларды таңдауда нені ескеру керек?
3. Спрайттар қозғалысының ерекшелігі неде?

2

Ойланайық, талқылайық

1. `event handlers` оқиға өңдеуіштерін қосу қаншалықты маңызды?
2. Спрайттың басқа спрайттармен соқтығысуы ойынға қаншалықты әсер етеді?

3

Талдап, салыстырайық

Нысанды пернетақтаны пайдалану арқылы басқару мен қозғалтуда, оның экрандағы ойын алаңынан тыс жерлерде қалай әрекет ететінін талдаңдар.

4

Дәптерге орындайық

Көлікті экранда жылжыту үшін шақыратын әдістерді дәптерге жазыңдар.



5

Компьютерде орындайық

Берілген командаларды орындайтын программа құрындар:

1. Экрандағы нысанды бейнелейтін кем дегенде екі түрлі элемент жасаңдар. Бұрын салған графикалық примитивтерді қайталамаңдар. Кеме, күн немесе үй т.б. салыңдар. Алдыңғы сабақтарда өз беттеріңмен салған суреттерің болса, соны пайдалана аласыңдар.
2. Пернетақтаның көмегімен графиканы жылжыту туралы айттық. Экранда екі түрлі элементті басқару үшін осы әдісті пайдаланыңдар.

6

Ой бөлісейік

Сабақта қандай оқиғаларды өңдеушілерімен таныс болдыңдар, өз сыныптастарыңмен талқылаңдар. Пернетақтың көмегімен дұрыс емес басқару кезінде қандай қақтығыстар болуы мүмкін?

7

Үй тапсырмасы

Пернетақтаның көмегімен спрайттардың қозғалысы және оларды басқару бойынша білімдеріңді қолданып, ойын құрып үйреніңдер.

§ 33–34. Практикум. Шарттарды программалау. Дайын сценарий бойынша ойын әзірлеу

PyGame-дегі кез келген ойын негізгі программа цикліне негізделген. Басты цикл құрылымы:

- Басы (ойынды инициализациялау ойын басталған кезде бір рет іске қосылады):
 - PyGame кітапханасын импорттау және инициализациялау;
 - мән беру және жаңа терезені ашу;
 - ойынның кейіпкерлерін (спрайттарын) инициализациялау.
- While негізгі циклі (шарт қанағаттандырғанда қайталанатын):
 - пайдаланушымен өзара әрекеттесу (пернетақта, тінтуір және т.б.);
 - ойын логикасы;
 - экранды жаңарту (кейіпкерлер мен фонды қайта салу).
- Соңы.

Осы сценариймен ойынды жазамыз: терезенің жоғарғы жағында бірнеше қатар кірпіш бар.

Ойынның мақсаты – допты басқара отырып, барлық кірпішті жою. Әрбір жойылған кірпіш үшін ұпай есептеледі. Ойыншының үш өмірі бар. Доп түскен кезде, ойыншы бір өмірін жоғалтады. [Алгоритм: <http://codeNtronic.com>]. Толық тізім төменде берілген.

```
import sys
import pygame
SCREEN_SIZE = 640, 480

# Нысандардың өлшемдері
BRICK_WIDTH = 60
BRICK_HEIGHT = 15
PADDLE_WIDTH = 60
PADDLE_HEIGHT = 12
BALL_DIAMETER = 16
BALL_RADIUS = BALL_DIAMETER // 2
```

```

MAX_PADDLE_X = SCREEN_SIZE[0] - PADDLE_WIDTH
MAX_BALL_X = SCREEN_SIZE[0] - BALL_DIAMETER
MAX_BALL_Y = SCREEN_SIZE[1] - BALL_DIAMETER
PADDLE_Y = SCREEN_SIZE[1] - PADDLE_HEIGHT - 10
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)
BRICK_COLOR = (200, 200, 0)

# ойын жағдайы
STATE_BALL_IN_PADDLE = 0
STATE_PLAYING = 1
STATE_WON = 2
STATE_GAME_OVER = 3

class Bricka:
    def __init__(self):
        pygame.init()

        self.screen = pygame.display.set_mode(SCREEN_
SIZE)
        pygame.display.set_caption("IGRA")

        self.clock = pygame.time.Clock()

        if pygame.font:
            self.font = pygame.font.Font(None, 30)
        else:
            self.font = None

        self.init_game()

    def init_game(self):
        self.lives = 3
        self.score = 0
        self.state = STATE_BALL_IN_PADDLE
        self.paddle = pygame.Rect(300, PADDLE_Y, PADDLE_
WIDTH, PADDLE_HEIGHT)

