



РЕСПУБЛИКАЛЫҚ
QAZBILIM
ОРТАЛЫҒЫ

ЖАҢАРТЫЛҒАН БАҒДАРЛАМА

ИНФОРМАТИКА
ҰБТ

ІІ БӨЛІМ

Гүлмира ЖАСҰЛНОВА



ЖҮЙЕЛІ БАҒДАРЛАМА
САПАЛЫ БІЛІМ
ҮЗДІК НӘТИЖЕ



РЕСПУБЛИКАЛЫҚ "QAZBILIM" ОРТАЛЫҒЫ

- Мектепке дайындық;
- НЗМ (НИШ), БИЛ, РФММ, «Зерде» мектебі т.б. дарынды балаларға арналған мектептерге дайындық;
- 1-11-сынып бағдарламасы бойынша қосымша білім беру;
- ҰБТ-ға кешенді дайындық;
- Тіл курсары (қазак, ағылшын, түрік, орыс, қытай) тілдері және шет тілдері емтихандарына (TOEFL, IELTS, Duolingo English Test) дайындық;
- Пәндік олимпиадаларға дайындық;
- ҰБТ, пәндік олимпиадалар мен күнделікті сабакқа арналған арнайы материалдар дайындау және сату;
- Мамандық таңдауға арналған арнайы сабактар жүргізу, кеңестер беру;
- Психологиялық тренингтер өткізу;
- Республикалық деңгейде мұғалімдер мен оқушыларға арналған шеберлік курсарын үйімдастыру.

1
Нұр-Сұлтан қаласы:
1. К. Сатбаев 22/1
+7775 431 8008
2. Сарыарқа 31/2
+7701 531 8008
3. Ә. Бекейхан 17/1
+7701 941 8008

2
Алматы қаласы:
1. Мамыр-4 шағын
ауданы, 14-й
+7701 731 8008
2. Н. Гоголь 73
+7701 751 8008

3
Ақтөбе қаласы:
К. Сатбаев 10А
+7701 732 8008

4
Шымкент қаласы:
Д. Конан 39/1
+7701 761 8008

5
Ақтау қаласы:
29-шагын аудан,
Толқын 2, 121/2
+7701 762 8008

6
Қызылжар
(Петропавл) қаласы:
Н. Назарбаев 69А
+7701 752 8008

7
Қарғанды қаласы:
Құртысышылар 4
+7701 973 8008

8
Семей қаласы:
Ә. Таңғыберген 1
+7701 972 8008

9
Қызылорда қаласы:
Ә. Бекейхан 146
+7701 934 8008

10
Түркістан қаласы:
К. Тотібаев 4А
+7701 954 8008

11
QAZBILIM ONLINE
1. Онлайн болімі
+7707 831 8008
2. QAZBILIM
платформасы
+7707 431 8008

12
Кітап болімі
+7776 431 8008
НЗМ болімі
+7701 754 8008



РЕСПУБЛИКАЛЫҚ
QAZBILIM
ОРТАЛЫҒЫ

Жасұланова Гүлмира Тілекбайқызы

ИНФОРМАТИКА

ҰБТ

II бөлім

(Жаңартылған бағдарлама)

*Республикалық «QAZBILIM» орталығы
Nur-Sultan 2022*

ӘОЖ 373.167.1

КБЖ 32.973 я72

Ж33

Ж33 ИНФОРМАТИКА. ҰБТ. 2-бөлім. Г. ТІЛЕКБАЙҚЫЗЫ. Нұр-Сұлтан: «QAZBILIM» баспасы, 2022. – 151 бет.

ISBN 978-601-08-2660-1

Информатика – акпаратты іздестіру, тасымалдау, сактау, өндіу және әртүрлі салада қолдану тәсілдерін зерттейтін ғылым.

Осы курстың негізгі мақсаты – оқушыларға осы курста информатика пәнін түсіндіре отырыш, теорияны практикада қолдануға мүмкіндік беру. Курс үш бөлімнен тұрады. Атап айтқанда, информатиканың «Компьютерлік жад және акпараттың өлшем бірліктері», «Компьютерлік желілер және акпараттық қауіпсіздік», «Электрондық кесте арқылы есептер шығару», «Деректерді ұсыну», «Web-жобалау» «Акпараттық жүйелер», «Жасанды интеллект», «Заттар интернеті», «IT STARTUP», «Алгоритмдеу және программалау» атты бөлімдері қамтылған.

Әр бөлім, тақырыптан соң есепті шығару жолдары мен тест тапсырмалары көрсетілген. ҰБТ-да кездесетін тест сұрақтары мен бірінші, екінші бөлімдегі тест сұрақтарының жауаптары үшінші бөлімде берілген. Бұл тест он нұсқадан тұрады, әр нұсқада бір немесе бірнеше жауапты және контекстен құралған 35 сұрақ бар.

ISBN 978-601-08-2659-5

ISBN 978-601-08-2660-1 (2-бөлім)

© Республикалық «QAZBILIM» орталығы

МАЗМУНЫ

VI БӨЛІМ. АҚПАРАТТЫҚ ЖҮЙЕЛЕР	
6.1. Деректер корына кіріспе.....	4
6.2. MySQL-да біркестелік деректер қорын жасау.....	9
6.3. SQL сұраныстар.....	17
6.4. Web-беттің деректер корымен байланысын орнату.....	28
6.5. Формалар.....	31
6.6. Есептер.....	33
6.7. BigData.....	35
VII БӨЛІМ. ЖАСАНДЫ ИНТЕЛЛЕКТ	
7.1. Жасанды интеллект, машиналық оқыту.....	43
7.2. Нейрондық желіні үйымдастыру.....	49
7.3. Нейрондық желіні жобалау.....	51
7.4. Виртуалды және кеңейтілген шынайылық.....	58
7.5. 3D панорама.....	59
7.6. Виртуалды тур.....	62
7.7. Виртуалды машиналар.....	63
7.8. МобиЛЬДІ құрылғылар.....	65
VIII БӨЛІМ. ЗАТТАР ИНТЕРНЕТІ	
8.1. «Заттар интернеті» дегеніміз не?.....	72
8.2. «Ақылды үй».....	73
IX бөлім. IT STARTUP	
9.1. Стартап қалай іске қосылады?.....	89
9.2. Жобаны алға жылжыту.....	91
9.3. IT STARTUP және жарнама.....	94
9.4. Қазақстандағы цифрландыру.....	97
9.5. Blockchain (блокчейн) технологиясы.....	99
X БӨЛІМ. АЛГОРИТМДЕУ ЖӘНЕ ПРОГРАММАЛАУ	
10.1. Python программау тілінде тармақталған алгоритмдерді жазу.....	107
10.2. Алгоритмнің трассировкасын жүзеге асыру	109
10.3. Сұрыптау алгоритмдерін қолдану.....	113
10.4. Python программау тілінде файлды оқу және жазу.....	115
10.5. Python программау тіліндегі кірістірілген шарттарды қолдану.....	121
10.6. Python программау тіліндегі күрделі шарттарды қолдану.....	123
10.7. While цикл операторын пайдалану.....	125
10.8. For цикл операторын пайдалану.....	130
10.9. Цикл басқару нұсқаулығын қолдану (continue, break, else).....	133
10.10. Бір өлшемді массивтер пайдаланып Python программау тілінде программалар жасау.....	139
10.11. Екі өлшемді массивтер пайдаланып Python программау тілінде программалар жасау.....	142
Пайдаланылған әдебиеттер тізімі	150

VI БӨЛІМ. АҚПАРАТТЫҚ ЖҮЙЕЛЕ

6.1. Деректер қорына кіріспе

Реляциялық деректер қорын қолдануы 1970 жылы IBM компаниясының докторы Эдгар Кодд ұсынды.

Ақпараттық жүйе – пайдаланушыларды қажетті ақпаратпен қамтамасыз ететін аппараттық және программалық құралдар.

Біздің өмірімізге ақпараттық жүйелер көптеп енді. Ақпараттық жүйелердің көмегімен ауа райы болжамы мен пойыздардың жүру кестесін білу, ұшактарға билет сатып алу, конакүй бөлмелерге тапсырыс беру және т.б. жүзеге асырылады.

Ақпараттық жүйенің екі негізгі міндеті:

- деректерді сактау;
- деректерге қол жеткізу, яғни деректерді іздеу және өзгерту мүмкіндігін қамтамасыз ету.

Ақпараттық жүйелер жұмыс істейтін деректер жиынтығының көлемі әдетте үлкен болады (бірнеше гигабайт, тіпті терабайт) және ол деректер компьютердің сыртқы жадында орналасады (мысалы, серверде, бұлттық сервистерде және т.б.). Деректер іздеуге және өзгертуге оңай болатындей түрде сакталады. Мұндай деректер жиынтығы деректер қоры деп аталады.

Деректер қоры – компьютердің сыртқы жадында сакталған белгілі бір сала туралы арнайы ұйымдастырылған деректер жиынтығы.

Реляциялық деректер қоры – кестелер жиынтығы ретінде ұсынуға болатын деректер қоры.

Эдгар Кодд келесі идеяларға негізделген жаңа деректер моделін ұсынды:

- барлық деректер кейбір объектілердің қасиеттерін көрсетеді;
- объектілер кластарға белінеді. Мысалы, музикалық топтар туралы деректерді сипаттағанда «Топ, Альбом, Эн» және т.б. кластарды пайдалануға болады.

Объектілер туралы деректер – қасиеттер жиынтығы (атрибуттар), онда әрбір қасиет «атауы-мәні» жұбы ретінде көрсетіледі. Мысалы, «Music 44» музикалық тобы туралы деректерді келесі түрде жазуға болады: Аты: «Music 44». Жетекші: Даниал Әнүарбек. Құрылған жылы: 1987.

Э. Кодд енгізген деректер моделі деректердің реляциялық моделі деп аталады (ағылш. Relation қатынас). Реляциялық деректер қоры кестелерден тұрады. Реляциялық деректер қорындағы кестенің негізгі қасиеттері:

- әр кесте объектінің бір класын сипаттайты;
- әрбір бағанның қайталаңбайтын аты бар (кесте ішінде);
- кестеде бағандардың орналасу тәртібі маңызды емес;
- бір бағанның барлық мәндері бір деректер типіне жатады (сан, мәтін, күні...);
- кестеде екі бірдей жол болмайды;

- кестедегі жолдар реті анықталмаған.

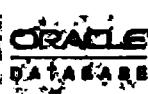
Егер деректермен жұмыс істей алмасақ, олар пайдасыз болады. Сондықтан деректерді іздеуге және өзгертуге мүмкіндік беретін арнайы программалық қамтамасыз ету қажет.

«Деректер қоры» мен «ДКБЖ» түсініктерінің айырмашылығы неде?

«Деректер қоры» және «ДКБЖ» терминдері әртүрлі ұғымдарды билдіргенімен, олар өзара тығыз байланысты: деректер қорының қасиеттері оны басқаратын ДКБЖ-де анықталады және көрініше. «ДК + ДКБЖ» кешені деректер қорының жүйесі (ағыл. Database system) немесе ақпараттық жүйе деп аталағы. ДКБЖ деректерді басқару бойынша барлық міндеттерді шешеді, соның ішінде:

- деректерді іздеу, деректерді өндіру;
- қарапайым есептеудерді орындау;
- деректердің тұгастығын қамтамасыз ету (дұрыстығын, дәйектілігін);
- ақаулардан кейін деректерді қалпына келтіру.

Реляциялық деректер қорын басқарудың танымал жүйелері:

1-кесте				
				
Oracle көп жағдайда кәсіби және үлкен қосымшалар үшін қолданылады.	Microsoft SQL Server - Microsoft компаниясының ДКБЖ-сі. Тек Windows операциялық жүйесі үшін қолжетімді.	MySQL – танымал, еркін таратылатын ашық бастапқы коды бар ДКБЖ.	IBM бірнеше ДКБЖ-сі бар, ең танымалы – DB2.	MS Access – MS Access пайдаланушыға арналған интерфейсі бар деректер қорын жасауға мүмкіндік беретін ДКБЖ.

Деректер қорын басқару жүйесі (ДКБЖ) – ДК құру, ақпаратты жаңарту және толықтыру мүмкіндігін беретін, ақпаратқа икемді колжеткізуі қамтамасыз ететін программалық жабдықтама.

Кез келген кестелер жинағы реляциялық деректер қоры ретінде қарастырыла алмайды. Жоғарыда айтылғандай, деректер қоры және ДКБЖ тығыз байланысты және деректер қоры жүйесін (ДК + ДКБЖ) белгілі бір түрге жатқызу үшін тиісті ДКБЖ-да қандай деректерді басқару әдістерін қолданылатынын болу қажет.

Қазіргі заманғы ақпараттық жүйелердің көпшілігінде деректерді басқару үшін келесі командалары бар SQL тілі қолданылады:

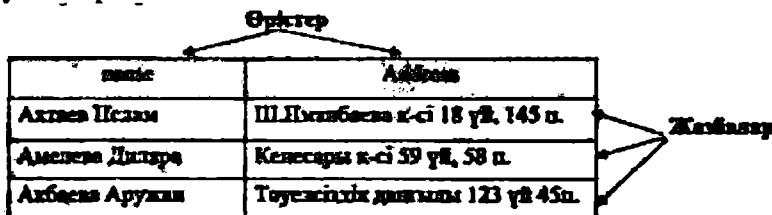
- жана кестелерді құру;

- жаңа жазбаларды косу;
- жазбаларды өзгерту;
- жазбаларды жою;
- берілген шартқа сәйкес бір немесе бірнеше кестелерден жазбаларды тандау және басқалар.

SQL тілінің командалары деректерді сактау форматына, яғни кестедегі бағандар мен жолдардың орналасу тәртібіне «тәуелсіз» басқаруға мүмкіндік береді. Операцияларды орындау үшін (тандау, кірістіру, жою, өзгерту) бағандар мен кестелердің атаулары ғана пайдаланылады. SQL командаларының көмегімен Э. Кодд енгізген барлық негізгі операцияларды орындауға болады, сондықтан SQL тілін қолданатын ДКБЖ (және деректер қорының тиісті жүйелері) реляциялық деп аталады.

SQL (Structured Query Language – сұраныстардың құрылымдық тілі) – реляциялық деректер қорына арналған деректер қорын басқару тілі.

«Окушылар тізімі»



1-сурет. Реляциялық деректер қорында кестенің берілуі

Кестенің бағандары өрістер деп, ал жолдар жазбалар деп аталады.

Кез келген өрістің қайталанбайтын атауы болуы керек. Мысалы, екі өріске Адрес деп атау беруге болмайды, бірақ бірін Адрес, ал екіншісін Үақытша адрес деп атауға болады.

Реляциялық деректер қоры өзара байланысқан кестелер жиынтығынан тұрады. Кестенің ербір жолы бір объект туралы деректерді қамтиды (мысалы, автомобиль, компьютер, клиент), ал кестенің бағандары осы объектілердің атрибуттар (мысалы, козғалтқыш нөмірі, процессор маркасы, фирмалар мен клиенттердің телефондары).

1-суреттегі кесте «объект-қасиеттер» типіне жатады, яғни жазба – бір объекттің сипаттамасы (бұл жағдайда адамның), ал өрістер осы объекттің қасиеттерін билдіреді. Берілген кестеде екі өріс (Name, Address) және үш жазба бар.

Жоғарыда айтылған адрестері бар кестеде көптеген адамдардың деректері жазылған болсын (нақты деректер қорында миллиондаған жазба болуы мүмкін).

Практикада әр кестеден бірнеше өрістер бойынша іздеу қажет болады және әр уақытта жазбаларды сұрыптау мүмкіндігі болмайды (көлемді деректер қоры үшін бұл өте үзак). Мұндай жағдайда жылдам іздеуді қалай қамтамасыз етуге болады?

Программалау кезінде жиі орын алатын әдіс қосымша жадты қолдану арқылы алгоритмнің орындалу жылдамдығын арттыруға болады. Жазбаларды іздеуді тездету үшін көптеген кітаптарға индексті қосады (2-сурет). Мұндағы индекс – жазбаларда кездесетін беттері көрсетілген кілттік сөздер тізімі. Деректер қорында арналы іздеу үшін қосымша кестелер құрылады және оларды индекстер деп атайды.

Индекс – негізгі кестедегі іздеуді тездету үшін көмектесетін көмекші кесте.

Енді тегі Ахтаев деген адамдарды іздеу қажет болса, жылдам екілік іздеуді пайдалану арқылы индекстегі Ахтаев нөмірлерін іздеуге болады (онда тегі алфавит бойынша орналасқан!). Содан кейін қажетті жазбалар нөмірлері анықталған кезде, Ахтаев туралы деректерді негізгі кестеден аламыз. Негізгі кестеден басқа нәрселерді іздеудің керегі жок. Осылайша екілік іздеуді әртүрлі өрістерде пайдалануға болады.

Тегі	№
Ахтаев	3
Амелева	2
Акбаева	1

2-сурет. Тегі өрісі бойынша индекс

Қалай ойлайсыңдар, бірегей кілт дегеніміз не?

Алғашқы кілт немесе **кілт** – өріс немесе өрістер комбинациясы, ол кестедегі әрбір жазбаны бір мәнде анықтайтын мәндер жиынтығы.

Кестедегі деректерді оқығанда және өзгертуенде, қажетті жазбаға жүгініп отырганымызды білу өте маңызды. Сенімді жұмыс істеу үшін әрбір жазбаның басқаларынан ерекше бірегей мәні болуы керек.

Бұл кілттің бірегейлік қасиетін білдіреді: кестеде кілттің бірдей мәні бар екі жазбаның болуы мүмкін емес. Мысалы, жеке куәлік нөмірі, үялы телефон нөмірі, электрондық пошта адресі және т.б. кілт бола алады.

Кілттік өріс

	Аты	Сынып	Тұган күni
1	Адылканов	Амир	11
2	Ахан	Алихан	10
3	Балмаганбет	Асанали	10

3-сурет. Кілттік өріс

Кілтті пайдаланудың мынадай артықшылықтары бар:

- жазбалардың бірегейлігі кестеде кілттік өріс бірдей мәні бар жазбаларды енгізуге және сактауға мүмкіндік бермейді;
- байланыстар: кілтті пайдалану арқылы кестелер арасында байланыстар орнатылады;

• жылдамдығы: кілттік өріс бойынша қажетті жазбаларды іздеуді және сұраныстарды орындауды тездегетін индекс құрылады;

• ретке келтіру кесте жазбаларын кілттік өрістегі мәндердің өсу немесе кему ретімен автоматты түрде сұрыптайды және бейнелейді.

ДК құру оны жобалаудан басталады. ДК жобалау процесі келесі негізгі кезеңдерден тұрады.

Деректер корын құру алгоритмі:

1. **ДК қызыметін анықтау.** Оның көмегімен шешілетін міндеттер тізімін және бұл үшін қандай деректер қажет екенін анықтау.

2. **Кестелер құрылымын анықтау.**

• Кестелердің әрқайсысы біртіпті объектілер жиынтығы туралы ақпаратты қамтуы керек, мысалы, окупшылар туралы ақпарат немесе емтиханның корытынды нәтижелері.

• Бұл деректер жиынтығының әрқайсысында жеке кесте болуы керек. Мысалы,

1 емтиханға қатысушылар және олардың бағалары туралы ақпарат әртүрлі кестелерде сакталуы тиіс. Сонда окупшының бағалары туралы ақпарат жойылғанымен, олар туралы деректер қорда қала береді.

• Кестедегі ақпарат қайталаңбауы керек. Кестелер арасында да қайталау болмауы тиіс. Бұл әртүрлі кестелердегі ақпараттың сәйкес келмей қалу мүмкіндігін болдырмайды және ДК-мен жұмысты тиімдірек етеді.

3. **Өрістерді анықтау.** Өр кесте бір типті объектілер жиынтығы туралы ақпараттан тұрады. Кестеге арналған өрістерді құру кезінде келесілерді есте сақтау қажет:

• кестеде берілген объектілер жиынтығы туралы барлық қажетті ақпарат болуы керек;

• әрбір өріс басқа объектілердің жиынтықтары туралы емес, олардың нақты сипатына қатысты ақпаратты қамтуы тиіс.

4. **Кілттік өрісті анықтау.** ДК әртүрлі кестелердегі деректерді байланыстыру үшін әр кестеде бастапқы кілт немесе жай кілт болуы керек.