```

```

        self.ball = pygame.Rect(300, PADDLE_Y - BALL_
DIAMETER, BALL_DIAMETER, BALL_DIAMETER)

        self.ball_vel = [5, -5]
        self.create_bricks()

    def create_bricks(self):
        y_ofs = 35
        self.bricks = []
        for i in range(7):
            x_ofs = 35

            for j in range(8):
                self.bricks.append(pygame.Rect(x_ofs, y_ofs, BRICK_
WIDTH, BRICK_HEIGHT))
                x_ofs += BRICK_WIDTH + 10
                y_ofs += BRICK_HEIGHT + 5

    def draw_bricks(self):
        for brick in self.bricks:
            pygame.draw.rect(self.screen, BRICK_COLOR,
brick)

    def check_input(self):
        keys = pygame.key.get_pressed()

        if keys[pygame.K_LEFT]:
            self.paddle.left -= 5
            if self.paddle.left < 0:
                self.paddle.left = 0

        if keys[pygame.K_RIGHT]:
            self.paddle.left += 5
            if self.paddle.left > MAX_PADDLE_X:
                self.paddle.left = MAX_PADDLE_X

        if keys[pygame.K_SPACE] and self.state ==
STATE_BALL_IN_PADDLE:
            self.ball_vel = [5, -5]
            self.state = STATE_PLAYING

```

```

        elif keys[pygame.K_RETURN] and (self.state ==
STATE_GAME_OVER or self.state == STATE_WON):
            self.init_game()

    def move_ball(self):
        self.ball.left += self.ball_vel[0]
        self.ball.top += self.ball_vel[1]

        if self.ball.left <= 0:
            self.ball.left = 0
            self.ball_vel[0] = -self.ball_vel[0]
        elif self.ball.left >= MAX_BALL_X:
            self.ball.left = MAX_BALL_X
            self.ball_vel[0] = -self.ball_vel[0]

        if self.ball.top < 0:
            self.ball.top = 0
            self.ball_vel[1] = -self.ball_vel[1]
        elif self.ball.top >= MAX_BALL_Y:
            self.ball.top = MAX_BALL_Y
            self.ball_vel[1] = -self.ball_vel[1]

    def handle_collisions(self):
        for brick in self.bricks:
            if self.ball.colliderect(brick):
                self.score += 3
                self.ball_vel[1] = -self.ball_vel[1]
                self.bricks.remove(brick)
                break

        if len(self.bricks) == 0:
            self.state = STATE_WON

        if self.ball.colliderect(self.paddle):
            self.ball.top = PADDLE_Y - BALL_DIAMETER
            self.ball_vel[1] = -self.ball_vel[1]
        elif self.ball.top > self.paddle.top:
            self.lives -= 1
            if self.lives > 0:
                self.state = STATE_BALL_IN_PADDLE

```

```

        else:
            self.state = STATE_GAME_OVER
def show_stats(self):
    if self.font:
        font_surface = self.font.render("SCORE: " +
str(self.score) + "LIVES: " + str(self.lives), False,
WHITE)
        self.screen.blit(font_surface, (205,5))

def show_message(self,message):
    if self.font:
        size = self.font.size(message)
        font_surface = self.font.render(message,False,
WHITE)
        x = (SCREEN_SIZE[0] - size[0]) / 2
        y = (SCREEN_SIZE[1] - size[1]) / 2
        self.screen.blit(font_surface, (x,y))
def run(self):
    while 1:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                sys.exit()

        self.clock.tick(50)
        self.screen.fill(BLACK)
        self.check_input()

        if self.state == STATE_PLAYING:
            self.move_ball()
            self.handle_collisions()
        elif self.state == STATE BALL IN PADDLE:
            self.ball.left = self.paddle.left + self.
paddle.width / 2
            self.ball.top = self.paddle.top - self.ball.
height
            self.show_message("PRESS SPACE TO LAUNCH
THE BALL")
        elif self.state == STATE_GAME_OVER:
            self.show_message("GAME OVER. PRESS ENTER
TO PLAY AGAIN")

```

```

elif self.state == STATE_WON:
    self.show_message("YOU WON! PRESS ENTER TO
PLAY AGAIN")
    self.draw_bricks()

    pygame.draw.rect(self.screen, BLUE,
self.paddle)

    pygame.draw.circle(self.screen, WHITE,
(self.ball.left + BALL_RADIUS, self.ball.top + BALL_
RADIUS), BALL_RADIUS)

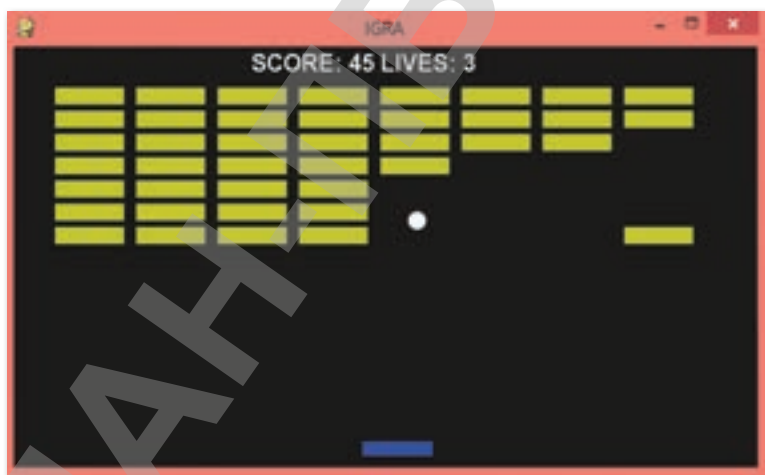
    self.show_stats()

    pygame.display.flip()

if __name__ == "__main__":
    Bricka().run()

```

Нәтижесі былай көрінеді (39-сурет):



39-сурет. Программа нәтижесі

Ұпайларды есептеу алгоритмін іске асыру үшін `score=0` айнымалы енгізіледі, сондай-ақ `lives = 3` айнымалысы бар, ол қалған өмір санына жауап береді. `State` айнымалысы ойын қандай күйде екенін көрсетеді. Барлығы 4 күй анықталды:

```

STATE_BALL_IN_PADDLE = 0 – тақтадағы доп
STATE_PLAYING = 1 – ойын жүріп жатыр
STATE_WON = 2 – ұтыс
STATE_GAME_OVER = 3 – ойын соңы
self.lives = 3
self.score = 0
self.state = STATE_BALL_IN_PADDLE

```

Бұл функция ұпайларды есептейді. Егер доп кірпішке түссе, ұпайдың жалпы санына 3 ұпай қосылады. Ал кірпіш ойын алаңынан жойылады.

Егер экрандағы кірпіштің саны 0 болса, онда ойынның күйі «жеңімпаз» жағдайына ауысады және ойыншы жеңді деп жарияланады.