5. **Кестелер арасындағы байланысты анықтау.** Деректерді кестелер бойынша таратып, кілттік өрістерді анықтағанинан кейін, әртүрлі кестелердегі деректерді байланыстыру үшін схеманы тандау қажет. Ол үшін кестелер арасындағы байланысты анықтаймыз. Бұл байланыстар көп кестелі сұраныстарды, формаларды және есептерді құру кезінде пайдаланылады.

6. ДК құру үшін программалық жабдықтарды тандау.

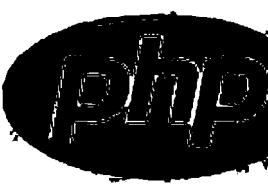
6.2. MySQL-да біркестелік деректер қорын жасау

Деректер қорымен жұмыс істеу үшін компьютерді дайындау

Бұл бөлімде біз деректер қорын жасауды және деректер қорымен web-беттердің байланысын орнатуды үйренеміз. Сондыктан біз компьютерде MySQL ДКБЖ-мен жұмыс істеу үшін визуалды интерфейсті ғана емес, Apache және PHP-ді орнату қажет. Бұл құралдар OpenServer деп аталатын орнату файлында жинақталған (www.ospanel.io) (2-кесте).

Деректер қорын жасауға және оны web-бетпен байланыстыруға арналған OpenServer құрамына кіретін құралдар

2-кесте.

 APACHE <small>HTTP SERVER PROJECT</small>	<p>Apache – HTTP web-сервер, Интернет желісінде деректерді беруге арналған сервердің арнайы қосымшасы. Серверге клиенттен сұраныс түскен кезде Apache сервері жауап береді. Ол келесі ОС: Linux, BSD, Mac OS, Microsoft Windows колдайды. Apache web-серверінің артықшылықтары: PHP, Python, Ruby, Perl, ASP программалау тілдерін қолдау. Біздің жағдайда жергілікті компьютер сервер және клиент болады.</p>
	<p>PHP (Hypertext Preprocessor гипермәтін препроцессоры) – web-серверде орындалатын web-қосымшаларды сценарийлерді) жазу үшін арнайы әзірленген программалау тілі. PHP тілінің артықшылығы – енгізілген PHP командалары бар HTML құжаттарын жасау мүмкіндігінде.</p>

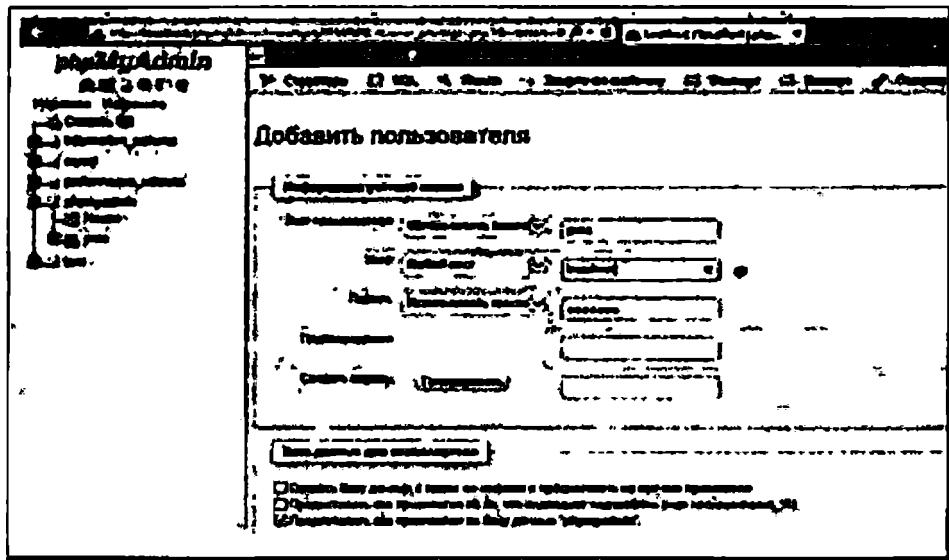
Жергілікті компьютерде OpenServer-ді орнатуды және бастапқы баптауды қосымша акпарат айдарынан қарандар.

OpenServer сәтті орнатылғаннан кейін, PhpMyAdmin-ды ашындар. PhpMyAdmin басты терезесі 4-суреттегідей көрінеді.

1. Бүйірлік тақта – дәл осы уақыттағы қолжетімді деректер қорының және олардың кестелерінің тік тізімі көрсетіледі. Кестелерді көру үшін олардың әрқайсысын ашуға болады. Сондай-ақ жылдам терілетін батырмалар бар – Жаңарту, Үйге және т. б.;

2. Басты меню – сервер атауының астында орналасқан жолак. Оның көмегімен деректер қоры, SQL, пайдаланушылар, экспорт және импорт сияқты программаның әртүрлі мүмкіндіктері бойынша қозғалуға болады.

3. Жұмыс аймағында жұмыс ақпараты көрсетіледі, қазіргі уақытта басты терезеде сервер туралы ақпарат бар, басқа жағдайларда деректер қорының мазмұны немесе мәтін енгізу өрісі болуы мүмкін.



4-сурет. PhpMyAdmin басты терезесі

Бір кестелік деректер қорын құру

CREATE DATABASE – деректер қорын құруға арналған SQL командасы.

Синтаксисі:

CREATE DATABASE name_database;

Мұндағы name_database – ДК атауы.

MySQL кестелерін құру

Кестелер CREATE TABLE нұсқаулығын пайдалана отырып жасалады және синтаксисі келесідей болады:

CREATE TABLE 'TableName'('fieldName' dataType [косямша параметр]);

- **CREATE TABLE** деректер қорында кестені құруға жауапты.

- **fieldName** – өрістің аты,

- **dataType** – деректер типі, өрісте сакталуы тиіс деректердің сипатын анықтайды.

- [қосымша параметрлер] auto_increment, NOT NULL және т.б. сияқты өріс туралы қосымша ақпарат.

ДЕРЕКТЕР ТИПТЕРИ

MySQL-дің 3 негізгі деректер типі бар:

- 1) сандық;
- 2) мәтін;
- 3) күні / уақыты.

Деректердің сандық типтері

INT ()	-2147483648-ден 2147483647-ге дейін қалыпты, 0-ден 4294967295-ке дейін ШЕКСІЗ.
FLOAT	жылжымалы үтірмен шамамен улкен емес сан.

Мәтіндік тип

CHAR ()	Ұзындығы 0-ден 255 таңбаға дейін бекітілген бөлім.
VARCHAR ()	Ұзындығы 0-ден 255 символға дейінгі айнымалы бөлім.

Күні/Уақыт

DATE	Күнін (дата) сактауға арналған деректер типі.
TIME	Уақытты сактауға арналған деректер типі.
YEAR	Жылды сактауға арналған деректер типі.

MySQL-де кейбір басқа деректер типтері де кездеседі.

BOOL	tinyint (1) синонимі логикалық мәндерді сактау үшін қолданылады.
------	--

Күрылған musical_instruments деректер қорында код, атауы, типі, data (жасалу уақыты) шек саны, фото өрістері бар instruments кестесін жасап шығарамыз.

```
create table instruments (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name varchar (20),
    type varchar (20),
    data date,
    num_string int(10),
    foto Varchar (20)
);
```

INSERT INTO дегеніміз не?

Деректер коры жүйесінің негізгі міндеті – деректерді кестелерде сактау. Деректер әдетте деректер қорының үстінен жұмыс істейтін қолданбалы программалармен беріледі. Осы мақсатта SQL-де кестедегі деректерді сактау үшін пайдаланылатын INSERT командасы бар.

Негізгі синтаксисі:

```
INSERT INTO 'table_name'
VALUES (value_1, value_2, ...);
```

INSERT INTO 'table name' MySQL серверіне table_name атымен кестеге жаңа Жол қосу туралы хабарлайтын команда.

- VALUES (value_1, value_2,...) жаңа жолға қосылатын мәндерді көрсетеді.

Жаңа кестеге енгізу қажет деректер мәндерін қойғанда, әртүрлі деректерді қараша кезінде келесілерді ескеру қажет:

- Деректердің жолдық типі – барлық жолдық мәндер бір тырнақшага алыныу тиіс.

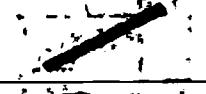
- Деректердің сандық типі – барлық сандық мәндер оларды бір немесе екі рет тырнақшага алмай тікелей берілуі тиіс.

- Құні тиіп «ГГГГ-ММ-ДД» форматындағы бір тырнақшага күннің мәнін енгізу.

Кестеде көрсетілген музикалық аспалты деректер қорына қосу керек

Instruments

3-кесте.

id	name	type	data	Num-string	photo
1	Домбыра	Шертпелі	15.07.2022	2	
2	Кобыз	Керілген қолмен	11.07.2021	2	
3	Жетіген	Шертпелі	15.05.2020	7	
4	Сыбызығы	Үрмелі	02.11.2021	0	
5	Дабыл	Сокпалы	23.06.2022	0	
6	Шертер	Шертпелі	07.11.2021	2	
7	Сазсырнай	Үрмелі	21.01.2022	0	

INSERT командасы деректерді бірінен кейін бірін енгізеді. Біз домбырадан бағтамыз. Ішектер санын сандық тип ретінде алаңыз. Санды тырақшага алушың қажеті жок.

```
INSERT INTO instruments VALUES
(1,'Домбыра','Шерпелі','2022-07-15',2,'dombyra.jpg');
INSERT INTO instruments VALUES (2,'Кобыз','Керілген Қым мен
-07-11',2,'kobyz.jpg');
```

Біз барлық деректер бір кестеге біріктірілген қарапайым құрылымды қарастырдық, онда ақпаратты іздестіру оңай. Алайда мұндай модельдің кемшиліктері де бар:

- деректердің қайталануы: мысалы, musical_instruments деректер қорында аспап түрі көп рет сакталады;
- кейбір деректерді өзерткенде (мысалы, Аспаптың аты) бірнеше жазбаны өзгерту керек болады;
- қате көп кетуі мүмкін.

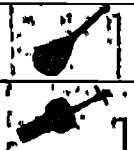
Біркестелік деректер коры – барлық карточкалар бірдей құрылымдалған картотекаға ұксас. Сонымен катар іс жүзінде бір деректер қорында бір-бірімен байланысты әртүрлі объектілерді сактау керек. Бұдан «бұл деректерді сипаттау және сақтау үшін қай модельді қолданған жақсы?» деген сұрақ туындаиды.

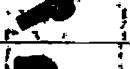
Мысалы, қазақ музикалық аспаптарын сататын интернет-дүкеннің деректер қорын қалай үйімдастыруға болатынын қарастырайық. musical_instruments деректер қорына келесі деректерді қосамыз: material (дайындағы материал), producer (өндіруші), price (бага), sales (жекілдіктер), num_instr (ішкендегі аспаптар саны) және жаңа деректер қорын kaz_mus_instruments деп атаемиз. Біркестелік деректер қорында «musical_instruments» 4-кестеде көрсетілген.

Кестеде қайталанулардың бар екенин көреміз – осы өндірушінің әрбір куралы үшін өндірушінің атауы (таңба жолы) қайталанады. Себебі берілген кестелерде тек аспап туралы ақпарат берілмей, онымен қоса басқа типті ақпарат, яғни өндірушілер туралы да мәлімет бар. Соңдықтан өндіруші туралы барлық ақпаратты сактау үшін бөлек кесте құру керек.

Musical_Instruments

4-кесте.

id	name	type	material	producer	Nu m_i nstr	price	sal es	data	Num - strin g	photo
1	Домбыра	Шерпелі	Карагай	Тұрдығұлов Ж	5	15000 0		15.07 .2022	2	
2	Кобыз	Керілген қолмен	Кайын	Абрамкин В.	4	16000 0		11.07 .2021	2	

3	Жетіген	Шертпелі	Үйенкі	ЖШС «Лира»	6	18000 0		15.05 .2020	7	
4	Сыбызығы	Ұрмелі	Бамбук	Юпитер зауыты	15	20000	20 %	02.11 .2021	0	
5	Дабыл	Сокпалы	Қайың	Юпитер зауыты	9	60000		23.06 .2022	0	
6	Шертер	Шертпелі	Жанғақ	ЖШС «Лира»	12	10000 0	40 %	07.11 .2021	2	
7	Сазсырна й	Ұрмелі	сазбал шық	«Sapaly Saz»	35	15000	20 %	21.01 .2022	0	
8	Домбыра	Шертпелі	Бамбук	«Sapaly Saz»	21	15000 0	28 %			
9	Қобыз	Керілген қолмен	Қарагай	Абрамкин В.	3	12000 0				
10	Шертер	Шертпелі	Қайың	«Sapaly Saz»	17	85000				

Produsers

5-кесте.

Id_produser	name	address
1	Тұрдығұлов Ж.	Алматы қ-сы, Абай к-си, 56
2	Абрамкин В.	Шымкент қ-сы, Шәкәрім к-си, 21
3	«Sapaly Saz»	Алматы қ-сы, Рысқұлов к-си, 141
4	ЖШС «Лира»	Павлодар қ-сы, Баянауыл к-си, 11
5	Юпитер зауыты	Мәскеу қ-сы, Калужская к-си, 8

Мұнда бірінші кілт – `id_producers`. Кестеге өндіруші туралы басқа деректерді қосуға болады, мысалы, құрылған жылы, телефоны және т.б. `instruments` кестесінде енді өндірушілердің аттары емес, олардың кодтары сакталады.

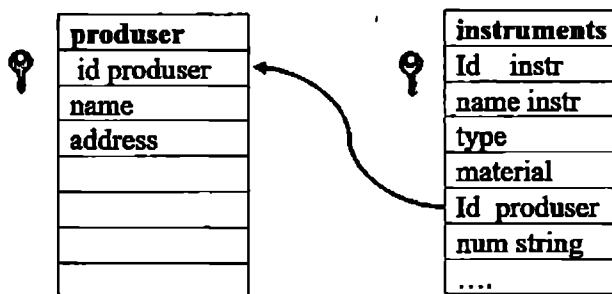
Реляциялық деректер қоры – байланысқан кестелердің жиынтығы.

Осылайша деректер қоры екі кестеден тұрады және объектілердің екі класы туралы акпаратты сактайды. Бұл кестелерді байланыстыру керек. Егер кестелер арасында байланыстар орнатылса, қорда сакталатын деректерді бакылауға болады.

Реляциялық ДК-да кестелер арасындағы байланыс кілттік өрістер арқылы орнатылады. Деректер қорында негізгі кестелер құрылғаннан кейін олардың бір-бірімен қалай байланысқанын көрсету керек. ДҚБЖ-нің бұл байланыстары бірнеше кестелерден акпаратты іріктеу кезінде сұраныстарда формалар және есеп жасауда пайдаланылады. Кестелер арасындағы байланыстарды орнату ДК деректердің тұтастығын қорғауға мүмкіндік береді.

Екі кесте арасындағы байланысты келесідей көрсетуге болады.

ДКБЖ маңызды міндеттерінің бірі – деректер қорының тұтастығын сактау, яғни кестелер арасындағы байланыстардың дұрыс болуын қамтамасыз ету.



5-сурет. Кестелер арасындағы байланыс

ДКБЖ-ның байланысқан кестелердегі әрекеттері:

- деректер қорында байланысты жазбалар бар болған жағдайда жазбаны жоюды боддырмау;
 - каскадтық жоюды орындау, яғни жойылған жазбамен бірге басқа кестелердегі барлық жазбаны жою;
 - езгерістер енгізуге руқсат беру не бермеу.
1. 4-кестедегі **name_instr**, **type** және **material** өрістері үшін индекстік кестелерді жасаймыз

NAME_INSTR 6-кесте		TYPE 7-кесте		MATERIAL 8-кесте	
id_name_ins	name_instr	id_type	name_type	id_material	name_material
1	Домбыра	1	шертпелі	1	қарағай
2	кобыз	2	керілген қылмен	2	қайың
3	жетіген	3	урмелі	3	үйенкі
4	сыбызығы	4	сокпалы	4	бамбук
5	дабыл			5	жанғақ
6	шертер			6	сазбалишық
7	сазсырнай				

2. 4-кестені индекстік кестеге байланысты түрлендіріндер.

Кесте Instruments

9-кесте.

Id_instr	id_name_ins	id_type	id_material	Id_producer	Nu_m_instr	price	sales	Data_begin	Data_end	photo
1	1	1	1	1	5	1500 00				
2	2	2	2	2	4	1600 00				
3	3	1	3	4	6	1800 00				
4	4	3	4	5	15	2000 0	20 %	02.11.20 21	30.11.20 21	
5	5	4	2	5	9	6000 0				
6	6	1	5	4	12	1000 00	40 %	07.11.20 21	30.11.20 21	
7	7	3	6	3	35	1500 0	20 %	21.01.20 22	01.02.20 22	
8	1	1	4	3	21	1500 00	28 %			
9	2	2	1	2	3	1200 00				
10	6	1	2	3	17	8500 0				

Барлық бес кесте үшін өріс типтерін анықтандар.

3. PhpMyAdmin ішінде **kaz_music_instruments** деректер қорын жасау үшін:

Create database **kaz_mus_instruments**;

4. phpMyAdmin ішінде **kaz_mus_instruments** ДК-да **instruments, producers, names, types, materials** кестесін құрамыз.

Create table name instr (

id name ins INT AUTO INCREMENT PRIMARY KEY,
name instr varchar (20));

Мұндагы:

AUTO INCREMENT – автоматты нөмірлеу;

PRIMARY KEY – кілттік өріс.

5. Барлық мәліметтерді мәтіндік режимде кестелерге енгіземіз. SQL қосынша беттің тәмемлі жағындағы батырмаларды пайдалануга болады.

6. INSERT командасын енгіземіз:

INSERT INTO `name_instr` VALUES (NULL, 'домбыра');

6.3. SQL сұраныстар

- Сұраныс – Запрос – Query
- Таңдау – Выбрать – SELECT
- Жою – Удалить – DELETE
- Жаңарту – Обновить – Update
- Біріктіру – Объединить – Join

Сұраныстар (query) – деректер қорынан қажетті ақпаратты таңдау құралы. Дерекқорға қатысты қалыптасқан сұрақ – **сұраныстар**.

SQL жұмыс принциптерін түсіну накты сұраныстарды жасауға көмектеседі және дұрыс емес нәтижелерді қайтаруға мүмкіндік береді, сұраныстарды түзетуді женілдетеді.

SELECT нұсқаулығы SQL тілінде деректер жинағын сипаттау үшін қолданылады. Ол келесілерді қоса алғанда, деректер қорынан алуға қажетті деректер жиынтығының толық сипаттамасын қамтиды:

- деректері бар кестелер;
- әртүрлі дереккөздерінен алынған деректер арасындағы байланыс
- өрістер немесе есептеулер, олардың негізінде деректер іріктеледі;
- сұрыптау қажеттілігі мен тәсілі.
- сұраныстар нәтижесіне енгізілетін деректерге сәйкес келетін іріктеу шарттары;

SQL нұсқаулығы ұсыныс деп аталатын бірнеше бөліктерден тұрады.

SQL нұсқаулығындағы әрбір ұсыныстың өз мақсаты бар. Кейір ұсыныстар міндетті. Төмендегі кестеде жиі қолданылатын SQL ұсыныстары көрсетілген.

10-кесте.

SQL тілінің нұсқаулығы

Командадағы мәндер	Міндетті
SELECT	Қажетті деректерден тұратын өрістерді анықтайды
FROM	SELECT ұсынысында көрсетілген өрістері бар кестелерді анықтайды
WHERE	Нәтижелерге енгізілетін барлық жазбаларға сәйкес келетін өрістерді іріктеу шарттарын анықтайды
ORDER BY	Нәтижелерді сұрыптау тәртібін анықтайды

SELECT синтаксисінің мысалы 6-суретте көрсетілген.

Кілттік сөздердің база әрптермен енгізу келісілген

SELECT таңдау тізімі
FROM кестелердің тізімі
WHERE шарты

6-сурет. SELECT синтаксисінің мысалы

Бұл сұраныстар жазудың ұсынылатын синтаксисі, шын мәнінде, кілттік сөздердің кез келген регистрде енгізуге болады. Таңдау тізімінде деректердің қандай өрістерден таңдау қажет екендігі көрсетіледі. Кесте тізімінде осы өрістердің қандай кестелерден екендігі анықталады. WHERE сөйлемінде қажетті жолдар көрсетіледі. Олар константалар да (тұрақтылар), өрнектер де болуы мүмкін.