Доптың тақтадан құлауы жағдайында өмірдің саны тексеріледі. Lives айнымалысы нөлден көп болса, ойын «тақтадағы доп» күйіне ауысады және ойын жаңартылады. Егер өмір болмаса, «ойын соңы» жарияланады.

```

def handle_collisions(self):
    for brick in self.bricks:
        if self.ball.colliderect(brick):
            self.score += 3
            self.ball_vel[1] = -self.ball_vel[1]
            self.bricks.remove(brick)
            break
    if len(self.bricks) == 0:
        self.state = STATE_WON
if self.ball.colliderect(self.paddle):
    self.ball.top = PADDLE_Y - BALL_DIAMETER
    self.ball_vel[1] = -self.ball_vel[1]
elif self.ball.top > self.paddle.top:
    self.lives -= 1
    if self.lives > 0:
        self.state = STATE_BALL_IN_PADDLE
    else:
        self.state = STATE_GAME_OVER

```

5-БӨЛІМДІ ҚОРЫТЫНДЫЛАУҒА АРНАЛҒАН ТЕСТ ТАПСЫРМАЛАРЫ

1. Дыбыстарды жүктеу және ойнату:

- A) pygame.mixer.
- B) pygame.cdrom.
- C) pygame.image.
- D) pygame.display.
- E) pygame.key.

2. Фигураларды, сызықтарды және нүктелерді сызу:

- A) pygame.draw.
- B) pygame.cdrom.
- C) pygame.image.
- D) pygame.display.
- E) pygame.key.

3. Сыртқы оқиғаларды басқару:

- A) pygame.event.
- B) pygame.cdrom.
- C) pygame.image.
- D) pygame.display.
- E) pygame.key.

4. Суреттерді өзгерту және жылжыту:

- A) pygame.transform.
- B) pygame.cdrom.
- C) pygame.image.
- D) pygame.display.
- E) pygame.key.

5. Уақыт пен кадр жиілігін басқару:

- A) pygame.time.
- B) pygame.cdrom.
- C) pygame.image.
- D) pygame.display.
- E) pygame.key.

6. pygame.rect -

- A) төртбұрышты аймақтарды басқару.
- B) меңзер бейнелерін жүктеу.
- C) пернетақтада басқан пернені оқу.
- D) суреттер мен экранды басқару.
- E) фигураларды, сызықтарды және нүктелерді сызу.

7. pygame.surface -

- A) суреттер мен экранды басқару.
- B) меңзер бейнелерін жүктеу.
- C) пернетақтада басқан пернені оқу.
- D) суреттер мен экранды басқару.
- E) фигураларды, сызықтарды және нүктелерді сызу.

8. Бейнені көрсету үшін терезені немесе экранды инициализациялау:

- A) pygame.display.set_mode.
- B) pygame.display.get_surface.
- C) pygame.display.set_caption.
- D) pygame.display.get_caption.
- E) pygame.display.quit.

9. Ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу:

- A) pygame.display.set_mode.
- B) pygame.display.get_surface.
- C) pygame.display.set_caption.
- D) pygame.display.get_caption.
- E) pygame.display.quit.

10. Толық экран мен терезе дисплей арасындағы ауысу:

- A) pygame.display.toggle_fullscreen.
- B) pygame.display.get_surface.
- C) pygame.display.set_caption.
- D) pygame.display.get_caption.
- E) pygame.display.quit.

11. Терезе атауын тағайындау:

- A) pygame.display.toggle_fullscreen.
- B) pygame.display.get_surface.

- C) `pygame.display.set_caption`.
- D) `pygame.display.get_caption`.
- E) `pygame.display.quit`.

12. `pygame.display.get_caption` –

- A) дисплей модулін баптау.
- B) дисплей бетін толығымен жаңарту.
- C) ағымдағы терезенің тақырыбын қайтару.
- D) ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу.
- E) терезе атауын тағайындау.

13. `pygame.display.get_surface` –

- A) дисплей модулін баптау.
- B) дисплей бетін толығымен жаңарту.
- C) ағымдағы терезенің тақырыбын қайтару.
- D) ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу.
- E) терезе атауын тағайындау.

14. `pygame.display.flip` –

- A) дисплей модулін баптау.
- B) дисплей бетін толығымен жаңарту.
- C) ағымдағы терезенің тақырыбын қайтару.
- D) ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу.
- E) терезе атауын тағайындау.

15. Спрайт –

- A) дисплей модулін баптау.
- B) дисплей бетін толығымен жаңарту.
- C) ағымдағы терезенің тақырыбын қайтару.
- D) ағымдағы уақытта орнатылған дисплей бетінен сілтеме алу.
- E) терезе атауын тағайындау.

Глоссарий

Аккаунт – сайтқа кіру үшін пайдаланушының жеке ақпараты сақталатын есептік жазба.

Ақпарат – қоршаған ортадағы нысандар мен құбылыстардың параметрлері, қасиеттері, жай-күйі туралы білімді кеңейтетін деректер жиыны.

Ақпараттың дәлдігі – қажетті дәрежедегі нақты нысанға байланысты ақпараттың сәйкестік дәрежесін сипаттайтын қасиет.

Аналық тақша – барлық негізгі бөлшектер мен тораптар орнатылған компьютердің басты микросызбасы.

Аралас бұлттар (hybrid cloud) – екі немесе одан көп бұлт түрлерінің (жеке, ортақ, қоғамдық) аралас комбинациясы.

Биоритмдер – тірі ағзадағы биологиялық процестер мен құбылыстардың сипаты мен қарқындылығының жиі (периодты) қайталанатын өзгерістері.

Бұлттық технологиялар – ыңғайлы интерфейсті пайдалана отырып тұтынушыға Интернет немесе жергілікті желі арқылы қызмет түрінде ұсынылатын, белгілі ресурстармен қашықтан қатынас құру мүмкіндігін беретін программалық-аппараттық жабдық.

Бірөлшемді массив – бір атауға біріктірілген көптеген бір типтес элементтерден тұратын жиын.

Виртуалды мәшине – компьютердің құрамдас бөліктерінің жұмысы үшін қолданатын программа, нәтижесінде кез келген операциялық жүйені орнатуға және іске қосуға мүмкіндік беретін виртуалды орта құрылады.

Деректер базасы (ДБ) – компьютердің көмегі арқылы сақтауға және өңдеуге арналған белгілі бір тақырыптық аймақтағы құбылыстар мен нысандар туралы деректердің реттелген жиынтығы.

Дыбыс картасы – дыбыстық деректерді динамик жүйесі арқылы пайдаланушыға беру үшін түрлендіруді қамтамасыз ететін құрылғы.

Есептер – сәйкес түрде безендірілген деректер жинағының баспадан шығаруы.

Жедел жадының (ЖЖҚ немесе RAM) қажетті көлемі компьютердің шешетін міндеттеріне байланысты. Көптеген

жағдайларда ДК-де ыңғайлы жұмыс істеу үшін 4 Гбайт жадымен қамтамасыз етілуі жеткілікті.

Жеке бұлттар (private cloud) – жеке кәсіпорындарының өзіне ғана, сондағы жеке тұлғалар мен олардың тұтынушыларының жұмыс істеуіне арналған инфрақұрылым.

Жүйелік программалық жасақтама – компьютерлік жүйенің жұмыс істеуін жалпы басқаруды және оның техникалық қызмет көрсету функцияларын орындауды қамтамасыз ететін программалар жиынтығы.

Инициализация – нысанның бастапқы күйін орнату.

Кадр – анимацияны құрайтын суреттердің бірі.

Кескін – компьютер экранындағы сурет.

Конструктор режимі – нысанның құрылымын құру немесе нысан құрылымын редакциялау режимі.

Конфигурация – компоненттердің мақсатына, санына және негізгі сипаттамаларына негізделген белгілі бір құрамдас бөлшектердің жиынтығы.

Кілттік өріс – кестедегі әрбір жазбаны бірегей түрде анықтайтын өріс.

Кітапхана – функция кодтары мен кластар жиынтығы.

Қатқыл диск – бірнеше өзара байланысты дискілерден тұратын аккумулятор.

Қоғамдық бұлттар (community cloud) – қоғамдық тұтынушыларға арналған ортақ мақсаттары бар инфрақұрылым.