Түсінікті болу үшін сұраныстарды талдайық:

- **SELECT** – қандай өрістерді экранға шығарамыз?
- **FROM** – мәліметтердің қандай кестеден аламыз?
- **WHERE** – шартты қанағаттандыратын жазбаларды ғана көрсетеміз.

SELECT операторының Instruments кестесі мысалында пайдаланудың қарастырайық. Барлық Instruments кесте жазбаларын шығару үшін қаралайық **SQL сұраныстар** жазайық:

```
SELECT* FROM Instruments
```

Мұнда * кестенің барлық өрістерін білдіреді. Осы үлгіні орындағаннан кейін 7-суреттегі нәтижекі аламыз.

id	name	type	data	Num-string	photo
1	Домбыра	Шертпелі	15.07.2022	2	Dombra.jpg
2	Қобыз	Керілген қолмен	11.07.2021	2	Kobyz.png
3	Жетіген	Шертпелі	15.05.2020	7	Jetygen.png
4	Сыбызғы	Үрмелі	02.11.2021	0	Sybyzgy.png
5	Дабыл	Сокпалы	23.06.2022	0	Dabyl.png
6	Шертер	Шертпелі	07.11.2021	2	Sherter.png
7	Сазсырнай	Үрмелі	21.01.2022	0	Sazsyrnai.png

7-сурет. Instruments кестесінен мәліметтердің таңдау

Instruments кестесінен шертпелі құралдардың таңдаймыз. Тек олардың атаулары мен жасалған күнін, яғни name және data істерін шығару керек. Содан кейін осы тізімді data өрісі бойынша есу ретімен орналастырамыз.

1. SQL сұраныстар келесідей болады:

```
SELECT name, data FROM Instruments WHERE type="шертпелі";
```

2. Таңдау нәтижесі

Ном	Найыншы күн
Домбыра	2022-07-15
Жетіген	2020-05-15
Шертер	2021-07-11

8-сурет. Шертпелі аспаптар

3. Сұраныстардың келесі шарты **data** өрісі бойынша тізімді өсу ретімен шығару болды. **ORDER BY** ұсынысы сұрыптауды аныктайды, ал **ASC** мәні өсу бойынша сұрыптау түрін көрсетеді (**DESC** – кемү):

```
SELECT name, data FROM Instruments WHERE type="шертпелі";
    ORDER BY data ASC;
```

4. Таңдау нәтижесі

Ном	Найыншы күн
Жетіген	2020-05-15
Шертер	2021-07-11
Домбыра	2022-07-15



9-сурет. Шертпелі аспаптар data өрісінің өсу реті бойынша

DELETE командасы

SQL-дің **DELETE** командасы деректер корының кестелерінен қажет емес жолдарды жою үшін пайдаланылады. **DELETE** командасы деректер қорынан уақытша немесе ескірген деректерді жоюға ыңғайлыш. **DELETE** командасы бір сұраныста кестеден көп жолды жоя алады.

Жол жойылған соң, оны қалпына келтіру мүмкін емес.

Егер **where** ұсынысы **DELETE** сұраныстарында пайдаланылмаса, осы кестедегі барлық жол жойылады.

Delete командасының негізгі синтаксисі төменде көрсетілген.

```
DELETE FROM table_name WHERE [шарт];
```

- **DELETE FROM table_name** MySQL серверіне кестеден жолдарды жою туралы хабарлайды.

- **WHERE [шарт]** міндетті емес және **DELETE** сұраныстарымен сұралған жолдар санын шектейтін сұзгіні (фильтрді) орналастыру үшін пайдаланылады.

Тапсырма:

Musical_instruments деректер қорындағы **instruments** кестесінен сазсырнайды өшіріндер.

1. Осы кестенің өріс атауларын еске түсіріндер. Мұны орындау үшін төмендегі команданы орындаап, кестеде көрсетілген барлық жазбаларды қарап шығындар:

```
SELECT X FROM Instruments;
```

2. Өшіруге арналған SQL сұраныстар:

```
DELETE FROM instruments WHERE name="сазсырнай";
```

UPDATE командасы

Update командасы кестедегі жолдарды өзгерту үшін қолданылады. Сонымен қатар, бір өрісті немесе бірнеше өрісті бір уақытта жаңарту кезінде пайдаланылады, басқа кестедегі мәндермен кестені жаңарту үшін пайдаланылуы мүмкін.

Update командасының негізгі синтаксисі төменде көрсетілген.

```
UPDATE table name' SET' column_name'= 'new_value WHERE' [шарт];
```

- **UPDATE table_name** – кестедегі деректерді жаңарту туралы MySQL хабарлайтын команда.

- **SET column_name = ' new_value'** – жаңарту қажет өрістердің атаулары мен мәндері. Жаңарту мәндерін баптау кезінде осы жолдардың типтері бір тырнақшада болуы тиіс екенін ескеріндер. Сандық мәндер тырнақшада болуы міндетті емес. Осы күннің (датаның) түрі бір тырнақшада және «ГГГГ-ММ-ДД» форматында болуы тиіс.

- **WHERE [шарт]** міндетті емес және UPDATE сұрауына әсер ететін жолдар санын шектейтін сұзгіні (фильтр) орналастыру үшін пайдаланылуы мүмкін.

Тапсырма:

Instruments (ДК musical_instruments) кестесіндегі мәліметтерді жаңартамыз. Сыбызының жасалған күнін 02.11.2021-ден 25.11.2021-ге ауыстырамыз.

1. Ең алдымен сыбызының id-ін анықтаймыз, себебі SQL-командаларды мәтіндік өрістерге қарағанда id-мен жазған ыңғайлы.

```
SELECT * FROM Instruments WHERE name="сыбызы";
```

Жоғарыда көрсетілген сценарийді орындағанда келесідей нәтижениі аламыз, id=4:

id	name	type	date	description	price
4	Сазсырнай	String	2021-11-02	Сазсырнай	100000

2. Күнді (датаны) жаңарту үшін төменде келтірілген скрипті орындау қажет:

```
UPDATE instruments SET data = "2021-11-25" WHERE id = 4;
```

3. Күнінің ауысқанына көз жеткізейік.

```
SELECT * FROM instruments WHERE id = 4;
```

Жоғарыда көрсетілген сценарийді орындағанда келесі нәтижени береді:

id	name	type	data	num strings	photo
3	Себарад	Drum	2021-11-25	1	0

LIKE операторы

LIKE операторы символдық жолдың ұлғімен сәйкестігін орнатады.

LIKE операторының синтаксисі келесідей:

```
expression LIKE pattern
```

Мұндай expression – кез келген символдық өрнек.

Pattern – expression өрнегіне тексеру жүргізілетін ұлғі. Үлгі мынадай арналы символдарды қамтуы мүмкін.

Арналы символдар

11-кесте.

Символ	Кестедегі мағынасы
%	Кез келген жолдың ұзындығы
_	Кез келген бір таңба
	Таңбалардың ауқымы немесе реті
[^]	Таңбалардың ауқымын немесе реттілігін болдырмау

Instruments (musical_instruments ДК) кестесінде:

LIKE операторының көмегімен өз атауында «шертпелі» сөзі бар аспап жазбаларын шығару:

```
SELECT * FROM Instruments WHERE type LIKE 'шерт%';
```

Бұл мысалда LIKE операторының үлгісі ретінде 'шерт %' қызмет етті. Есептің шартына қарай, «шерт» сөзінен кейін кез келген ұзындықтағы символдар болуы мүмкін.

Сұраныс ішіндегі сұраныстар

Көпкестелік деректер қорына күрделі сұраныстар тән; яғни сұраныс ішіндегі сұраныстар. Мысалы, kaz_mus_instruments деректер қорынан домбыраның бағалары туралы акпарат алу керек болсын. Бұл деректер қорында аспалтардың атаулары және олардың бағасы екі түрлі кестеде орналасқан, яғни деректерді іріктеу instruments, name_instr екі кестеден жүзеге асырылады. Мұндай жағдайларда IN () функциясы пайдаланылады. Сұраныс синтаксисі келесідей:

<code>SELECT * FROM table1 WHERE id IN (SELECT id FROM table2);</code>
--

Енді осы сұраныс бойынша біздің мысалға қатысты SQL командасын жазамыз:

```
SELECT id name ins, price
FROM instruments
WHERE id name_ins
IN (SELECT id name ins FROM name instr WHERE name_instr = "домбыра");
```

Осы сұранысты келесідей де жазуға болады. Мысалы, алдыңғы сұраныстың синтаксисі төмендегідей қайта жазылуы мүмкін:

<code>SELECT table1.* FROM table1, table2 WHERE table1.id=table2:id;</code>

Сәйкесінше SQL сұраныстар келесідей болады:

```
SELECT instruments.id_name_ins, instruments.price
FROM instruments, name instr
WHERE instruments.id_name_ins=name_instr.id_name_ins
AND name_instr="Домбыра";
```

Екі жағдайда да сұраныс нәтижесі:

id	price
1	150000
1	160000

Өздерін білетіндей, бірінші нөмірлі домбыра бізде, ал нәтиже құралдың атын беру үшін келесі SQL командасы JOIN қажет.

JOIN командасы

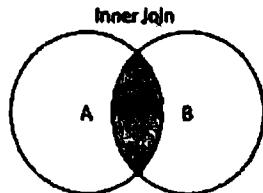
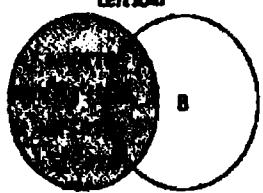
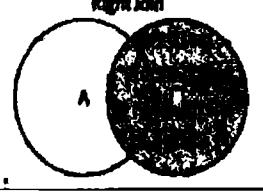
JOIN «біріктіру» мағынасын білдіреді. MySQL деректер көрінінде түрлі кестелерден өрістерді таңдау кезінде біріктіруге болады. Біріктіру уақытша кестелерді құрмай және бір сұраныспен ғана бірнеше кестелерден деректерді алуға мүмкіндік береді.

Нәтижеге қойылатын талаптарға байланысты MySQL үш типті біріктіруге мүмкіндік береді:

1. INNER JOIN (CROSS JOIN) – ішкі (қызылысу) біріктіру.
2. LEFT JOIN – сол жақты сыртқы біріктіру.
3. RIGHT JOIN – он жақ сыртқы біріктіру.

JOIN командасының синтаксисі

12-кесте.

<code>SELECT column_name(s) FROM table1 INNER JOIN table2 ON table1.column_name = table2.column_name;</code>	
<code>SELECT column_name(s) FROM table1 LEFT JOIN table2 ON table1.column_name = table2.column_name;</code>	
<code>SELECT column_name(s) FROM table1 RIGHT JOIN table2 ON table1.column_name = table2.column_name;</code>	

Енді JOIN синтаксисін пайдалана отырып, аспаптың коды емес, атауы (домбыра) шығу үшін алдыңғы сұранысты қайта жазамыз. JOIN үш типінің бірін таңдау үшін алдымен қандай кесте table1 және қандай table2 болатынын анықтау керек. Бізде table1 name_instr, table2 instruments болсын. Price өрісі table2-де болғандықтан, бұл өріс оң жағынан шығу үшін қажет, онда right JOIN синтаксисін таңдаймыз:

```
SELECT name_instr.name_instr, instruments.price
FROM name_instr
RIGHT JOIN instruments
ON instruments.id name_ins=name_instr.id name_ins
WHERE name_instr="домбыра";
```

Сұраныс нәтижесі:

name_instr	price
домбыра	150000
домбыра	160000

ТАҚЫРЫПТЫҚ ТЕСТ

1. Жазу косу үшін INSERT INTO командасы колданылады. Команданың жазылу синтаксисі:

CustomerId	CustomerFName	CustomerLname	Age
1	Artur	Baitursynov	15
2	Madrina	Saperbaeva	17
3	Indira	Mukanova	14
4	Ramazan	Bolat	20

- A) кесте_атауы (eric_атауы1, eric_атауы2, ...)
- B) eric_атауы(кесте_атауы1,кесте_атауы2, ...)
- C) кесте_атауы (жазба1,жазба2, ...)
- D) файл_атауы (eric_атауы1, eric_атауы2, ...)
- E) файл_атауы(кесте_атауы 1, кесте_атауы 2, ...)

2. INSERT INTO

Customer ('CustomerId', 'CustomerFName', 'CustomerLname', 'Age')

Берілген кесте ішінен _____(01, 'Artur', 'Baitursynov', 15) сұрыптал
көрсету үшін қолданылатын команда

- A) SELECT
- B) VALUES
- C) UPDATE
- D) FROM
- E) DELETE

3. Берілген кестеден Bolat-тың 20 жасын ауыстыру үшін жазылатын код:

CustomerId	CustomerFName	CustomerLname	Age
1	Artur	Baitursynov	15
2	Madrina	Saperbaeva	17
3	Indira	Mukanova	14
4	Ramazan	Bolat	20

- A) FROM Customer
SET Age = 18
WHERE CustomerID = 04;
- B) UPDATE Customer
SET Age = 18
WHERE CustomerID = 04;
- C) VALUES Customer
SET Age = 18
WHERE CustomerID = 04;
- D) DELETE Customer
SET Age = 18
WHERE CustomerID = 04;
- E) INSERT Customer

SET Age = 18

WHERE CustomerID = 04;

4. SELECT *CustomerID*, *CustomerFName*, ... қолданылады:

- A) Деректер базасындағы деректерді таңдау үшін
- B) Деректер базасындағы деректерді ауыстыру үшін
- C) Деректер базасындағы деректерді жою үшін
- D) Деректер базасындағы деректерді сұзгіден өткізу үшін
- E) Деректер базасындағы деректерді шарт қою үшін

5. AND және OR операторларын жазбаларды бірнеше шарт негізінде ...

үшін қолданады:

- A) ауыстыру
- B) сұзгіден өткізу
- C) шарт қою
- D) жою үшін
- E) таңдау

6. Кестеден жасы 18-дегі және Артур атты клиенттердің тізімін көрсетеу керек

<i>CustomerID</i>	<i>CustomerFName</i>	<i>CustomerLName</i>	<i>Age</i>
1	Artur	Baltursynov	15
4	Ramazan	Bolat	18

A) SELECT * FROM customer

INSERT Age='18' OR CustomerFName='Artur';

B) SELECT * FROM customer

AND Age='18' OR CustomerFName='Artur';

C) SELECT * FROM customer

UPDATE Age='18' OR CustomerFName='Artur';

D) SELECT * FROM customer

WHERE Age='18' OR CustomerFName='Artur';

E) SELECT * FROM customer

DELETE Age='18' OR CustomerFName='Artur';

7. Жасы 15-тегі және Артур атты клиенттердің тізімін көрсетеу керек.

<i>CustomerID</i>	<i>CustomerFName</i>	<i>CustomerLName</i>	<i>Age</i>
1	Artur	Baltursynov	15
2	Madiha	Saparbekaeva	17
3	Indira	Mukanova	14
4	Ramazen	Bolat	20

A) SELECT * FROM customer

WHERE Age='15' AND CustomerFName='Artur';

B) SELECT * FROM customer

WHERE Age='15' OR CustomerFName='Artur';

C) SELECT * FROM customer

UPDATE Age='15' OR CustomerFName='Artur';

D) SELECT * FROM customer

UPDATE Age='15' AND CustomerFName='Artur';

E) SELECT * FROM customer

INSERT Age='15' AND CustomerFName='Artur';

8. Барлық шарт ақырат болған жағдайда жазбадан көрсететін логикалық оператор:

A) OR

B) NOT

C) AND

D) NAND

E) FALSE

9. Шарт ақырат болмаған жағдайда жазбадан көрсететін логикалық оператор:

A) OR

B) NOT

C) AND

D) NAND

E) FALSE

10. Деректер базасынан деректерді таңдау үшін қолданылатын команда:

A) SELECT

B) VALUES

C) UPDATE

D) FROM

E) DELETE

11. Okushylar_11 кестесінен 16 жастағы және одан үлкен тұтынушылардың тегін енгізу қажет болса, SELECT командасы пайдаланылған мысалдың қатесі:

SELECT Age

FROM Okushylar_11

WHERE Age>'16';

A) SELECT жолына жазбалардың мәнін жазу керек

B) WHERE Age>= 16 деп жазу керек

C) FROM жолында деректер базасының атауын FirstDatabase деп жазу керек

D) SELECT командасы INTO көмекші сөзімен бірге қойылады

E) SELECT жолына жазбалар, ал FROM жолына өріс атауын жазу керек

12. Деректер базасындағы кестенің жаңарту үшін қандай сұранысты колдану керек?

A) \$sql="UPDATE...SET..."

- B) \$sql="UPDATE...WHERE..."
 C) \$sql="UPDATE... WHERE... SET..."
 D) \$sql="UPDATE...SET ...WHERE..."
 E) \$sql=" WHERE ...UPDATE...SET..."

13. Сұраныс mysql_multi_query()-дің атқаратын қызметі:

- A) Бірнеше жазбаны қосады
 B) Сұраныстардың дұрыс жасалғандығын тексереді
 C) Байланыспен сұраныс параметрлерін анықтайды
 D) Деректер базасындағы қажетті ақпаратты таңдайды
 E) Сұраныстағы жазбаларды жояды

14. Сұраныс mysql_query(\$conn,\$sql)-дің атқаратын қызметі:

- A) Бірнеше жазбаны қосады
 B) Сұраныстардың дұрыс жасалғандығын тексереді
 C) Байланыс пен сұраныс параметрлерін анықтайды
 D) Деректер базасындағы қажетті ақпаратты таңдайды
 E) Сұраныстағы жазбаларды жояды

15. Сұраныс \$conn->query(\$sql)==TRUE-дің атқаратын қызметі:

- A) Бірнеше жазбаны қосады
 B) Сұраныстардың дұрыс жасалғандығын тексереді
 C) Байланыс пен сұраныс параметрлерін анықтайды
 D) Деректер базасындағы қажетті ақпаратты таңдайды
 E) Сұраныстағы жазбаларды жояды

16. Сұраныс фрагментіндегі кеткен қатені тап.

```
$sql="INSERT INTO(first,name,email, age)
VALUES('Dilyara','Ameleva','Dilyaraameleva@gmail.com');
```

- A) Өрістер атауы тырнақшага алынбаған
 B) Кесте атауы жазылмаған
 C) Түсіндірме жоқ
 D) Өріс саны көп
 E) Жазба жолы бір жолда жазылмаған

17. Төмендегі сұраныстың орындалу нәтижесін анықта.

```
$result=mysqli_query($conn,$query);
```

- A) Деректер базасына сұраныс жасайды
 B) Деректер базасындағы жазбаларға сұраныс жасайды
 C) Деректер базасындағы кестеге сұраныс жасайды
 D) Деректер базасындағы бағанға сұраныс жасайды
 E) Деректер базасындағы жолға сұраныс жасайды

18. Сұранысты қайда жасауға болатынын анықта

- A) Командалық жолда
 B) Арнайы мәзірде
 C) MySQL интерфейсінде
 D) Графикалық блокта
 E) HTML+Excel редакторында

19. Келесі код \$sql = "SELECT*FROM okulyktar test ORDER BY id ASC"); белігі жазбаларды... сұрыптайтыңыз:

- A) Белгілі бір шарт бойынша
- B) Белгілі бір өріс бойынша
- C) Шектеулі саны бойынша
- D) Өсу реті бойынша
- E) Кему реті бойынша

20. Келесі код \$sql = \$mysqli->query("SELECT id FROM test LIMIT5"); белігі жазбаларды... сұрыптайтыңыз:

- A) Белгілі бір шарт бойынша
- B) Белгілі бір өріс бойынша
- C) Шектеулі саны бойынша
- D) Өсу реті бойынша
- E) Кему реті бойынша

6.4. Web-беттің деректер қорымен байланысын орнату

Деректер қоры бар web-беттер – деректер қорымен динамикалық байланысты қолдайтын және мәліметтерді қарастыруға, өндіреуге, сонымен бірге базага енгізуге мүмкіндік беретін диалогтық web-бет.

HTML-да веб-бетті жасайық. Біздің деректер қорда сакталғандықтан, алдымен бізге web-беттің деректер қорымен байланысын орнату қажет.

Ол үшін HTML-ден басқа қандай да бір web-программалау тілі қажет. PHP-ге мысалдар қарастырамыз.