Қолданбалы программалық жасақтама – компьютерді пайдаланушыларға арналған белгілі бір мәселелерді шешуге көмектесетін программалар жиынтығы.

Макростар – деректер базасында деректерді өңдеу үшін командалардың стандартты тізбектерін іске қосуға арналған нысандар.

Массив – бір атаумен біріктірілген, бір типтегі деректерден тұратын айнымалылардың жиынтығы.

Модель – нысанның, құбылыстың немесе процестің маңызды ерекшеліктерін көрсететін белгілі бір жаңа нұсқа.

Модельдеу – практикалық мәселелерді шешу үшін модельдерді құру және пайдалану процесі.

Модульдер – Visual Basic кіріктірілген программалау тілінде жазылған программалары бар нысандар.

Мөлдірлік – экранда шығарылмайтын кескін бөлігі, сондықтан артқы жағындағы графикалық нысандарды жасырмайды.

Оқиға – программаны орындау кезінде орын алатын жайт. Мысалы, тінтуірдің жылжу оқиғасы, түймені басу немесе пернетақтадан енгізу.

Ортақ бұлттар (public cloud) – көпшіліктің Интернетте еркін жұмыс істеуіне арналған инфрақұрылым.

Пиксель – компьютерде бейнені көрсетуге қабілетті экрандағы ең кіші нүкте.

Пішіндер – деректер базасындағы жазбаларды енгізу, көрсету және өңдеу үшін жауап беретін нысан.

Серверлік программалық жасақтама – ақпараттық технологияларда белгілі бір ресурстарға немесе қызметтерге қол жеткізе отырып, тапсырыс берушінің сұрауы бойынша сервистік функцияларды орындайтын есептеу жүйесінің программалық компоненті.

Соқтығыс – компьютерлік ойындарда, ойынның бір сипаты басқа кейіпкермен немесе кейбір экран нысанымен соқтығысатын жағдай.

Спрайт – кейіпкер немесе компьютерлік ойындағы графикалық нысан.

Сұраныстар – бір немесе бірнеше кестеден деректерді таңдау үшін шарттар орнатылатын нысан.

Сұрыптау – деректерді белгілі бір тәртіпте орналастыру (өсу немесе кему ретімен).

Сүзгілер – шартты немесе шарттар жиынтығын қанағаттандыратын жазбаларды таңдау.

Ұтқырлық – программалық жасақтаманы бір ортадан (ортаға) екіншіге, атап айтқанда, бір компьютерден екіншісіне ауыстыру мүмкіндігі.

Файлдарды бұлттық сақтау орны – бұлттық сақтау қызметін көрсететін, жеткізуші жабдықтарында орналасқан, Интернет желісіндегі қолжетімді ақпараттарды сақтайтын орын.

Флейм – әңгімеге қатысушылардың пікірін ескермей, жиі айтылған эмоционалдық ескертулер.

Форма – деректер базасындағы жазбаларды енгізуді, қарауды және редакциялауды жеңілдететін нысан.

Этикет – тәртіптің негізгі нұсқауы және қоғамдағы тәртіп.

Vox.net – дербес аккаунттар мен кәсіпорындарға арналған файлдарды сақтау мен орналастыруды қамтамасыз ететін бұлттық сервис.

CPU-Z (Central Processing Unit – Z) – қатқыл дискіден басқа, орталық процессордың, аналық тақшаның және BIOS, жедел жадының, бейнекартаның техникалық сипаттамасын анықтайтын программа.

Dropbox – компьютерді Интернетке қосқан кезде бірлесіп жұмыс істеу мүмкіндігі бар файлдарды сақтауға арналған сервис.

Event модулі – оқиғалармен және кезектермен әрекеттесу үшін қолданылатын PyGame нысаны.

EVEREST Ultimate Edition – компьютердің аппараттық және программалық құралдарының жұмысының тиімділігін диагностикалау, тестілеу және баптауға арналған программа.

GoogleDocs – пайдаланушының құжаттармен тегін жұмыс істеуіне, Google қосымшаларын ыңғайлы және қарапайым пайдалануына арналған онлайн-кеңсе.

Image модулі – кескіндерді жүктеу және сақтау функциялары бар нысан.

Mouse модулі – тінтуірмен жұмыс істейтін PyGame нысаны.

Skype – бүкіл әлем бойынша әріптестеріңмен, достарыңмен, туыстарыңмен Интернет арқылы сөйлесуге мүмкіндік беретін программа.

Пайдаланылған әдебиеттер

1. Безручко В.Т. Информатика курс лекций: Учебное пособие. – М.: ИД. «Форум»: ИНФРА – М. 2013. – 432 с.
2. Семакин И. Информатика. Базовый курс. 7–9 классы / И. Семакин, Л. Залогова, С. Русаков. – М.: Бином. Лаборатория знаний, 2015. – 390 с.
3. Хахаев И.А. – Практикум по алгоритмизации и программированию на Python 2014. – 240 с.
4. Соболев Б.В. и др. Информатика. Учебник. / 3-е изд., доп. и перераб. – Ростов н/Д: Феникс, 2007. – 446 с.
5. Казиев В.М. Информатика в примерах и задачах: кн. для уч-ся 10–11 кл. / В. М. Казиев. – М.: Просвещение, 2007. – 304 с.: ил. – (Профильная школа). – ISBN 978-5-09-016550-1.
6. Информатика: программы для общеобразоват. учреждений: 2–11 кл. / сост. М.Н. Бородин. – 5-е изд., испр. – М.: БИНОМ. Лаборатория знаний, 2008. 463 с. – ISBN 978-5-94774-826-0.
7. Босова Л.Л., Босова А.Ю., Информатика: учебник для 9 класса. М.: БИНОМ. Лаборатория знаний, 2017. – 208 с.
8. Златопольский Д.М. Основы программирования на языке Python. М.: ДМК Пресс, 2017. – 284 с.: ил. – ISBN 978-5-97060-552-3.

Электрондық қорлар

1. www.intuit.ru
2. <http://school-collection.edu.ru/>
3. <https://infourok.ru/>
4. <http://www.klyaksa.net/>
5. <https://www.yaklass.ru/>
6. <https://pythontutor.ru/>
7. <https://pythonworld.ru/bookshop>
8. http://www.booksgid.com/operating_systems_databases/
9. <https://www.tiensmed.ru/programmer1.html>
10. <https://comprost.ru/komplektuiushchie/sborka-pc>
11. https://5urokov.ru/bosova_9_uch/1_6

Мазмұны

Алғы сөз	4
1-БӨЛІМ. АҚПАРАТПЕН ЖҰМЫС ЖАСАУ	5
§ 1. Ақпараттың сипаттары	6
§ 2. Компьютерде ұзақ уақыт жұмыс істеу тәуекелін сыни түрде бағалау	10
§ 3. Құжаттармен бірлесіп жұмыс жасау	15
§ 4. Желілік этикет	24
1-бөлімді қорытындылауға арналған тест тапсырмалары	28
2-БӨЛІМ. КОМПЬЮТЕР ТАҢДАЙМЫЗ	31
§ 5. Компьютердің конфигурациясы	32
§ 6. Практикум. Компьютердің конфигурациясы	38
§ 7. Программалық жасақтаманы таңдау	40
§ 8. Компьютер құнын есептеу	45
2-бөлімді қорытындылауға арналған тест тапсырмалары	54
3-БӨЛІМ. ДЕРЕКТЕР БАЗАСЫ	57
§ 9. Деректер базасы: негізгі түсініктер мен анықтамалар	58
§ 10. Кесте. Өріс және жазба туралы түсінік	62
§ 11. Деректер базасы. Электрондық кестеде деректер базасын құру	65
§ 12. Ақпаратты іздеу әдістері	69
§ 13. Деректерді сұрыптау және сүзгілеу	74
§ 14. Практикум. Деректер базасымен жұмыс. «Пәндік олимпиада» деректер базасын құру	78
§ 15. Практикум. Деректер базасында іздеу және сүзгілеу бойынша жұмыс	81
3-бөлімді қорытындылауға арналған тест тапсырмалары	82
4-БӨЛІМ. PYTHON ПРОГРАММАЛАУ ТІЛІНДЕ АЛГОРИТМДЕРДІ ПРОГРАММАЛАУ	85
§ 16–17. Бірөлшемді массив	86
§ 18–19. Берілген сипаттары бар элементтерді іздеу	93
§ 20. Практикум. Элементтердің орнын ауыстыру	100
§ 21. Практикум. Массив элементтерін сұрыптау	103