PHP сценарийлері қарапайым мәтіндік файлдар. Демек оларды жасаған кезде Notepad+ сияқты мәтіндік файлдармен жұмыс істейтін кез келген белгілі құралдарды пайдалануға болады. Бірақ PHP-те жазылған программа жергілікті серверсіз іске аспайды.

Біздің жағдайда локальды сервер ретінде компьютер, ал web-сервер Apache қолданылады.

Сендердің компьютерлерінде OpenServer (Apache + PHP + MySQL) орнатылған, сондыктан деректер қоры бар web-бетті жасау үшін PHP скрипті тілімен жұмыс істеуге болады.

PHP және MySQL байланыстыру үшін, яғни PHP скриптінде MySQL деректер қорынан деректерді алу үшін келесілерді білу керек:

- 1) хост – MySQL деректер қоры серверінің адресі;
- 2) деректер қорының атауы;
- 3) пайдаланушының аты;
- 4) ДК-ға кіру үшін құпиясөз;
- 5) SQL командалар.

Тапсырма:

OpenServer пакетінде доменді жасандар:

- Сервердің түпкі буласы D:/OpenServer/domains/localhost орналасқан;
- Түпкі булаада доменнің жана атымен аталатын буласы құру (мысалы, test2.kz);

3. OpenServer-ді қайта іске қосу (немесе іске қосылмаған жағдайда іске қосу).

4. Notepad+-та php-скриптін құрамыз (сурет 5.6.1) және main.php деп аталатын файлды жергілікті сервердегі

D:/OpenServer/domains/localhost/test2.kz/

main.php буласында сактаймыз.

Келесі код main.php файлындағы kaz_mus_instruments деректер корына қосылады және producers кестесінен аспап жасаушыларды арестерімен іріктеуді орындайды. PHP коды деректер корынан алғынған барлық деректерді массивте сактайды.

```

<?php
header("Content-Type: text/html; charset=UTF-8");
echo <html>
<head>
<title>100страниц</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<? $con = mysql_connect("localhost", "root", "") ;
if (!($con))
die ('Did not connect: ' . mysql_error());
mysql_select_db("kaz_mus_instruments", $con);
mysql_query("SET NAMES utf8", $con);
$result = mysql_query("SELECT * FROM producers");
while ($row = mysql_fetch_array($result))
{
echo $row['id_producers'] .. $row['name'] .. $row['address'];
echo "<br />";
}
mysql_close($con);
?>
</body>
</html>

```

10-сурет. Деректер қоры мен web-беттің байланысын орнату үшін PHP коды.

Әр жол туралы толығырақ қарастырамыз

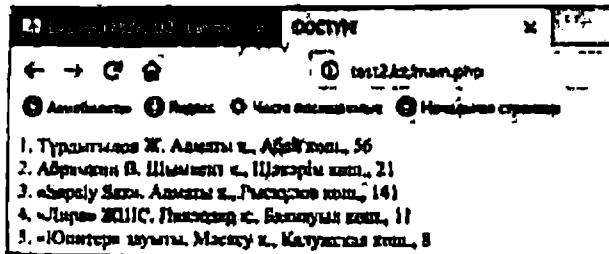
HTML тегтерден PHP кодты әрқашан <? ... ?> символдары арқылы ажыратып отыру қажет. Барлық PHP айнымалылар \$ символынан басталады.

13-кесте.

1.	<?	Бірінші жол web-серверге әрі қарай PHP коды керектігі туралы хабарлайды
2.	\$conn=mysql_connect ("localhost", "root", "")	Деректер қорына қосылу үшін web-серверге оның орналасқан жерін, пайдаланушының атын және құпиясөзін хабарлау қажет (OpenServer-ді орнатқанда, ағымдагы жағдай бойынша пайдаланушының аты – root, ал құпиясөз қойылмайды). Осы деректер бойынша фсопп аты берілетін ДК-на қосылу жолы құрылады. PHP mysql_connect() функциясы осы ақпаратты алады және оны ДКБЖ-не косу үшін пайдалануға тырысады.
3.	if(!\$conn) { die ('Did not connect:'; mysql_error()); }	Егер косу жүргізілмесе, PHP себебін сипаттай отырып, хабарлама жібереді, ал PHP кодын өндөу тоқтатылады.
4.	mysql_select_db("kaz_mus_instruments", \$conn);	ДКБЖ қосылуы сәтті өтті. Енді біз қажетті деректер қорын PHP-те көрсетуіміз керек.
5.	mysql_query("SET NAMES utf8", \$conn);	Бұл жол ASCII кодында ұсыныла алмайтын символдармен берілген деректерді серверге жіберу үшін пайдаланылады.
6.	\$result = mysql_query ("SELECT * FROM producers");	Деректер қоры таңдалған, байланыс жасалған, бірақ орындау үшін сұраныс жок. Сұраныс жасаймыз және оны mysql_query () функциясымен ДКБЖ-не жібереміз. Барлық қайтарылатын жазбалар \$result массивінде сақталады.
7.	while (\$row = mysql_fetch_array(\$result)) {	Енді PHP синтаксисі web-бette \$result мәндерін шығару үшін қолданылады. While циклі деректердің сонына жеткенше жазбаларды таңдайды.
8.	echo \$row['id_producers']. "".\$row['name']." ". \$row['address']; echo " "; }	PHP echo екі командасты әр жазбаның id_producers, name не address өрістерін web-бетке шығарады. Шығарылатын жолдар HTML-дің тегімен бөлінеді.
9.	mysql_close(\$conn);	Барлық деректер шығарылғаннан кейін,

		ДКБЖ-не қосылу жабылады (терминалдағы жұмыс сеансының аяқталуына үксас).
10	?>	PHP сценарийі аяқталды.

main.php файлы дайын, енді оны браузер арқылы ашамыз. Егер бет суретте көрсетілгендей болса, компьютерде барлық OpenServer қызметтері дұрыс орнатылған.



11-сурет. main.php файлының нәтижесі

6.5. Формалар

Форма – деректер қорындағы немесе сұраныстағы кестелерде сәкталатын акпаратты енгізууді, өндеуді және көрсетуді жөнілдететін құрал.

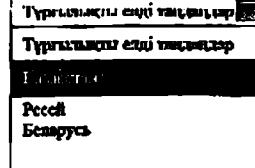
Форма – акпаратты көрсету, деректерді енгізу және өзгерту, белгілі бір әрекеттерді орындау үшін пайдаланылатын функционалды элементтер жиынтығы. Функционалды элементтердің мысалдары: батырма, мәтіндік өріс, тізім және т.б.

Біз HTML формаларына тоқталайық, өйткені сөндер HTML-ді білесіндер және web-бетті деректер қорымен байланыстыра аласындар.

Форманың барлық коды <form> элементінің ішінде орналасады. <input> элементі – форманың ең маңызды элементі. <input> элементі type атрибутына байланысты бірнеше жолмен көрсетілуі мүмкін. Кейбір мысалдар кестеде көлтірілген.

Форма жасауға қажетті негізгі HTML тәгтер

14-кесте.

Түрі / Сипаттама	Код мысалы	Браузердегі нәтиже
<input type="text"> Бір жолды мәтінді енгізу өрсін анықтайды	<form> Логин: <input type="text" name="login"> Құпиясөз: <input type="text" name="password"></form>	Логин <input type="text"/> Кұпиясөз: <input type="text"/>
<input type="radio"> Косқышты анықтайды (опциялардың біреуін тандау үшін)	<form> <input type="radio" name="gender" value="male" checked> Ер <input type="radio" name="gender" value="female"> Әйел </form>	<input checked="" type="radio"/> Ер. <input type="radio"/> Әйел
<input type="submit"> Жіберу батырмасын анықтайды (форманы жіберу үшін)	<form method="post" action="action_page.php"><input type="submit" value="OK"></form>	
<select>...</select> Ашылмалы тізім жасайды	<select name="country"><option>Түргылықты елді тандандар</option><option value="Қазақстан"> Қазақстан</option><option value="Ресей">Ресей </option><option value="Беларусь"> Беларусь</option></select>	

6.6. Есептер

Есеп – экранға, баспаға немесе файлға шығарылатын деректердің форматтаған түрі.

Есептер деректер қорынан қажетті деректерді алғып, оларды ыңғайлы турде ұсынуға, сонымен қатар деректерге талдау жасауда мүмкіндік береді.

Өндірушілер бойынша музыкалық аспаптар (Күні)

Өндіруші _____

№	Аспап атауы	Бағасы	Жекелдік
_____	_____	_____	_____
_____	_____	_____	_____

Барлығы:

12-сурет. Өндірушілер бойынша музыкалық аспаптар есебінің макеті

Есептер web-бет түрінде болады. Алдымен web-беттің безендірілмеген түрінің PHP кодын жазайык. Мәліметтерді деректер қорынан алу үшін JOIN командасының көмегімен курделі сұраныстарды пайдаланамыз:

Курделі сұранысты бірнеше жолға жазудың синтаксисіне назар аударындар:

```
$sSQL1= "SELECT instruments.price, instruments.sales, name_instr.  
name_instr FROM instruments";  
$sSQL1.= " LEFT JOIN name_instr ON instruments.id_name_ins-  
=name_instr.id_name_ins";  
$sSQL1.= " WHERE instruments.id_producers=$row[id_producers]";
```

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE test-report SYSTEM "http://www.softwaretestinghelp.com/test-report.dtd">
</head>
</body>
</html>

```

Вот каким образом можно проверить наличие тестов в проекте:

```

1. В меню "File" выберите "Open" и откройте нужный проект.
2. В меню "Tools" выберите "Test Manager".
3. В списке "Test Cases" убедитесь, что там присутствуют ваши тесты.
```

Помимо этого, можно использовать инструменты для автоматизированного тестирования, такие как Selenium WebDriver или Appium для мобильных приложений. Эти инструменты позволяют автоматизировать выполнение тестов и обнаруживать ошибки.

13-сурет. Өндірушілер бойынша музикалық аспаптар есебі кодының бөлігі

Өндірушілер бойынша музикалық аспаптар

Өндіруші: Тұрдыгұлов Ж.

1. Домбыра 200 000

Барлығы: 200 000

Өндіруші: Абрамкин В.

I. Кобыз 150 000

2. Кобыз 110 000

Барлығы: 260 000

Өңлірүші: «Sapaly Saz»

Сандруш: «Zapaly въз»

2. Помбыра 150 000 28%

2. Домовіра 150 000
3. Шертер 90 000

3. Шерпър 90 000
Бартичън 255 000

Чинірчукі: Пирас ЖІІС

«Андрюш. Лира» Ж
1. Жетіған 180 000

2. Шартар 100 000 40%

Барлығы: 280 000

Өндіруші: «Юпитер» зауыты

1. Сыбызы 20 000 30%

2. Дабыл 60 000

Барлығы: 80 000

14-сурет. Өндірушілер бойынша музыкалық аспаптар есебінің форматталмаған түрі

Ал web-бетті форматтау үшін HTML тегтерін пайдалану кажет. Есеп көрсетілгендей болу үшін деректерді кестенің ішіне толтырамыз.

Өндірушілер бойынша музыкалық аспаптар

Өндіруші: Тұрсынгүлов Ж.

№	Аспап атауы	Бағасы	Жекеілдік
---	-------------	--------	-----------

1. Домбыра 200000

Барлығы: 200000

Өндіруші: Абрамжол В.

№	Аспап атауы	Бағасы	Жекеілдік
---	-------------	--------	-----------

1. Кобыз 150000

2. Кобыз 110000

Барлығы: 260000

15-сурет. Кесте қосқаннан кейінгі reports.php файлының браузердегі түрі

6.7. BigData

«Большие данные» терминін Nature журналының редакторы Клиффорд Линч 2008 жылы енгізді.

Кеп жылдар бойы үйымдарда үлкен көлемді деректер жинақталады (web-журналдар, бейнеказбалар, мәтіндік құжаттар, машина коды немесе геокеңістік деректер). Мұның бәрі түрлі сақтау құрылғыларында сакталады, кейде тіпті үйимнан тыс жерде. Нәтижесінде корпорациялар өз деректерінің үлкен көлеміне қол жеткізе алады. Осы деректер арасындағы өзара байланысты орнату және олардың негізінде маңызды қорытындылар жасау үшін қажетті құралдары болмауы мүмкін. Акпаратты талдаудың дәстүрлі әдістері үнемі жаңартылып отыратын деректердің үлкен көлеміне ілесе алмайды. Бұл нәтижесінде үлкен деректердің технологияларына әкелді.

15-кесте.

Сипаттамасы	Дәстүрлі деректер қоры	Үлкен деректер қоры
Ақпарат көлемі	Гигабайттан терабайтқа дейін	Петабайттан эксабайтқа байтқа дейін

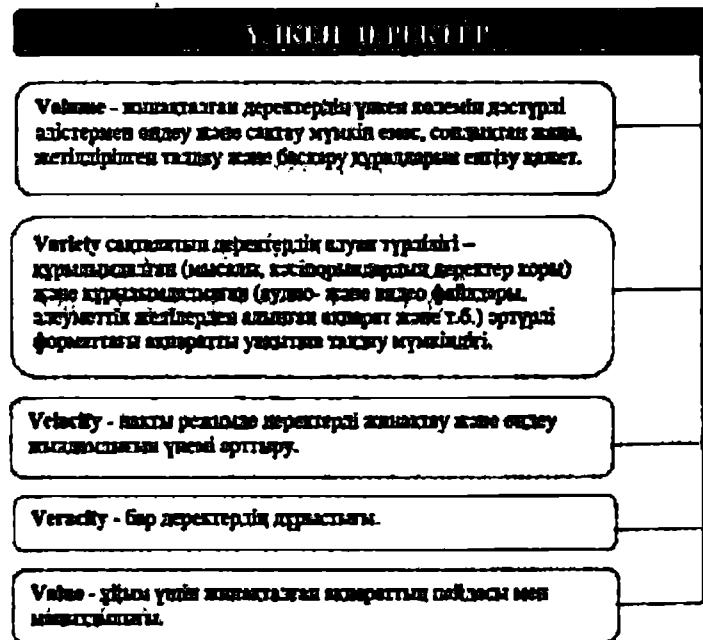
Сақтау тәсілі	Орталықтандырылған	Орталықтандырылмаған
Деректердің күрылымы	Күрылымдалған	Жартылай күрылымдалған немесе күрылымдалмаған
Деректерді сақтау және өндөу моделі	Тік модель	Көлденен модель
Деректердің өзара байланысы	Күшті	Әлсіз

Улken деректер әртүрлі арналар арқылы әр секунд сайын берілетін, әдетте онлайн, улken көлемді ақпараттарға жатады. Бұл деректер өте улken, күрделі және динамикалық, оларды дәстүрлі күралдармен пайдалануға келмейді. Технологиялық прогрестік арқасында бұл деректерді енді түсіндіруге және талдауға болатын болды. Үкімет, деңсаулық сақтау, бөлшек сауда өндірісі, электрондық оқыту және деректермен басқарылатын басқа да салалар үшін улken артықшылықтары бар.

Әдетте улken деректер үш дереккөзден түседі:

- Интернет (әлеуметтік желілер, форумдар, блогтар, БАҚ және басқа сайттар);
- құжаттардың корпоративтік мұрагаттары;
- датчиктер, аспаптар, камералар және басқа да құрылғылардың көрсеткіштері.

Жинақталған деректер көлемін әрдайым улken деректер қатарына жатқызуға болмайды. Улken деректер саласын сипаттайтын 5 белгісі бар:



16-сурет. Улken деректердің белгілері

Барлық ірі банктер мен телеком-нарық көшбасшылары Big Data технологиясын пайдаланады. Сондай-ақ медиа мен Интернет-компанияларға да қатысты. Тіпті мемлекеттік сектор да. Бүгінгі таңда Big Data-ны пайдаланудағы көшбасшылардың бірі – өнеркәсіптік кәсіпорындар мен өндіруші секторлар, мысалы: металлургия және мұнай өндіру секторы.

Артық жүктеме, әсіресе, Интернет-компанияларға қатысты. Google күніне петабайттан көбірек деректерді өнддейді. Facebook-те сағат сайын 10 миллионнан астам жаңа фотосуреттерді жүктейді. Адамдар «сұнайды» батырмасын басады немесе күніне үш миллиардтан астам пікір жазады, сандық қолтаңба калдырады, ол арқылы компания пайдаланушылардың қалауын зерттейді. Ал Google компаниясының YouTube қызметінің 800 миллион ай сайынғы пайдаланушылары секунд сайын ұзақтығы бір сағаттан астам бейне жүктейді.

Big Data-ның классикалық түрі Amazon сауда аланды немесе оның AliExpress (қытай) нұсқасы. Қарап шығуды, сатып алады, қызығушылықтарды талдай отырып, программа заңдылықтарды қадағалайды және пайдаланушыларды нақты қызықтыра алатында ұсыныстар береді. Мұндай құралдың арқасында сату деңгейі айтарлықтай артады.

«Kaspi.kz» банк клиенттерін идентификациялау үшін фотосуреттерді талдайтын және алаяқтықтың алдын алатын жүйені енгізді. Жүйе 2014 жылы енгізілді. Жүйе негізінде «компьютерлік көрү» арқасында базага фотосуреттерді web-камерадан келіп түсетін фотосуреттермен салыстыру жүзеге асады. Сонымен үлкен деректердің артықшылықтары мән кемшіліктері де жоқ емес.

Үлкен деректердің артықшылықтары мен кемшіліктері

16-кесте.

Артықшылықтары	Кемшіліктері
<i>Шешім қабылдаудың тиімділігі.</i> Аналитика бизнес-шешімдер қабылдайтын тұлғаларға олардың компанияларына бәсекелесуге және дамуға көмектесу үшін қажетті ақпаратты бере алады.	<i>IT мамандарына қажеттілік.</i> IT саласындағы маманды жалдау немесе оқыту шығындарды едәуір арттыру мүмкін, ал үлкен деректермен жұмыс істеу дағдыларын қалыптастыру процесі айтарлықтай уақыт алуы мүмкін.
<i>Өнімділікті арттыру.</i> Үлкен деректермен жұмыс істеу үшін қазіргі заманғы құралдар талдау жүргізушилерге көбірек деректерді тез талдауға мүмкіндік береді, бұл олардың жеке өнімділігін арттырады.	<i>Деректер сапасы.</i> Үлкен деректерді пайдаланбас бұрын, пайдаланылатын ақпарат дәл, өзекті және талдау үшін тиісті форматта екенине көз жеткізу қажет.
<i>Клиенттерге қызмет көрсетуді жақсарту.</i> Әлеуметтік желілер және клиенттермен қарым-қатынасты басқарудың басқа жүйелері кәсіп-	<i>Күпиялық.</i> Компанияның үлкен деректерін сақтау қоймаларында ақпараттың көп бөлігі күпия немесе жеке. Бул фирмада

<p>орындарға өз клиенттері туралы кең ақпарат береді. Бұл деректерді клиенттерге жақсы қызмет көрсету үшін пайдалану ыңғайлы.</p>	<p>деректерді өндөу және сақтау кезінде салалық стандарттарға немесе мемлекеттік талаптарға сәйкестігін қамтамасыз ету талап етілуі мүмкін дегенді білдіреді.</p>
<p><i>Алаяқтықты анықтау.</i> Үлкен деректерді талдау жүйесі зандылықтар мен ауытқуларды тануға қабілетті. Мисалы, банктер мен несие карталарын шығаратын компаниялар ұрланған несие карталарын немесе алаяқтықпен сатып алуды анықтауға мүмкіндік береді.</p>	<p><i>Киберқауіпсіздік тәуекелдері.</i> Компаниялардың үлкен деректерді, әсіресе, қупия деректерді сақтауы, оларға зиянкестер тарапынан қауіп төндіруі мүмкін.</p>
<p><i>Табыстың өсуі.</i> Ұйымдардың үлкен деректерді қолдану нәтижесі көп жағдайда табыстың өсуіне мүмкіндік береді.</p>	<p><i>Жылдам өзгерістер.</i> Бұл технология тез өзгереді.</p>
<p>Икемділікті арттыру. Көптеген ұйымдар IT және бизнес саласындағы өз жұмысын жақсы үйлестіру үшін үлкен деректерді пайдаланады. Олар өз бизнес стратегияларын тез және жеке өзгерістерді қолдау үшін аналитиканы пайдаланады.</p>	<p><i>Жабдықта қажеттілік.</i> Кәсіпорындардың жабдықпен, техникалық қызмет көрсетумен және ілеспе қызметтермен (деректерді сақтауға арналған кеңістік, желінің еткізу қабілеті, есептеу ресурстары және т.б.) байланысты едәуір шығындары болады.</p>
<p><i>Үлкен инновациялар.</i> Көптеген басшылар, ең алдымен, инновацияга арналған құрал ретінде талдаушыға қаржы салады. Талдау нәтижесі бойынша бәсекелестері жоқ идеялар пайда болса, олар жаңа өнімдер мен қызметтермен басқа нарыққа шығуға мүмкіндік болады.</p>	<p><i>Жүйелердің интеграциясының күрделілігі.</i> Көптеген жыл бойы келе жатқан кәсіпорындардың көпшілігі деректерді түрлі қосымшалар мен жүйелерде жинаиды. Барлық осы бытыранқы деректер көздерін біріктіру үлкен деректермен жұмыс істей уақыты мен шығындарын арттырады.</p>

ТАҚЫРЫПТЫҚ ТЕСТ

1. Select id, date, customer_name from Orders – бұл сұраныстардан қандай деректерді аламыз:

A) Тұтынушы аты, нөмірлері мен күні бар, барлық тапсырыстар сұрыпталмаған

B) Ешқандай, сұраныс қате құрылған

C) барлық тапсырыстың нөмірлері мен күні, тұтынушы аттары бар барлық бағандар бойынша сұрыпталған, Orders сезі бар

D) Бірінші баған бойынша тұтынушы аттары бар барлық тапсырыстың нөмірлері мен күндегі сұрыпталған

E) Жол бойынша тұтынушы аттары бар барлық тапсырыстың нөмірлері мен күндегі сұрыпталған

2. Тапсырыстар кестесіндегі барлық мәнді көрсету үшін ORDER сұранысы қалай жазылады:

A) select ALL from Orders;

B) select % from Orders;

C) select * from Orders;

D) select *.Orders from Orders;

E) select &.Orders from Orders;

3. Сұраныста қате бар ма? select id, date, customer_name from Orders where customer_name = Mike;

A) Сұраныс дұрыс құрылған

B) Mike-тың нақшада жазу керек ‘Mike’

C) Сұраныстан артық өрістерді алғып тастау керек

D) Where жолын from-мен аудостыру керек

E) Select жолын from-мен аудостыру керек

4. Берілген сұраныс көрсетеді: select * from Orders where date between '2022-01-01' and '2022-12-31'

A) 2022 жылғы 01 қаңтар айын қоспағанда, 2022 жылы жасалған тапсырыстар бойынша барлық деректер

B) 2022 жылғы 01 желтоқсан айын қоспағанда, 2022 жылы жасалған тапсырыстар бойынша барлық деректер

C) 2022 жылы жасалған тапсырыстар бойынша барлық деректер

D) Ешқандай, сұраныс қате құрылған

E) 2022 жылы жасалған тапсырыстар бойынша барлық деректер дұрыс емес

5. Сұраныста не дұрыс емес? select id, date from Orders where seller_id = NULL;

A) Барлығы дұрыс, сұраныста сатушылар көрсетпеген тізімде жоқ барлық тапсырыстарды көрсетеді

B) NULL-ді тыңнақша ішінде көрсету керек

C) NULL тенестірілмін IS операторымен ғана жазу керек

- D) NULL тәсестірлімін ON операторымен ғана жазу керек
 E) NULL тәсестірлімін SELECT-пен алмастыру керек

6. AND және OR операторының орындалуы

- A) Бірінші AND, содан соң OR
 B) Бірінші OR, содан соң AND

C) AND және OR операторының орындалуы, басында тұрған операторға байланысты

- D) AND және OR операторы бір уақытта орындалады

E) AND және OR операторының орындалуы, сонында тұрған операторға байланысты

7. Сұраныс нені көрсетеді: select DISTINCT seller_id,order by seller_id from Orders;

- A) Сатушылар ID өсу бойынша сұрыпталған

- B) Сатушылар ID кему бойынша сұрыпталған

C) Ештеңе, сұраныс дұрыс құрылмаган ORDER BY сұраныс соңында қойылады

- D) Сатушылар ID-ді select алмастыру қажет

- E) Select-ті from-мен алмастыру қажет

8. LIKE операторы '_ символымен мыңа сұраныста не жасайды? : select * from Orders where customer_name like 'mik_';

- A) mik-тан басталатын 4 символдан тұратын барлық атауды тауып береді

B) символдар санына қарамастан mik-тан басталатын барлық атауды тауып береді

- C) mik-қа тен барлық сұрауларды тауып береді

D) сұраныс дұрыс құрылмаган, like операторымен ештеңе қосылып жазылмайды

- E) where операторы жазылған жерде like операторы жазылмайды

9. CONCAT функциясы жазылған дұрыс мысалды көрсетіңіз:

- A) select concat = index and city from Orders;

- B) select concat IN ('index', 'city') from Orders;

- C) select from Orders ('index', " ", 'city') concat;

- D) select concat('index', " ", 'city') from Orders;

- E) дұрыс мысал көрсетілмеген

10. ROUND функциясы жазылған дұрыс мысалды көрсетіңіз:

- A) select id, price * discount AS total price from Orders ROUND (2);

- B) select id, price * discount ROUND (2) AS total price from Orders;

- C) select id, ROUND (price * discount, 2) AS total price from Orders;

- D) select id, price * ROUND (2) AS total price from Orders;

- E) дұрыс мысал көрсетілмеген

11. Сұраныс нені көрсетеді: select id from Orders where year (date) > 2022;

- A) 2022 жылға дейінгі жасалған тапсырыстар нөмірі

- B) 2022 жылында жасалған тапсырыстар нөмірі

- C) Бірегей әдемі тапсырыстар нөмірі

D) 2022 жылынан кейін жасалған тапсырыстар нөмірі
 E) дұрыс жауап көрсетілмеген

12. LIMIT не үшін қолданылған: select * from Orders limit 10;

- A) 10 саны бар барлық тапсырысты көрсетеді
- B) Сұраныстағы алғашқы 10 жазбаны көрсетеді
- C) Сұраныстағы рандомды 10 жазбаны көрсетеді
- D) 10 саны бар барлық тапсырысты көрсетпейді
- E) Бұндай оператор жоқ

13. GROUP BY функциясымен дұрыс жазылған сұранысты көрсөтіңіз:

- A) select count(*) from Orders GROUP seller_id;
- B) select seller_id, count(*) from Orders GROUP seller_id;
- C) select seller_id, count(*) from Orders GROUP BY seller_id;
- D) select count(*) from Orders GROUP ON seller_id;
- E) select count(*) from Orders GROUP *BY seller_id;

14. Сұраныс нени көрсетеді: select seller_id, count(*) from Orders GROUP BY seller_id HAVING seller_id IN (2,4,6);

- A) сатушылардың топтастыруымен 2, 4 және 6 тапсырыстар саны жинақталады
- B) 2, 4 және 6 тапсырыстар саны бар, сатушылар саны
- C) Сұраныс дұрыс құрылмаған, HAVING топтастыруға дейін жазылады
- D) Сұраныс дұрыс құрылмаған, шарт құру үшін WHERE жазылады
- E) 2, 4 және 6 тапсырыстар саны бар, сұраныс саны

15. Ең қымбат құны бар тапсырыс туралы ақпаратты көрсететін ішкі сұрауды пайдаланып жақсы жазылған сұраудың мысалын таңдаңыз

- A) select * from Orders where price = (select big(price) from Orders)
- B) select * from Orders where price = max
- C) select count(*) from Orders
- D) select * from Orders where price = (select max(price) from Orders)
- E) select * from Orders where price = (max(price) select from Orders)

16. JOIN дегеніміз?

- A) Біріктіру операциясы
- B) Толтастыру операциясы
- C) Қосу операциясы
- D) Құру операциясы
- E) Сұрыптау операциясы

17. Төмөнде көрсетілген JOIN операциясына жатпайтыны:

- A) LEFT JOIN
- B) RIGHT JOIN
- C) INNER JOIN
- D) TRUE JOIN
- E) OUTER JOIN

18. JOIN мен құрылған дұрыс мысалды таңдаңыз; берілген сұраныс тапсырыс ID, тапсырыс берушінің атын және сатушыны анықтап береді

A) select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN ON Sellers AND Orders.seller_id = Sellers.id;

B) select id, AND customer_name AND seller_id from Orders LEFT JOIN Sellers ON seller_id = id;

C) select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN Sellers ON Orders.seller_id = Sellers.id;

D) select Orders.id, Orders.customer_name, Sellers.id from Orders JOIN Sellers WHEN Orders.seller_id = Sellers.id;

E) select Orders.id, Orders.customer_name, Sellers.id from Orders RIGHT JOIN Sellers WHEN Orders.seller_id = Sellers.id;

19. SQL-да берілгендер типі жок:

A) VARCHAR

B) STRING

C) CHAR

D) TEXT

E) DATE

20. CHAR және VARCHAR айырмашылығы?

A) Екеуі бірдей

B) VARCHAR болған емес

C) CHAR берілгендер типі, ал VARCHAR ішкі тип

D) CHAR максималды ұзындыққа жолдағы бос орындарды толтырады, ал VARCHAR жол ұзындығын сақтауга арналған жадтағы орынды алады

E) Даұрыс жол көрсетілмеген

VII БӨЛІМ. ЖАСАНДЫ ИНТЕЛЛЕКТ

7.1. Жасанды интеллект, машиналық оқыту

Жасанды интеллект компьютерлік ғылымдар жүйесіндегі бағыт ретінде XX ғасырдың 60 жылдарының ортасында адамның зияткерлік функцияларын компьютерлік модельдеуге байланысты пайда болды. Бірінші интеллектуалдық жүйе теориялық дәлелдеуге арналған «Logic Theorist LT» программасы болды. Оның жұмысы 1956 жылдың 9 тамызында көрсетілді. Программаны құруға атақты ғалымдар А. Ньюэлл, А. Тьюринг, С. Шенон, Г. Саймон және тағы басқалар қатысты. Сол кезден бастап интеллектуалдық деп аталатын жасанды интеллект саласында көптеген компьютерлік жүйелер дамыды.

Заманауи интеллектуалдық технологиялар – деректер өндірілетін, талданатын, түсіндірілетін және пайдаланылатын компьютерлік жүйелер.

ЖИ (жасанды интеллект) саласы келесі міндеттерді шешетін программаны құруға бағытталған:

Жасанды интеллект (ЖИ; ағыл. artificial intelligence, AI) – дәстүрлі түрде интеллектуалдық деп саналатын, адам қызметін компьютерлік модельдеу туралы ғылым мен технология.

- арнайы математикалық аппаратты қолдану барысында бейнелер мен сәйлеуді қабылдау және тану;
- деректерді талдау және жалпылау арқылы өзін-өзі оқыту. Бұл бағытқа компьютерлік деректер қорындағы заңдылықтарды іздеу жүйелерін енгізу;
- интеллектуалды роботтарды басқару;
- машиналық шығармашылық және компьютерлік ойындарды қамту.