§ 22. Практикум. Элементті жою және кірістіру	107
§ 23. Екіөлшемді массивтер	111
§ 24. Практикум. Екіөлшемді массивте элементті жою және кірістіру	114
§ 25. Практикум. Екіөлшемді массивтерді шешу	116
4-бөлімді қорытындылауға арналған тест тапсырмалары	118

5-БӨЛІМ. PYTHON ПРОГРАММАЛАУ

ТІЛІНДЕ 2D ОЙЫНЫН ҚҰРУ	121
§ 26. PyGame кітапханасы. Кітапхананы қосу	122
§ 27. Ойын терезесін жасау үшін PyGame кітапханасының дайын модульдерін пайдалану	127
§ 28. PyGame кітапханасының басқа да дайын модульдерімен танысу	131
§ 29. Артқы фон мен ойын кейіпкерлері	139
§ 30. Кейіпкерлерді анимациялау. Спрайттарды жасау. Ойынға дайын кейіпкерлерді жүктеу	145
§ 31–32. Спрайттарды пернетақтамен қозғалту және басқару	150
§ 33–34. Практикум. Шарттарды программалау. Дайын сценарий бойынша ойын әзірлеу	158
5-бөлімді қорытындылауға арналған тест тапсырмалары	165
Глоссарий	168
Пайдаланылған әдебиеттер	172

Оқулық басылым

**Салғараева Гүлназ Ибрагимқызы
Калымова Кульзия Акрашевна
Орынтаева Жанна Абдыхамитовна**

ИНФОРМАТИКА

Жалпы білім беретін мектептің
9-сыныбына арналған оқулық

Бас редакторы	Қ.Қараева
Редакторы	А.Умбеткалиева
Техникалық редакторы	В.Бондарев
Көркемдеуші редактор	Е.Мельникова
Бильд редакторы	Ш.Есенкулова
Суретші-безендіруші	О.Подопригора
Мұқабаның дизайны	В.Бондарев, О.Подопригора
Беттегендер	Л.Костина, С.Сулейменова Г.Илишева, Т.Макарова



Назар аудар

Электронды қосымша жүктелген CD қолжетімсіз болған жағдайда, қосымшаны *arman-pv.kz* сайтынан тауып, өз компьютеріңе жүктеп алуыңа болады

Сатып алу үшін мына мекенжайларға хабарласыңыздар:

Нұр-Сұлтан қ., 4 м/а, 2 үй, 55 пәтер.

Тел.: 8 (7172) 92-50-50, 92-50-54. E-mail: astana@arman-pv.kz

Алматы қ., Ақсай-1А м/а, 28Б үй.

Тел.: 8 (727) 316-06-30, 316-06-31. E-mail: info@arman-pv.kz

«Арман-ПВ» кітап дүкені

Алматы қ., Алтынсарин к/сі, 87 үй. Тел.: 8 (727) 303-94-43.

Теруге 21.07.18 берілді. Басуға 06.06.19 қол қойылды. Пішімі 70 x 100 ¹/₁₆.

Қағазы офсеттік. Қаріп түрі «ММ Мектептік». Офсеттік басылыс.

Шартты баспа табағы 14,19. Таралымы 50000 дана.

Артикул 809-006-001к-19