Бұл саладағы барлық қындықтар адамның интеллектуалдық қызметтінің принциптері мен шешімді қабылдау процестерінің жеткіліксіз зерттелгендейгіне байланысты. Соңдықтан жасанды интеллект саласындағы жұмыстар психология, физиология және лингвистика бөлімдері бойынша тиісті зерттеулермен тығыз байланысты болуы керек.

ЖИ қолдану салалары:

Жасанды интелекттің қолдану салалары	
Ауыл шаруашылығы	<ul style="list-style-type: none"> • Егін жинауды автоматтандыру • Жылышай шаруашылығын дамыту
Өнеркәсіп	<ul style="list-style-type: none"> • Шығынды азайту • Өндіріс сапасын көтеру
Бизнес	<ul style="list-style-type: none"> • Халыққа қызмет көрсету • Алайқтыққа жол бермеу
Транспорт	<ul style="list-style-type: none"> • Жолдағы қауіпсіздік • Жолда көптелістің алдын алу
Күнделікті өмірде	<ul style="list-style-type: none"> • Өмір сапасын жақсарту • Тұргын үй апартарының алдын алу
Қылмыспен курес	<ul style="list-style-type: none"> • Қылмыстың алдын алу

	<ul style="list-style-type: none"> • Эділ жаза қолдану
Медицина	<ul style="list-style-type: none"> • Диагностика • Телемедицина
Білім беру	<ul style="list-style-type: none"> • Эділ бағалау • Білім алушының потенциалын жүзеге асыру

1-сурет. Жасанды интеллекттің қолдану салалары

Өнеркәсіпте және білім беруде жасанды интеллекттің қолдану

Әр адамның өзіндік оқу қарқыны болады. Адаптивті оқыту жүйесі мұғалімдердің жұмысын азайтып, әр окушының мәліметті толық игеруін қамтамасыз етеді. Адаптивті оқыту жүйесінің мысалы ретінде дайындалып жатқан Third Space Learning платформасын алуға болады. Бұл жүйе мәліметті игерудің тиісті жылдамдығын таңдал, окушылардың іс-әрекеті мен аудиториядағы жағдайға жауап береді. Егер жағдай мұғалімнің араласуын қажет етсе, жүйе оны шақырады.

Автоматты түрде бағалау окушылардың білімін бағалап, жауаптарын талдайды және кері байланыс орнатып, жеке оқу жоспарларын құрады. Сондай-ақ ол – жазылған эссе үшін баға қоятын мұғалімнің іс-әрекетін қайталайтын, жасанды интеллектке негізделген арнайы компьютерлік программа.

Аралық оқыту сіздің бұрын ұмытып қалған біліміңізді тексереді. ЖИ (жасанды интеллект) сенің нені және қашан билетініңде қадағалап отырады. Осының нәтижесінде ол ұмытылған ақпаратты тауыш және оны қайталауға көнесп береді.

Виртуалды көмекшілер. Қазіргі таңда мұғалімдерде көмекшілер бар. ЖИ (жасанды интеллект) орнатылған компьютерлердің арқасында олар окушылардың сұрақтарына дәл және жылдам жауап бере алады.

Proctoring немесе Proctored Test – тестілеуші адамның шынайылығын қамтамасыз ететін және тестілеу кезінде проктор арқылы алдаудың алдын алатын механизм.

ЖИ (жасанды интеллект) өндіріс кәсіпорындарының барлық деңгейінде қолданыла алады:

Жобалау деңгейінде: жаңа өнім жасау деңгейін жоғарылатып, қосалқы бөлшектерге қойылатын талаптарды талдау барысында жеткізушілерді бағалау мен тандауды автоматтандыру.

Ондіріс деңгейінде: кескінді анықтау функциясы және диалогтік интерфейсті қолдану арқасында қателіктердің санын азайту, бизнес процестерді жетілдіру, жаңадан жасау процестерін онайлату үшін жасанды интеллект қолданылады.

Логистика деңгейінде: көліктердің бағытын жоспарлауын жетілдіру, шикізатты жеткізу уақытын қысқарту, интерактивті байланыс арқылы тұтынушылармен және жеткізушілермен өзара әрекеттесуді жақсарту үшін қолданылады.

Дамыту деңгейінде: бағаны басқару кезінде қолдау және техникалық қызмет көрсету көлемін болжайды.

Жасанды интеллекті өндірісте қолданудың жалпы тенденциясы бойынша оны қолданудың нәтижесінде өндіріс адамның жұмыс істеге іс жүзінде жойылғанға дейін автоматтандырылады. Атап айтқанда, LG компаниясы 2023 жылы зауыт ашуды жоспарлап отыр, мұнда барлық процестер – шығын материалдарын сатып алу мен оларды жөнелтүге дейінгі бақылау жасанды интеллект көмегімен жүзеге асырылады. Сондай-ақ ЖИ (жасанды интеллект) әдетте адам бақылайтын процестерді, жабдықтың тозуын, белгіленген жоспарлардың орындалуын және басқа да факторларды басқарады.

Кибернетика – жасанды және биологиялық жүйелердегі басқару механизмдерін зерттейтін техника саласы. Рединг университетінің кибернетика профессоры Кевин Уорвик – киборг, жартылай адам, жартылай машина. Оның жүйке жүйесіне шағын электрондық құрылғылар қосылған. Бірінші имплант оны гимарат ішінде бақылауга мүмкіндік береді, сондай-ақ есіктерді ашуға және жақындаған кезде жарықты қосуға болады. Екінші оның жүйке жүйесін интернетпен байланыстырады. Ал үшінші имплант Атлант мұхитының ар жағында болсан да, роботтың қолын басқаруға мүмкіндік береді.

Қазіргі кезде «жасанды интеллект» термині есептеу техникасы әлемінде және жаңа технологияларды әзірлеуде өзекті. Алайда осы үлкен технологиялық прогрессе тап болған кезде, машина логиканы қалай үйренеді және дамытады деген сұрақ туындаиды. Осының бәрі машиналық оқытуудың көмегімен жасалады.

Машиналық оқыту – адамның көмегінсіз оқыту, ойлау және шешім қабылдауға мүмкіндік беретін, үлгілер мен деректер құрылымын талдау және түсіндіруге бағытталған компьютерлік ғылым саласы. Басқаша айтқанда, бұл – компьютерлерге өздігінен тапсырмаларды орындауға мүмкіндік беретін оқыту түрі.

Машиналық оқыту тұжырымдамасы үлкен жетістікпен 1950 жылдары «деректерді талдау» деп аталатын деректер ғылымының қозғалтқышы ретінде жасалды. 1956 жылы IBM инженері Артур Самуэл дойбы үткіп алу мақсатында IBM 701 компьютерін менигеріп, «машиналық оқытудың» алғаш рет қолданды.

Жаңа компьютерлік технологиялардың арқасында қазіргі машиналық оқыту өткен замандағы машиналық оқытудан озып түсті. Ол компьютерлер тапсырмаларды белгілі бір «программасыз үйреніп», «орындаі алады» деген теориядан пайда болды. Модельдер жаңа деректер әсеріне ұшырайды және өздігінен бейімделе алатындықтан, машиналық оқытуудың итерациялық аспектісін білу маңызды. Олар алдынғы есептеулерден сенімді, қайталанатын шешімдер мен нәтижелер алуды үйренеді.

Машина өздігінен алдын ала болжам жасай алады, есте сактайды және ең жақсысын таңдай алады. Бірақ ол жаңа нәрсе құра алмайды, бірден тапсырмадан асып кете алмайды. Сонымен қатар машина барлық адамзатты күрта алмайды.

Машина жасай алады

- Болжам жасау
- Есте сактау
- Ең жақсысын таңдау

Машиналық оқытудың негізгі мақсаты – келген деректер негізінде нәтижелерді болжау. Келген деректер негұрлым әртүрлі болса, машинаға үлгіні табу оңай және нәтиже дәлірек болады.

Машинаны оқыту үшін үш нәрсе қажет:

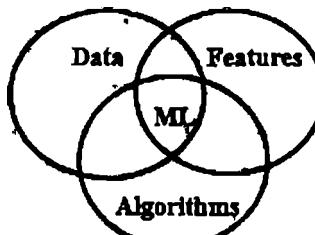
1. **Деректер.** Егер біз пайдаланушыға ұсыныстың топтамасын жинағымыз келсе, онда оның қаралу тарихы қажет. Пощадағы спамды анықтағымыз келсе, бізге осындай хаттардың мысалдары қажет. Деректерді алудың екі негізгі әдісі бар: қолмен және автоматты. Қолмен жиналған деректердің қателіктепі азырақ болады. Бірақ оны жинауга көп уақыт кетеді. Деректер құнды, бірақ қымбат болады. Автоматты тәсіл арқылы табуға болатын деректерді жинайсын. Кейбір ірі компаниялар өз пайдаланушыларын деректерді тегін белгілеу үшін пайдаланады.

Жақсы деректерді жинау (әдетте деректер базасы деп аталады) өте киын. Компаниялар тіпті өз алгоритмдерінің құпиясын ашса да, деректер қорын ешқашан көрсетпейді.

2. **Касиеттер.** Олар белгілер және сипаттамалар деп те аталады. Бұл автомобиль жүгірісі, пайдаланушының жынысы, науқан бағасы, мәтіндегі сөздердің пайда болу жиілігі болуы мүмкін. Машина неге қарайтынын накты білуі керек. Деректер кестелерде сақталса, біз олардың касиеттерін бағандарының атаулары арқылы оңай ажыратса аламыз. Бірақ бізде 100 ГБ иттердің фотосуреті болса ше? Біз әр пиксельді белгі ретінде қарастыра алмаймыз. Сондыктан дұрыс функцияларды, яғни қасиеттерді таңдау машиналық оқытудың барлық бөліктерінен қарағанда көп уақытты алады. Сонымен қатар ол – қателіктердің негізі. Адамдар көбінесе өздеріне ұнайтын функцияларды немесе «манызды» деп санайтын қасиеттерді таңдайды. Нәтижесінде модель жалған жауаптар берे бастайды.

Алгоритм (Algorithms) – компьютерде орындалатын операциялардың жиынтығын билдіретін процедура және проблеманың дұрыс шешімі шекті уақыт аралығында табылғандығын немесе шешімінің жоқтығын хабарлайтынын көпілдендіреді. Программалық жүзеге асырудың орындалу жылдамдығы, сонымен қатар нәтиженің накты шешімді қабылдау алгоритмін таңдауға тәуелді.

Машиналық оқыту принциplerі осы құраушылардың негізінде орындалады, яғни бастапқы деректер жинақталады, деректердің ерекше белгілері сұрыпталады, оқыту алгоритмі таңдалады және модель құрылады, нәтиже болжанады.

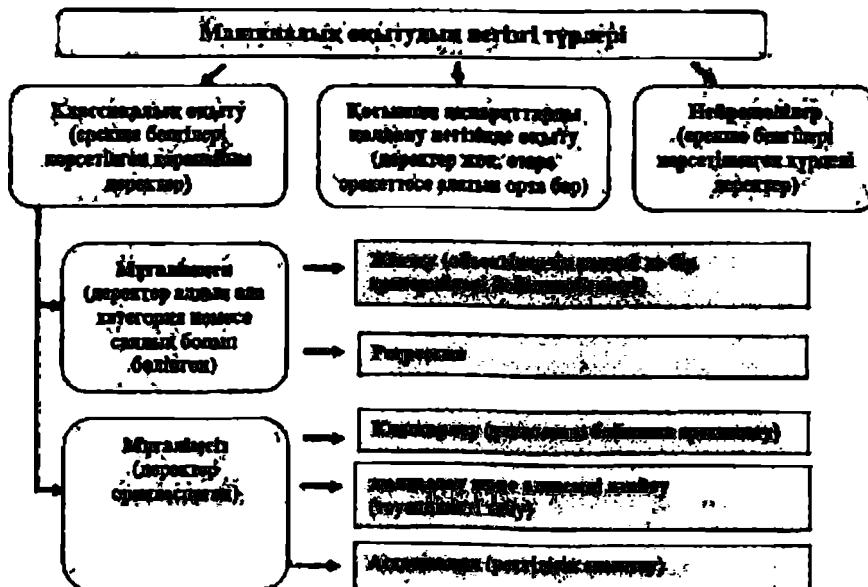


2-сурет. Машиналық оқытудың құраушылары

Ерекше белгілері (Features) – объектілердің сипаттамалары. Ерекше белгілер сандық (мысалы, тенге түріндегі кіріс немесе сантиметрмен берілген өлшем, акция бағасы т. б.) немесе категориялық (автомобиль маркасы, телефон моделі және т. б.) болып келеді. Ерекше белгілерді, ерекше айнымалыларды, атрибуттарды сұрыптау – модельді құруда қолданылатын маңызды белгілердің жиынын сұрыптау процесі.



3-сурет. Видео, аудио және мәтінді тану мысалындағы машиналық оқыту жұмысының принциптері



4-сурет. Машиналық оқытудың негізгі түрлері

Мұғаліммен машиналық оқыту – машинаның алдын-ала мұғалім көрсеткен нақты мысалдарға сүйеніп үйрену әдісі.

Мысалы, компьютердің мұғалімі бар деп есептейік. Ол мына суретте ит, ал мына суретте – жылқы деп, ит пен жылқының ерекшеліктері мен белгілерін алдын-ала көрсетіп береді. Ол машинаның жадында сакталады. Машина сол мысалдарды пайдаланып, нақты жауапты анықтайды.

Компьютер шешетін тапсырмалар екі типке бөлінеді: жіктеу және регрессия.

Жіктеу – объектілердің қандай да бір критерийлері бойынша бөлінуі. Сондықтан компьютерге белгілері мен категориялары бойынша мәліметтерді орналастыру үшін мұғалім қажет болады.

Мысалы, жіктеу пайдаланушыларды бірнеше критерийлерге сәйкес бөледі: қызығушылық, жас ерекшелігі, кесіп және т.б. Google-дің пошта жәшігіндегі хаттар мазмұны бойынша сұрыпталады: әлеуметтік желілер, жарнамалық акциялар және т.б.

Регрессия жіктелу түріне ұқсас, бірақ категория орнына санды болжайды. Мысалы, kolesa.kz сайтында автомобильдер туралы мәліметтер бар. Сайтта көліктің бағасы, шығарылған жылы, жүріп өткен қашықтығы, қозғалтқыштың көлемі және т.б. белгілері келтірілген. Компьютер таңдалған белгілері негізінде автомобиль күнына талдау жасайтын модель құрады.

Мұғалімсіз машиналық оқыту – машинаның өз бетімен белгісіз зандылықтарды табу арқылы үрленуі.

Мұғалімсіз оқытқанда, компьютер белгісіз мәліметтермен кездеседі, ол кез келген зандылықты өз бетімен табуы керек ([Интернеттегі https://www.boredpanda.com/dog-food-comparison-bagel-muffin-lookalike-teenybiscuit-karen-zack/](https://www.boredpanda.com/dog-food-comparison-bagel-muffin-lookalike-teenybiscuit-karen-zack/) сайтында жарияланған тағам түріне ұқсас жануарлардың суреттерін немесе айналадағы заттарға ұқсастығын ажыратуын мысалға келтіруге болады). Бұл жағдайда компьютер өзара ұқсас суреттерді ажыратса алмай қиналады.

Жалпылау – жоғары деңгейдегі нақты ерекше белгілері бойынша жинақтау. Мысалы, ашық қызыл түсті, үлкен, тәтті-қышқыл болатын және күздің сонына қарай пісетін алма «капорт» тобына кіреді. Егер объектінің белгілері көп болса, бұл оның моделін жасауды киыннатады, сондықтан мәліметтердің санын екі немесе үшке дейін азайтады. Мұндай модельді жазықтықта да, кеңістікте де көргө мүмкін болады.

Кластерлеу – алдын-ала белгісіз деректерді кластер бойынша жіктеу. Компьютер сипаттамасы ұқсас объектілерді іздейді және оларды кластерге біріктіреді.

Ассоциативтік ережелер дегеніміз – өзара байланысқан элементтер (деректер немесе объектілер) арасындағы қисынды қагидаларды табу механизмі.

Ассоциация (Ассоциативтік ережелер), мысалы, нарықтағы тауарды талдау кезінде, дүкен сөрелерін безендіруде, сатылымды жобалауда қажет болады. Алғаш рет мұндай тапсырмалар супермаркеттегі сауданың типтік үлгісін қалыптастыруда қолданылды. Нейрон желісінің жұмыс істеу принциптері туралы ақпаратты түсіну үшін, Google-дің Teachable Machine оқыту жобасын қолданындар <https://teachable machine.withgoogle.com/v1/>.

7.2. Нейрондық желінің ұйымдастыру

Адамның жүйке жүйесі нейрон деп аталатын жүйке жасушалар тобынан тұрады. Оның ұзындығы 1 метрге дейін жетеді немесе одан да көп болады.

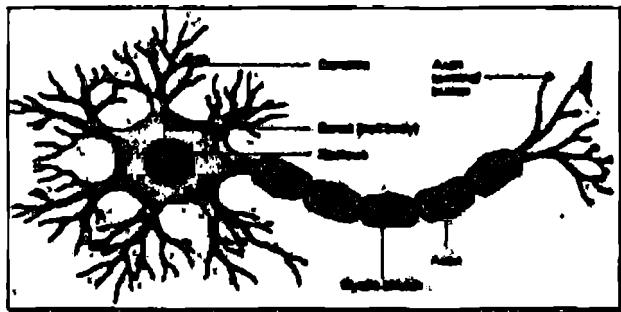
Машиналық оқытудың бастапқы кезеңіндегі тағы бір алгоритмдік тәсіл – нейрондық желілер.

Нейрон – мидаң негізгі құрылымдық-функционалдық бірлігі. Ол – миға берілетін ақпаратты өңдеу функциясымен айналысатын талшық. Олар ақпараттың қабылдануы мен бойға тарауына жауапты.

Нейрондық желілер адамның миң туралы білімге, жасанды нейрондардың арасындағы байланыстарға негізделген. Адамның ойы – мидағы нейрондар желісі жұмысының нәтижесі. Нейрондар бір-біріне электрлік импульстер арқылы химиялық сигналдар беріп отырады, мұндай ұжымдық белсенділіктен біздін сезімдеріміз бен ойларымыз туындаиды. Нейрон жасушалық денеден, ұзын және қыска талшықтардан тұрады. Ұзын талшықтар аксон, ал тарамдалған қыска талшықтар дендриттер деп аталады. Басқаша айтқанда, жекелеген нейрондар мен олардың компьютер арқылы құрылған өзара әрекеттесу желісі деп аталады.

Жасанды нейрондық желі – биологиялық нейрондық желілерді (тірі организмнің жүйке жасушалары) ұйымдастыру және жұмыс істей принципі негізінде құрылған нейрондар желісінің математикалық немесе компьютерлік моделі.

Жасанды нейрондық желінің құрылымы компьютерге келіп түскен ақпаратты талдап, сактап, артынша оны өз жадында қалпына келтіріп отыру мүмкіндігін ұсынады.



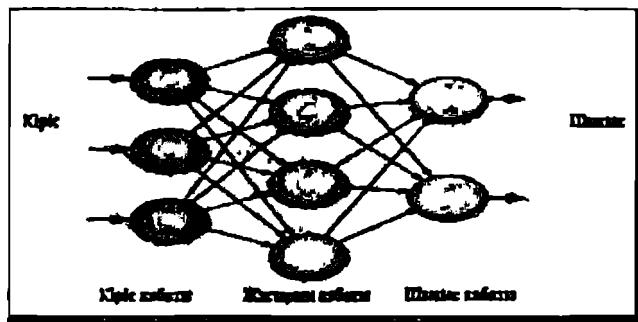
4-сурет. Нейрон құрылышы

Нейрондық желінің құраушылар және олардың параметрлері

Жасанды нейрон

Жасанды нейрон биологиялық нейронның қасиеттеріне ұксас. Жасанды нейронның кірісіне сигналдардың қандай да бір жиыны келіп түседі, олардың әрқайсысы – басқа нейронның шығысы. Әр кіріс сәйкес синаптикалық күштің салмағына кебейтіледі және барлық кебейтінділер қосылып, нейронның белсендірілу деңгейін анықтайды.

Жасанды нейрондық желі нейрондарды топтастыратын бірнеше қабаттан тұрады.



5-сурет. Жасанды нейрондық желі

Синапс – екі нейронның арасындағы байланыс. Синапстардың бір параметрі бар. Ол – салмақ. Оның көмегімен бір нейроннан келесі нейронға таралып жатқанда кіріс акпараты өзгереді. Айталық, акпаратты тарататын үш нейрон бар болсын. Оnda әр нейронға сәйкес үш салмақ болады. Үлкен салмағы бар нейрондағы акпарат келесі нейронда басым болады. Осылай салмақтардың арқасында кіріс акпараттары өндөліп, нәтиже береді.

Нейрондық желілер адам миына ұқсас аналитикалық есептеулерді қажет ететін күрделі мәселелерді шешу үшін қолданылады. Нейрондық желілер жіктеу (сұрыптау), болжау және объектілерді немесе оқигаларды тануда қолданылады.

Жіктеу – деректерді параметрлер бойынша бөлу. Мысалы, несие беру орталығында кезекте тұрган бір топ адамның қайсысына несие беру немесе бермеу қажеттігін шешу керек. Бұл жұмысты нейрондық желі жас шамасы, төлем қабілеттілігі, несие тарихы және т.с.с. акпаратты талдау арқылы жасай алады.

Болжау – деректерді негізінде зерттеу объектісінің болашақ жағдайы туралы пайымдау. Мысалы, қор нарығындағы жағдайларды талдауға негізделген акциялардың көтерілгендейдігін немесе құлдырауын болжау.

Тану – қазіргі уақытта нейрондық желілердің ең көп қолданатын әдісі. Мысалы, Google-дан немесе камералық телефондардан фотосуреттерді іздеген кезде немесе бет-бейнелердің орнын анықтағанда және оны көрсететін кезде қолданады.

Фотосуреттегі адамдарды анықтау функциясы программалық жасақтамада белсенді пайдаланылады.

Терендетілген оқыту – машиналық оқытудың бір түрі. Терендетілген оқыту кезінде компьютер адам миының нейрондық желілеріне ұқсас үлкен мөшерде жасанды нейрондық желінің қалыптастырады. Жасанды нейрожелілерді үйрету үшін бағдарлама жасаушы олардың көптеген күрделі архитектураларын ойлап шығарды. Олардың ішіндегі келешегі зор түрі – Deep

Learning (терендетілген оқыту) архитектурасы, яғни «көпқабатты нейрондық желілерді терендептіл оқыту».

Deep Learning архитектурасы машиналық оқытуды практикаға кеңінен енгізді. Жүргізушісіз жүретін автокөліктер, денсаулықты сақтау шаралары, фильмдер мен музыкаға пікір білдіру – бүгінде мұның барлығының бастамасы бар. Терендетілген оқытуға сүйеніп, жасанды интеллект өздігінен ойлап немесе шешім қабылдайды деп айтуда болмайды. Бірақ болашакта нейрондық желілер медицина мен қауіпсіздік саласында көмегін тигізіп, көптеген процесті автоматтандырып, кеңес беруші-менеджерлерді алмастырып, билет сатып алуға, зангерлік кеңес беруде және т.б. көмектесетінің айдан анық.

7.3. Нейрондық желіні жобалау

Заманауи цифрлық компьютерлердің ресурстарымен салыстырғанда, дәңгелек құрттың (нематодтың) миы 302 нейронды ғана құрайды, бұл өте аз сан! Бірақ құрт дәстүрлі компьютерлер әлі де шеше алмайтын қурделі тапсырмаларды шеше алады.

Кіріс сигналын алатын, бірақ шекті мәнін ескеріп, шығыс сигналын түрлендіретін функция белсендіру функциясы деп аталады.

Біз нейронның биологиялық мидың құрылымдық қызметтік бірлігі екенін білеміз.

Нейронның әртүрлі формалары электронды сигналдарды нейронның (дендриттің) бір жағынан екіншісіне (терминал) аксон арқылы жіберетініне әсер етпейді. Нейрон (кіріс) электр сигналын қабылдайды және басқа (шығыс) электр сигналын шығарады. Содан кейін бұл сигналдар басқа нейронға беріледі.

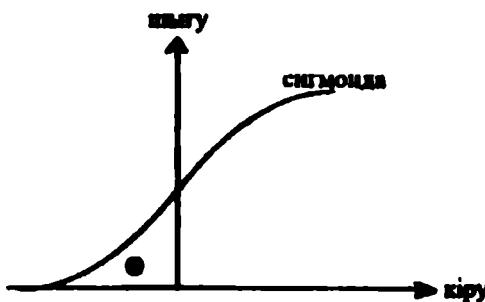
Байқау нәтижелері бойынша белгілі бір шекті мәннің болуы анықталды, оның нейрон кіріс сигналынан күтуде және осы сигналға жеткенше оның бәсендегі, содан кейін кіріс сигналын генерациялауды іске қосады.

Салыстыру үшін сұйықтық ыдыска толмайынша төгілмейді. Осылайша нейрондар пайдалы ақпаратты тасымалдайтын күшті сигналдарды шығарады.

Математикалық тұрғыдан алғанда, мұндай эффект қадамдық функцияларды беру мүмкіндігін қамтамасыз етуі мүмкін.

Сигмоида немесе сигмоидалы функция – жетілдірілген, тегістелген S тәрізді функция. Бұл – табиғи және шынайы функция, онда қадамдық функцияның өткір тіктертбұрышты шекаралары тегістелген.

Нейрондық желі құру үшін сигмоиданың қолданамызы.



6-сурет. S тәрізді сигмоида

Жасанды нейронды модельдеу

I. Шынайы биологиялық нейрондар бірнеше кірістерден тұрады. Кірістердің біріктіру, сигмоидтан шығу мәнін бақылау үшін сәйкес мәндердің қосындысын есептеу қажет.

Логистикалық функция деп аталатын сигмоида формуласы:

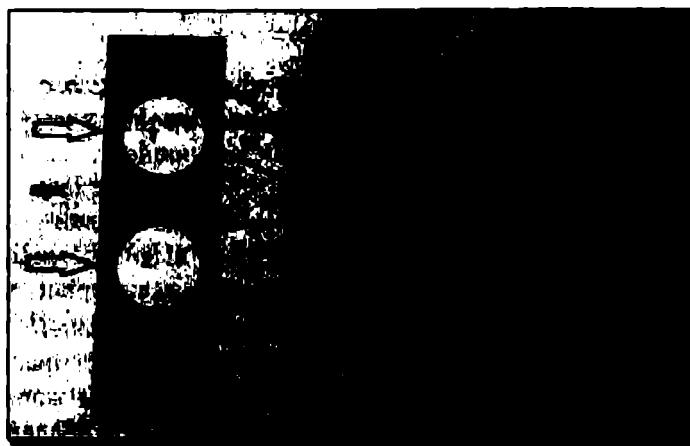
$$y = \frac{1}{1+e^{-x}}$$

мұндағы x – нейронға кіретін сигналдардың қосындысы, ал y – осы нейрондардың шығу сигналы.

II. Табиғатта байқалатын нейрондардың ретіне сәйкес әр нейронның алдында және одан кейін бірнеше нейрон орналасады, олардан ол кіріс сигналын алады және тиісінше, қоздыру жағдайында сигнал жібереді.

Жасанды модельде осындай процесті жүргізу үшін көпқабатты нейрондық күрьылымды күру әдісі пайдаланылады, онда алдынғы және кейінгі қабаттар бөлінген. Әрбір нейрон (түйін) сәйкес қабаттардағы әрбір нейронмен байланысқан. Арасы қабаттар жасырын қабаттар деп аталады.

Түйіндер арасындағы байланысты жүзеге асыратын реттегіш шамалар да (салмақ) бар. Өзара байланысқан түйіндер (нейрондар) арасында қандай да бір байланыс күшінің ($W_{i,j}$) бар екендігі көрсетілген. Мысалы, $W_{1,1}$ – бірінші қабаттың бірінші түйіні мен екінші қабаттың бірінші түйінің және т.б. арасындағы байланыстың салмақтық коэффициентін білдіреді.



7-сурет. Көпқабаттық нейрондық құрылым

Сәйкесінше тәмен салмақтық коэффициент сигналды бәсендесе, ал жоғарғы салмақтық коэффициент сигналды күшайте түседі.

Табигат зандарына сәйкес ең мықтылар ғана женіске жетеді, нейрондық желі салмақтың коэффициенттерін аныктау арқылы өзінің шығу тәртібін жақсартуды үйренеді, желідегі ішкі байланыстарды реттеуді үйрентгеннен кейін, кейбір салмақтар нөлге түседі немесе нөлге жақын болады.

Мұндай қосылымдар желіге эсер етпейді, өйткені олардың сигналдары берілмейді. Нөлдік салмаққа сигналды көбейтудің нәтижесі нөлге тең болады да, нақты байланыс үзіледі.

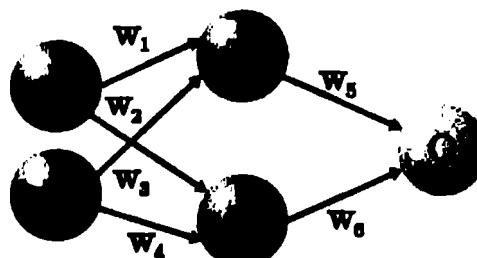
Орташа квадраттық қатені есептеу формуласы:

$$\Delta = \frac{(O_1 - A_1)^2 + (O_2 - A_2)^2 + \dots + (O_n - A_n)^2}{n}$$

Орташа квадраттық ауытқуды есептеу формуласы:

$$\Delta_r = \sqrt{\frac{(O_1 - A_1)^2 + (O_2 - A_2)^2 + \dots + (O_n - A_n)^2}{n}}$$

мұндағы, O_1, O_2, \dots, O_n – шығару (енгізу) кезінде қабылданған нақты сигнал, A_1, A_2, \dots, A_n – шығару (енгізу) кезінде қабылданатын идеал сигнал, n – мәліметтер жынтығы.



I_1, I_2 – кіріс нейрондары;
 H_1, H_2 – жасырын нейрондар;
 O_1 – шығыс нейроны.

Егер нейронның шығу сигналы 0,79 болса, орташа квадраттық қате неше пайыз?

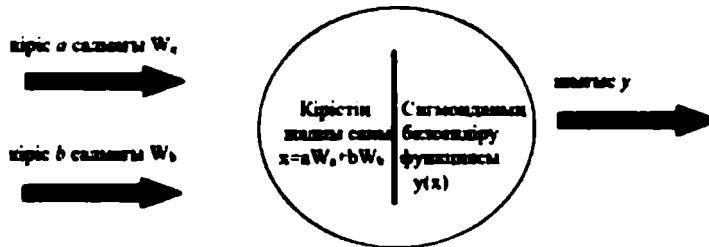
$$\Delta = \frac{(0.79 - 1)^2}{1} \cdot 100\% = 0.04 \cdot 100\% = 4\%$$

Нейрондық желі бойынша сигналдардың таралуы

Кіріс сигналдарының шығыс сигналдарына аудиосканга дейін барлық қабаттар бойынша таралтуын қалай есептеу керектігін карастырайык.

Түйіндердің ең бірінші қабаты кіріс болып саналады және оның максаты кіріс сигналдарын қабылдау. Кіріс түйіндерінде кіріс сигналдарына белсендіру (қарапайым сыйықтық функция) функциясын колданбаймыз. Екінші (жасырын) қабатпен кейбір есептерді орындаймыз. Бұл қабаттағы әрбір түйін үшін кіріс сигналының анықтау керек.

Әр түйін екі кіріс сигналдарының косындысын белсендіру функциясы арқылы бір шығыс сигналына айналдыратынын білеміз, сондыктан сигмоиданы колданамыз. Енді осы функциядағы x – байланыстардың салмақтық коэффициенттерімен тегістелген алдыңғы қабаттағы байланысқан түйіндердің өнделмелеген шығыс сигналдарынан қалыптасатын комбинацияны карастырайык.



8-сурет. Нейрондық желінің жұмыс істеу принципі

Кіріс сигналдарына $a = 1,1$ және $b = 0,6$ мәндері сәйкес келсін делік.

Кездейсок салмақтардың салмақтық коэффициенттерінен бастайык:

$$W_{1,1} = 0,8; W_{1,2} = 0,3; W_{2,1} = 0,2; W_{2,2} = 0,8$$

Тегістелген кіріс сигналы келесі өрнектердің көмегімен есептеледі:

$x = (\text{бірінші түйіннің кірісі} \cdot \text{байланыс салмағы}) + (\text{екінші түйіннің кірісі} \cdot \text{байланыс салмағы})$

$$x = (1,1 \cdot 0,8) + (0,6 \cdot 0,2) = 0,88 + 0,12 = 1.$$

Белсендіру функциясының көмегімен осы түйін үшін шығыс сигналын есептейік, мұндағы $e = 2,71828$:

$$y = 1 / (1 + e^{-1}) = 1 / 1,3679 = 0,7310. \text{ Осылайша, } y = 0,7310.$$

Осы есептеулерді қалған түйіндер үшін қайталайық – екінші қабаттың екінші түйіндері үшін:

$$x = (1,1 \cdot 0,3) + (0,6 \cdot 0,8) = 0,33 + 0,48 = 0,81;$$

$$y = 1 / (1 + 0,4448) = 1 / 1,4448 = 0,6921.$$

Нейрондық желілерді оқыту үшін колданылатын накты мәндермен берілген мысалдарды жаттығу мәліметтері деп атайды.

Қазіргі таңда нейрондық желілерді модельдеу үшін көптеген программалық жасактамалар бар, олар **программалық-имитатор** немесе **нейропакет** деп аталады. Бұлар – кәдімгі компьютерде пайдалануыш үшін нейрокомпьютердің ортасын эмуляциялайтын программалық қабықшалар. Нейропакеттерде нейрондық желілерді оқыту алгоритмінің үш тобы жүзеге асады: **градиенттік, генетикалық, стохастикалық**.

Желі параметрлеріне сәйкес қателер функцияның дербес туындыларын есептеу негізінде градиенттік алгоритмдер (1 және 2-ретті) құрылады. **Стохастикалық** алгоритмдер кездейсок жолмен қате функциясының минимумын іздейді. **Генетикалық** алгоритмде **градиенттік** және **стохастикалық** алгоритмдердің қасиеттері бірдей болады.

Нейрондық желілер олардың байланыстарының салмақтық коэффициентін нактылау арқылы оқытылады. Ол жаттығу кезіндегі мәліметтердің дұрыс жауаптары мен накты шығыс мәнінің айырmasымен басқарылады. Бұл процесс қатемен **басқару** процесі деп аталады.

Кейбір нейропакеттерді атап өтсейік: NeuroSolutions, NeuralWorks Professional II/Plus, Process Advisor, NeuroShell 2, BrainMaker Pro, Нейросимулятор Neuro Office және т.б.

Персептрондар – жасанды нейрон желілерінің карапайым түрі. Басқаша айтқанда, бұл – қабылдау, танып-білу процесін модельдейтін құрылғы.

Жеке секіру функциясы – белсендіру функциясының карапайым түрі, онда шығыс нейрондары тек 0 немесе 1-ге тең болады.

Жеке секіру функциясының математикалық формуласы келесідей:

$$\phi(\text{net}) = \begin{cases} 0, \text{net} < b \\ 1, \text{net} \geq b \end{cases}$$

Егер өлшенген сома белгілі бір b шекті мәнінен үлкен болса, онда шығыс нейроны 1-ге тең болады.

Функцияны қалай пайдаланады? Мысалы, біз теңізге өлшенген сома 5-тен үлкен немесе тең болғанда ғана барамыз делік. Егер өлшенген сома = 2, нейронның шығысы = 0 болса, онда біз ешкайда бармаймыз. Егер өлшенген сома = 7 болса, онда нейронның шығу сигналы 1-ге тең болады, яғни теңізге барамыз. Нейрон өлшенген сомага байланысты болады, егер ол шекті мәннен үлкен болса, онда нейрон 1-ге тең шығыс сигналын береді.

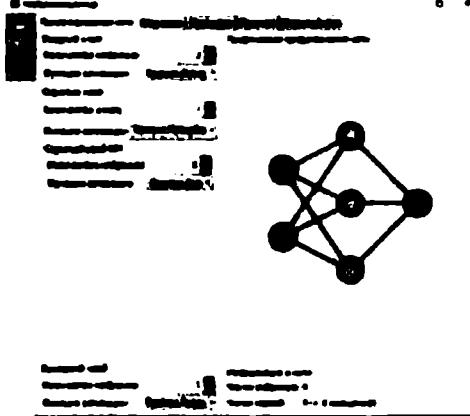
Тапсырма:

Нейрондық желіні «метрді сантиметрге айналдыру» амалын орындауға үйретейік, 1 м=100 см персептронды түрдегі нейрондық желілерді құру және қолдану мүмкіндігін беретін «Нейросимулятор 5.0» программасында орындаіык. Персепtron датчиктерден (сенсорлардан), ассоциативтік және реакциялық элементтерден тұратын мидың ақпараттарды қабылдаудының

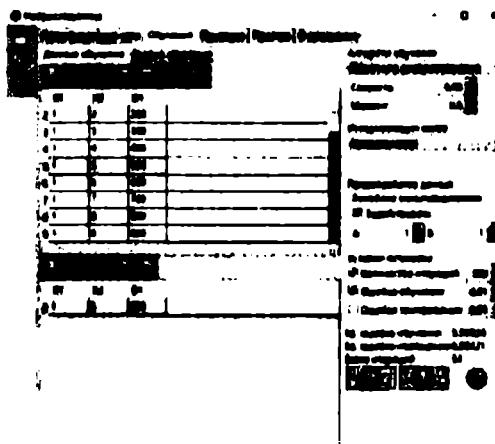
математикалық моделіне негізделген. Датчиктер арқылы қабылданатын сигналдар алдымен ассоциативтік, сонан соң реакциялық (жауап қайтаратын) элементтерге беріледі. Осылайша персепtronдар кіріс акпараттараты мен шығыс кезіндегі қажет реакция арасындағы «ассоциациялар» жиынтығын құруға мүмкіндік береді. Биологиялық түрғыдан қарағанда бұл көру акпаратының қозғалыстағы нейрондардың физиологиялық, жауап беруі сияқты түрленуге сәйкес келеді.

1. <http://www.lbai.ru/il:show;install> мекен-жайы бойынша «Нейросимулятор» жүктеп алып, іске қосындар.

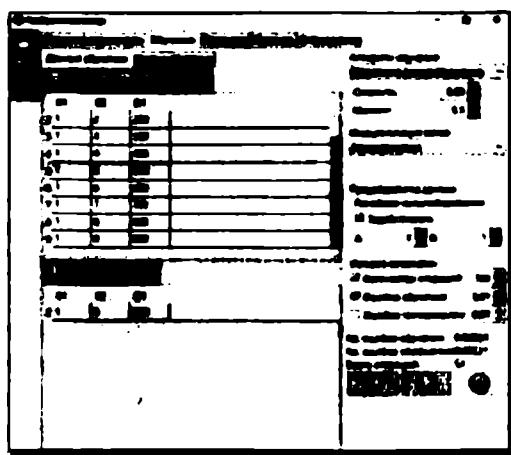
2. Нейросимулятордың ашылған терезесінің Желіні жобалау (Проектирование сети) қосымша бетінде нейрондар санын (2 кіріс, 1 шығыс, 3 жасырын) және белсендіру функцияларын белгілендер.



9-сурет. Желіні жобалау қосымша парапкшасы



Желіні оқыту батырмасын басамыз, сонда персепtronды оку процесі қатені кері тарату әдісі іске қосылады. Итерацияның санын өзгерте отырып, максималды (көк түспен көрсетілген) және орташа квадратты оқыту (қызыл түспен көрсетілген) категілерінің тәуелділік графигін көруге болады.



10-сурет. Оқыту қосымша беті

Тексеру қосымша бетіне өтіндер және D1 желісінің қажетті шығуын Y1 накты шығысымен салыстырындар. Ол үшін Оқыту жиынтығынан мәліметтерді көшіру батырмасын таңдаңдар, содан кейін Тестіленетін жиынтықтан мәліметтерді көшіру таңдаңдар.

Найросимметриялық				
	X1	X2	D1	V1
1	1	100	172.326	72.9233
2	2	200	205.954	6.9546
3	3	300	267.29	32.71
4	4	400	365.753	34.3488
5	5	500	488.121	1.8785
6	6	600	631.804	31.8049
7	7	700	733.044	33.0441
8	8	800	795.97	4.99
9	9	900	831.888	68.1884

a)

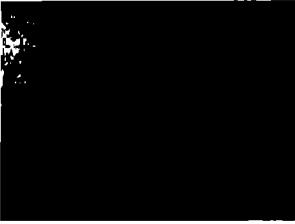
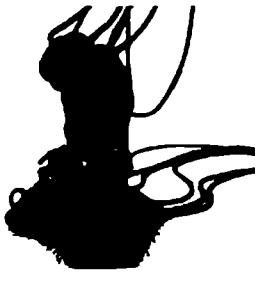
Найросимметриялық				
	X1	X2	D1	V1
1	1	3	100	100.121
2	2	4	200	1.200.121

11-сурет. Тексеру қосымша параптасы

7.4. Виртуалды және кенейтілген шынайылық

Виртуалды шындық (VR) – пайдалануышы VR гарнитурасы арқылы цифрлық ортага толығымен енү үшін шынайы ортадан шығатын толығымен иммерсивті тәжірибе. Бұл құрылғы адамдарды виртуалды ортамен байланыстырады және шынайы әлемді имитациялайтын ортада өздерінің іс-әрекеттерін басқаруға мүмкіндік береді.

Виртуалды шындықтың құрылғылары мен компоненттері:

	<p>Мамандандырылған шлемдер (көзілдіріктер). Бұл құрамға дисплейлер, линзалар жүйесі, трекингтік жүйелер кіреді. Дисплей кескіндерді шығарады, линзалар жүйесі кескіннің геометриясын түзетеді, ал трекингтік жүйелер құрылғының кеңістіктегі бағытын кадағалайды.</p>
	<p>Виртуалды шындық қолғабында (акпараттық колғап) кол мен саусактың қымылдарын бақылай алатын датчиктер колданылады.</p>
	<p>Виртуалды шындық костюмі пайдаланушының бүкіл дене қалпының өзгеруін кадағалайды, тактильді, температуралық және діріл сезімдерін және шлеммен бірге есту және көру сезімдерін жеткізеді.</p>

Кенейтілген шындық (AR) – шынайы әлемге цифрлық акпаратты қосатын технология. Кенейтілген шындық кескіндермен, мәтіндермен және басқа да виртуалды акпараттармен проекциялық дисплей, смартфондар, ақылды линзалар және ақылды көзілдіріктер сиякты құрылғылар арқылы шынайы әлемді жақсартады.

Кенейтілген шындықта өз әлемің кескін және тағы да басқа нысандар орналастырылатын ортага айналады. Біз көріп тұрған нәрсениң бәрі накты әлемде, сондықтан гарнитура тағу міндетті емес. Бұл тұжырымдаманың ең көрнекті және кең таралған мысалы: Pokémon Go ойыны. Кенейтілген шындық

шынайы әлемді цифрлық кескіндермен, графикамен және бейнелермен жақсарту үшін ноутбуктар, смартфондар, планшеттер сияқты мобильді құрылғыларда колданылады.

Виртуалды шындық пен кеңейтілген шындықтың арасындағы айырмашылықтың қарастырайық. Кеңейтілген шындық шынайы әлем мен виртуалды әлемді біріктіру арқылы жұмыс істейді. Әзірлеушілер нақты контентпен үйлесу үшін кеңейтілген шындық қосымшасында кескіндер илесе дыбыстар жасайды. Керінше виртуалды шындық компьютермен құрылған және басқарылатын өзіндік шындықты жасайды. Виртуалды шындық пен кеңейтілген шындық пайдаланушыға шынайылықты сезіндіру максаты жағынан ұқсас, дегенмен екі жүйе оны әр түрлі жасайды.

7.5. 3D панорама

Компьютерлік графика – арнайы бағдарламалар көмегімен компьютер экранында кескіндерді құру және өндеу әдістерін зерттейтін информатиканың арнайы бөлімі. Қағазға бір заттың суретін салу, мысалы, адам немесе үй, бұл аналогтық ақпарат болады: салынған сурет нақты әлемдегі затқа ұқсас бейне болады. Егер сурет компьютер экранында салынса, бұл сандық ақпараттың бір бөлігі болады. Кескінді қалыптастыру әдісіне байланысты компьютерлік графика әдетте 4 түрге бөлінеді:

- растрлық графика;
- векторлық графика;
- 3D-жобалау;
- фракталдық графика.

Үшөлшемді (3D) графика кеңістікте үш өлшемді объектілерді салудың әдістері мен тәсілдерін зерттейді. Ол кескіндерді салудың векторлық және расторлық тәсілдерін біріктіреді. Панорамалар мен виртуалды турлар – 3D графиканы колдану салаларының бірі.

Панорамалар – XVIII ғасырдың сонында пайда болған 360 градустық көрнекі бейнелер, соңғы кездері «виртуалды шындық» жүйелерінің дамуына байланысты қызығушылықты қайта арттырды.

3D-панорамалардың негізгі түрлері:

1. **Ішінәра панорамалар**, камерамен және ұялы телефондарға арналған қосымшалармен жасалған әр түрлі панорамалық режимдерден құралады және пейзаждық фотосуреттен тұрады.

2. **360° панорамалар**, олар неғұрлым жетілдірілген, себебі олар айналадағы бүкіл көріністі барлық бағытта бейнелейді.

3. **360° сфералық панорамалар**, олар айналадағы толық көріністі беріп қана коймай, сонымен қатар жоғары және төмен бүруга мүмкіндік береді, оларды жасау өте кын.

4. Цилиндрлік панорамалар, сурет цилиндр пішінің күрілады, ал көрермен цилиндрдің ортасында орналасады. Панораманың бұл түрімен кескінді интерактивті форматка түрлендіру әлдекайда қызықты, яғни кескіннің ішінде қозғалу мүмкіндігі туады.

5. Виртуалды тур – көрнекі шешім, онда бірнеше үшолшемді панорамалық фотосуреттер демонстрацияның толық жүйесіне біріктіріледі.

Туризм мен жылжымайтын мүлік саласында 3D панорамалар ақпараттық, ойын-сауық құралы ретінде кеңінен тарапған.

3D-панорамаларды қолданудың артықшылықтары:

- кез келген кеңістіктің ерекше көрінісі, мысалы: ландшафттар, конак бөлмелері, үйлер, ғимараттар, конак үйлер, кенселер;
- объектінің шектеулі уақыт аясында бейнелеуге көмектесу;
- кейде қымбат анимацияға бюджеттік балама бола алады;
- көрнекіліктің түсініктірек етеді, себебі объектінің барлық артықшылықтарын көруге болады;
- егер веб-парадашаларға кіріктірілген болса, қолданушының сайтта өткізетін уақытын көбейтеді;
- нысанды көру кезінде адамда қатысу әсері пайда болады.

Панорамалар бірнеше бөлек суреттерді бір суретке біріктіру арқылы жасалады. Суреттерді жинау арнайы штативтің басын қолданып, жеке кескіндерді жоғары дәлдікпен жазуды қажет етеді. 3D-панорамаларды қарау арнайы бағдарламалар арқылы жүзеге асырылады, мысалы, 3D Panorama Viewer.

3D-панорамаларын көруге арналған бағдарламалардың ерекшеліктері келесідей:

- кескіндерді үш өлшемде көруге болады;
- фотосуреттің ішіндегі сілтемелерді пайдаланып суреттен суретке өтуге болады;
- панорамадағы қысқаша ақпараттың көмегімен суретте қандай қала бейнеленгені жайлы білуге болады.

Сфералық панорама – панорамалық фотосурет түрлерінің бірі, онда кеңістіктің көру бұрышы 360×180 градус. Олар, ең алдымен, арнайы бағдарламалық жасақтама көмегімен компьютерде көрсетуге арналған. Сфералық панорама сфералық немесе кубтық проекцияға негізделген көптеген бөлек кадрлардан құрастырылған. Қарапайым фотосуреттерден 3D панорамалардың артықшылығы: қатысу эффектісі. Бейнеленген жерге қолданушы өзі барған сиякты әсер қалдырады.

Сфералық панорама қолданылады:

- әртүрлі елдердің мұражайларына, әдемі әрі тарихи маңызы бар жерлеріне виртуалды тур жасағанда;
- коммерциялық және жеке жылжымайтын мүлік саласында;
- ашық және жабық жерлерде пайдалануға арналған 360 градустық бейнебақылаудың арнайы жүйесінде;

- кафелерге, мейрамханаларға, қонак үйлерге көбірек назар аударту үшін жақсы жарнама жасағанда.

3D панораманы көріп тұрған адам өзін қоршаган кеңістіктің суреті бейнеленген сфераның ішінде тұрғандай сезінеді. Сонымен қатар оптикалық бағдарламалар көрінбейді. Сфералық панораманың **кубық проекциясы** келесі суретте көрсетілген:

Үшөлшемді сфералық панорамалар қалай жасалады?

Қазіргі таңда түсірілген кадрларды сфералық панорамаға өздігінен жинайтын құрылғылар (фотоаппараттар, смартфондар, планшетті компьютерлер) бар. Алайда сапалы панорамалар жасау үшін оны сандық фотоаппаратқа түсіріп, артынан арнайы компьютерлік бағдарламаларда жеке кадрларды ретімен біріктіру қажет. Олар: PTGui және Spherical Panorama бағдарламалары.

Сфералық панораманы жасау үшін камераның жан-жағындағы барлық кеңістікті қамтитындаі етіп, бірнеше сурет қатарын жасау керек. Томас Шарплесс – көптеген стереоскопиялық 3D панорамалар мен PTGui-де оларды дәл тігүге көмектесетін бағдарламалық жасақтама жасаған американлық ғалым, бағдарламалық жасақтама инженері. Бағдарлама компьютерде жергілікті жерді көру үшін де, веб-бетте орналастыру үшін де интерактивті панорамалық көру құралдарын қамтиды, сфералық панорама қалыптастыру үшін суреттерді біріктіруге көмектеседі. Тышканды шерту және сүйреу арқылы сахна бетінде жоғары, төмен және жан-жақты қарауга болады.

PTGui - дың артықшылықтары:

- кез келген бастапқы суреттерден (саны маңызды емес) сфералық панорамалар жасайды;
- jpg, tiff, png және bmp-ның бастапқы форматтарын қолдайды;
- көптеген панорамалық проекцияларды қолдайды;
- сфералық панораманы көру құралын және веб-басылым жабдығын қамтиды.

360 градус форматындағы панорамалық бейнежазбага арналған бағдарламалық жасақтаманың әлемдегі жетекші өндірушілерінің бірі **Spherical Panorama** болып табылады. Бұл – панорамалық суреттер мен виртуалды турларды жасап шығаратын бағдарламалық жасақтама. Spherical Panorama Inc. Компаниясы 1998 жылы Украинағы мамандандырылған ғылыми әзірлемелер негізінде құрылды. Компанияның филиалы Ресей мемлекетіндегі Ростов қаласында орналаскан. Олар панорамалық мамандандырылған навигациялық оптикалық бақылау жүйелері негізінде дәл оптика саласындағы ғылыми зерттеулер мен бағдарламалық жасақтаманы әзірлеуге мамандандырылған.

Spherical Panorama-ның артықшылықтары:

- 360 градус панорамалық фотосуреттер мен бейнелердің Real Estate веб-шешімін жасайды және қолданады;
- кез келген панорамалық жабдықпен сфералық және цилиндрлік форматта иммерсивті 360 градусты бейнелер мен фотосуреттер жасайды;

- интерактивті және жергілікті режимдерде карау: сфералық бейне 360 градуста иммерсивті және сфералық фотосуреттер, тегіс панорамалар, карапайым фотосуреттер, 3D фотосуреттер және 3D бейнелер;
- IP / TCP желілік байланыстары негізінде панорамалық көрінісі бар түрлі қауіпсіздік жүйелерін дамыту;
- 360 градустық контентті панорамалық сурет немесе 360 градустық бейнежазба ретінде тікелей эфирде немесе тапсырыс бойынша жасайды, басқарады және таратады;
- әр түрлі 360 градустағы панорамалардан аудио және фондық музыка арқылы түрлі виртуалды турларды құра алды;
- 360 Flash карау құралын пайдаланып Flash интернет шешімі ретінде виртуалды тур жасайды.

7.6. Виртуалды тур

Pano2VR – панорамалар мен виртуалды тур жасауға арналған бағдарлама, сонымен қатар ол көптеген басқа нәрселерге қабілетті, мысалы, анимация жасау, флэш жасау және т.б. Бұл сабакта сендер 3D тур жасаудың бірінші бөлімін жақастырасындар.

Pano2VR ашылған кезде түрлі опциялар жиынтығы ұсынылады:

«Жаңа жоба» (Новый проект) батырмасын басқан кезде пайдаланушы «Файлды ашу» (Открыть файл) диалогтық терезесіне енеді, мұнда панораманың жақетті жеріне барып, оны ашуға болады.

Панорама ашылғаннан кейін егер Pano2VR жаңартылмаса, жаңарту параметрлері бар диалогтық терезе ашылады. Оны жаңарту үшін «Жаңартуды жүктеу» (Загрузить обновление) батырамасын басу керек. Бұл жаңартуды өткізіп жіберу үшін «Жаңартуды өткізіп жіберу» (Пропустить обновление) немеме жұмысты жалғастыру үшін «Кейінірек еске түсір» (Напомнить мне позже) батырмасын басу кажет.

Егер бұл терезе ашылмаса, «Файл» мәзірінен «Жаңа» (Новый) бөліміне етіп, суретті органды кара тақтага апарып немесе құралдар тақтасындағы «Импорт» батырмасын басу арқылы жаңа жобаны бастауға болады.

Енді виртуалды экскурсияға арналған басқа суреттерді төмөнгі «Тур браузер» тақтасына апаруға болады.

Панельдер

Pano2VR-да жобаның барлық параметрлерін жасыруға, бекітуге, қайта топтастыруға және бұктеуге болады. Бұл параметрлер **Garden Gnome** бағдарламалық жасақтамасында сакталады (Pano2VR жасаушылары оларды Панельдер деп атайды). Барлығы он екі панель: свойства, параметрлерді карау, пайдаланущы деректері, тур браузер, (тур картасы), экспорт, google көшелерді карау, анимация, обзор параметров (параметрлерге шолу), әрекеттер тізімі, операциялар тарихы және карауды баптау.

Панельді көрсету үшін: жоғарғы панельден немесе терезе мәзірінен кажетті панельді таңдау керек.

Панельді жасыру үшін X пернесін басу кажет.

Белсенді аймақ нұктелері

Суреттердегі белсенді аймақ нұктелері – графика орналастырылатын нұктелер. Пайдаланушы нұктелерді басқан кезде акпараттық терезе ашылады, оны веб-сайтқа немесе келесі панорамалық кескінге апарады.

Белсенді аймақ нұктесін косу үшін:

1. Нұктені орталық терезеде қайда қою керектігін көру үшін суретті жүктеу;

2. Жоғарғы сол жактағы панель параметрлері батырмасын басу керек, нәтижесінде параметрлер жиынтығы көрсетіледі;

3. Сары нысанан белгішесін басыңыз;

4. Панорамадағы белсенді аймақ нұктесі жұмыс істейтін орынды екі рет басу қажет;

5. Нұктенің атауы мен тақырыбын, сипаттамасын қосып, сілтеме түрі ретіндегі «Связка тұра»-ны таңдал, кескін қарау құралының сол жағындағы нұктенің қасиеттері панеліндегі ашылмалы мәзірден екінші суретті таңдау керек;

6. Енді екінші суретті таңдал, жоғарыдағы белсенді аймақ нұктесі процесін қайталауға болады.

7.7. Виртуалды машиналар

Виртуалды машиналар

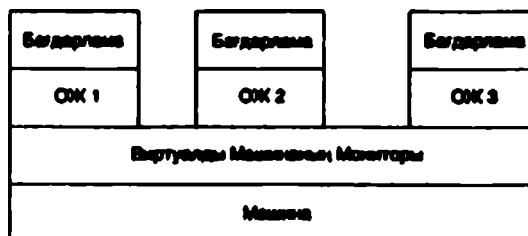
Виртуалды машина (ВМ) – компьютердің бағдарламалық көшірмесі және оның ішінде бағдарламаны іске қосуға мүмкіндік беретін арнайы бағдарламалық жасақтама. Басқаша айтқанда, бұл – компьютер ішіндегі компьютер.

Виртуалды машиналарды алғаш рет 1960 жылдары IBM компаниясы жасап, 1970 жылдары танымал болды. Ол кезде компьютерлік жүйелер үлкен және қымбат болды, сондықтан IBM виртуалды машиналар тұжырымдамасын әртүрлі пайдаланушылар арасында машина ресурстарын бөлісу тәсілі ретінде ойлап тапты.

Роберт Голдберг 60-70 жылдардағы виртуалды машиналарды зерттей отырып, олардың енгізу принциптерін корытындылады. Оның айтуынша, виртуалды машиналардың негізгі мақсаты бағдарламалық қамтамасыздандыру, операциялық жүйелерді ретке келтіру, тестілеу және диагностикалық бағдарламаларын іске қосу болды.

Виртуалды машина – физикалық жүйенің толық коргалған және оқшауланған көшірмесі. Виртуалды машинаны құру үшін Виртуалды Машинаның Мониторы (Virtual Machine Monitor) немесе гипервизор деп аталатын қабат қажет. Ол виртуалды компьютерлік жабдықтама ортасын құрып,

онымен жұмыс істейтін операциялық жүйелер (ОЖ) үшін интерфейсті қамтамасыз етеді.



Қарапайым виртуалды машина моделі

Кәдімгі компьютерлердегідей виртуалды машинаға операциялық жүйені орнатуға болады. Бұл осы операциялық жүйені колданатын бағдарламалық жасактаманы орнатуға және іске косуға мүмкіндік береді. ВМ-да жұмыс істейтін ОЖ-ны компьютерде жұмыс істейтін басқа бағдарламалардың бүтіндігін бұзбай қайта жүктеуге, өзгертуге және қайта орнатуға болады.

Сонымен қоса бір уақытта бірнеше виртуалды машинаны қолдануға болады және олардың әрқайсысы негізгі аппараттық жүйеден және басқа жүйелерден оқшаулауды қамтамасыз етеді.

ВМ-да жұмыс істейтін ОЖ «Қонақ ОЖ» деп аталады. Бастапқыда накты жабдықта жұмыс істейтін жергілікті ОС «Хост ОЖ» деп аталады.

Дегенмен накты компьютерге қарағанда аппараттық ресурстарды көбірек қажет ететін виртуалды машинаны жасау мүмкін емес. Мысалы, қолмен жұмыс істейтін калькулятордың кішкене процессорында дербес компьютер бағдарламаларын іске косу үшін виртуалды машиналарды қолдана алмаймыз.

ВМ-ның артықшылықтары:

- жаңа операциялық жүйелермен үйлеспейтін ескі бағдарламаларды іске қосу мүмкіндігі;
- бір компьютерде бірнеше операциялық жүйе жұмыс істей алады;
- ВМ-ны орнатканнан кейін көшіру және резервті көшірмені жасау өте оңай;
- зиянды бағдарламалардан корғау.

ВМ-ның кемшіліктері:

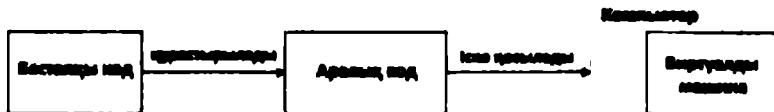
- виртуалды машинада бағдарламалар бастапқы жабдықка қарағанда баяу жұмыс істейді;
- виртуалды машинада операциялық жүйені және бағдарламарды іске қосу үшін лицензия қажет.

Операциялық жүйелерді орнатудан басқа, виртуалды машиналар белгілі бір виртуалды машинаға арналған кодты іске қосу үшін де колданылады.

Алдымен бағдарламаның бастапқы коды жазылады, содан кейін ол виртуалды машинаға арналған аралық кодқа немесе байт-кодқа жинақталады.

Бұл әдістің артықшылығы: бағдарламашыларға барлық операциялық жүйелерде жұмыс жасау үшін кодталған әр бағдарламаны қайта жазу және

компиляциялау қажет емес. Оның орнына олар виртуалды машиналардың белгілі бір түрлерінде жұмыс істеуге арналған бағдарламалар жазады. Виртуалды машинаны хост-компьютер қай операциялық жүйені қолданғанына карамастан іске қосуға болады.



Кодты іске қосу

Мысалы, Java виртуалды машинасы (JVM) өте танымал және барлық дерлік аппараттық платформаға көшірілді. Оның мақсаты: Java аралық кодын іске қосу. Бастапқы код Java-да жазылады, кейін оны компилятор аралық кодка айналдырады.

7.8. Мобильді құрылғылар

Мобильді құрылғы – алақанда ұстауға болатындей мөлшердегі сымсыз есептеуіш құрылғы. Мобильді құрылғылардың түрлері:

- ✓ дербес есептеуіш көмекші;
- ✓ смартфон;
- ✓ планшетті ДК (дербес компьютер).
- ✓ мобильді құрылғылардың негізгі сипаттамалары:
- ✓ аккумулятордың сыйымдылығы;
- ✓ тасымалданатындығы;
- ✓ қуаттылығы;
- ✓ сымсыз байланыс орнату мүмкіндігі;
- ✓ энергияны пайдалану;
- ✓ экранның тұныктығы;
- ✓ камера.

Мобильді операциялық жүйе – басқа бағдарламалардың жұмыс істеуіне мүмкіндік беретін мобильді құрылғыларға арналған бағдарламалық платформа. Қазір үш негізгі операциялық жүйе кең таралған. Олар: Apple iOS / Android / Windows.

Аккумуляторының сыйымдылығы (мАсаг.) өлшем бірлігімен өлшенеді (миллиампер/сағат). Бұл телефон бір сағат үздіксіз жұмыс істегендеге канша энергия жұмсайтынын көрсетеді. Осылайша сыйымдылығы 1000 мАс аккумулятор бір сафатта 1000 миллиампер жұмсайтынын білдіреді.

Смартфонның экран өлшемін әртүрлі жолмен көрсетуге болады. Бірінші – диагоналдегі дюйммен көрсету болса, екіншісі – ұзындығы және ені бойынша өлшемін пиксельмен көрсету.

Сенсорлы экран – пайдалануши саусактарының жанасуына немесе стилус деп аталағын күралдың әсеріне жауап беретін енгізу күрылғысы ретінде колданылатын визуалды дисплей. Сенсорлы экран түрлері:

Резистивті сенсорлық экраны

Мұнда экран бірнеше қабаттан тұрады, олардың ең жоғары жағы саусактың немесе стилустың астына бүгіліп, төмендегі қабатқа тиу үшін артқа сырғиды. Бұл телефонға экранның қай белігі басылғанын көрсететін тізекті аяктайды.

Сыйымдылық сенсорлық экраны

Мұнда экран жанасу күшіне тәуелді емес, электродтарды саусактың өткізгіштік касиетін анықтау үшін пайдаланады. Осылайша мұнда пайдалануши олардың экранды қаншалықты қатты басқанына емес, саусактың жанасуына жауап береді.

Мобильді дисплейлердің өндірудің танымал технологиялары:

- LCD немесе сұйық кристалды дисплей;
- TFT LCD;
- IPS LCD;
- OLED;
- AMOLED;
- SuperAMOLED.

Экран тұнықтығы – солдан онға қарай немесе төмennен жоғарыға қарай берілген экрандағы пиксельдер саны. Экранның анықтығы суреттің күрылғыда қаншалықты әдемі көрінетіндігіне әсер етеді.

Жоғары тұнықтық (HD)	1280x720 пиксель
Full HD	1920x1080 пиксель
Quad HD (QHD немесе 2K)	2560x1440 пиксель
4K	4096x2160 пиксель
Ultra HD	3860x2160 пиксель

Мобильді процессор – ұялы телефонның операциялық жүйесінде жұмыс істейтін қосымшалардың колдауга арналған микросхемалық жүйе.

Бұл ұялы қосымшаларды, соның ішінде жадыны басқару, графикалық өндеу және мультимедианы колдау үшін қажетті барлық жүйелік мүмкіндікті қамтамасыз ететін дербес жұмыс ортасын ұсынады.

Мобильді күрылғылардың «процессоры» көбінесе кристалдағы жүйеге (SoC) қатысты. SoC – күрылғының жұмыс істеуі үшін қажетті негізгі компоненттердің көп белігін қамтитын чип. SoC-та орналасқан компоненттер:

- орталық процессор;
- графикалық процессор;
- бейнепроцессор;
- аудиопроцессор.

Мобильді процессордың сипаттамалары:

- сәулеті;

- тект жиілігі;
- ядро саны.

Тект жиілігі – процессордың секундына қанша амал орындаітынын білдіретін мінездемесі. 1 гигагерц (ГГц) процессор секундына 1 миллиард нұсқаулық өндей алады.

Процессор ядролы – орталық процессордың маңызды элементі. Ол – нұсқаулардың бір легін орындауга қабілетті процессордың бір бөлігі. Бір ядролы процессорда бір процессор, екі ядролы процессорда екі біріктірілген процессор чиптері болады, төрт ядролы процессорда төрт біріктірілген процессор чиптері болады және т.с.с.

Графикалық процессор – кескіннің дисплейге арналған рамалық буферге шығуын жеделдетуге арналған арнайы схема.

МобиЛЬДІ смартфонда қолданылатын жады түрлері:

- жедел жады (RAM);
- тұракты жады (ROM).

Жедел жады – телефонға жаңа деректерді өте тез жазуға және өшіруге мүмкіндік беретін, жылдам жауап қайтаратын жады түрі. Ол «энергияға тәуелді жады» деп те аталады. Ұялы телефондағы жедел жады өте маңызды және үзіл телефонның жедел жады негұрлым көп болса, соғұрлым ол тез жұмыс істейді.

Телефонның операциялық жүйесін сактайтын жады микросхемасы ROM-да (тек окуға арналған жады) орналасады. iOS, Android немесе Windows сияқты операциялық жүйе жаңартылған кезде, олар тек окуға арналған деп аталса да, бұл чиптердегі акпараттарды қайтадан жазуға болады.

Негізгі мобиЛЬДІ процессор өндірушілері:

Apple

Жетекші смартфон өндіруші. 6,9 миллиард транзисторы бар алғашқы 7 нм смартфон чипі A12 Bionic чиптерін жасады.

Qualcomm

МобиЛЬДІ ақылды құрылғыларды жасайтын әлемдік көшбасшысы және болашақты болжай үшін жасанды интеллекті қолданады.

MediaTek

Тізімдегі ең жаңа өндіруші. Бұл Тайвандық чип өндірушісі смартфондарға бірегей және жана интеллектуалды мүмкіндіктерді енгізу бойынша белсенді жұмыс жасап келеді.

Samsung

Samsung микросхемалардың (SoC) мобиЛЬДІ жүйелерінің келесі буынын үсынды:

2G	деректерді жіберуге қабілетті, бірақ өте баяу жәнс сенімсіз.
3G	ең алдымен мәліметтерді жіберуге арналған. 2 Мбайтқа дейінгі жылдамдықта жұмыс істейді.
4G	кең жолақты мобиЛЬДІ байланыс технологиясының буыны.

5G	ұялы байланыстың соңғы буыны. Бұл өте жылдам, бірақ көзірігі уақытта оның қамту аймағы өте шектеулі.
Bluetooth	өте қысқа қашықтықта пайдалануға арналған сымсыз технология. Ол көбінесе сымсыз құлакқаптар үшін, сонымен катар басқа құрылғымен байланысу арқылы деректерді өте қысқа қашықтыққа жіберу үшін де қолданылады.
WiFi (Wireless Fidelity)	компьютерлер, мобилді құрылғылар және басқа құрылғылардың Интернетпен байланысуына мүмкіндік беретін сымсыз желі технологиясы.

ТАҚЫРЫПТЫҚ ТЕСТ

1. Бұл жүйе тестілеуші адамның шынайылығын қамтамасыз етіп, оның тестілеу кезінде прокторды алдауына жол бермейді.
- A) Proctoring
 B) Виртуалды кемекшілер
 C) Оқыту аралығы
 D) Third Space Learning
 E) Логистика
2. Бұл деңгейде кескінді анықтау функциясы және диалогтік интерфейсті колдану аркасында категістер санын азайту, бизнес процестерді жетілдіру, жаңадан жасау процестерін онайлату үшін ЖИ (жасанды интеллект) колданылады.
- A) Дамыту деңгейі
 B) Өндіріс деңгейі
 C) Жобалау деңгейі
 D) Логистика деңгейі
 E) Виртуалды кемекшілер
3. Proctored Test дегеніміз не?
- A) Үмытып қалған білімінізді тексеретін программа
 B) ЖИ (жасанды интеллект) колданатын компьютерлердің аркасында оқушылардың сұраныстарына дәл және жылдам жауап берे алатын жүйе
 C) Мұғалімнің іс-әрекетін кайталайтын жасанды интеллектіге негізделген арнайы компьютерлік программа
 D) Кескінді анықтау функциясы
- E) Бұл тестілеуші адамның шынайылығын қамтамасыз етіп және проктор арқылы алдауына жол бермейтін механизм
4. Виртуалды кемекшілер дегеніміз не?
- A) Үмытып қалған білімінізді тексеретін программа

- В) Мұғалімнің іс-әрекетін қайталайтын жасанды интеллектіге негізделген арналы компьютерлік программа
- С) Бұл тестілеуші адамның шынайылығын қамтамасыз етіп және проктор арқылы алдауына жол бермейтін механизм
- Д) ЖИ (жасанды интеллект) колданатын компьютерлердің арқасында окушылардың сұраныстарына дәл және жылдам жауап береді алатын жүйе
- Е) Кескінді анықтау функциясы
5. Нейрондық желіде өлшенген сома = 9, шекті мәні $b = 5$ болса, жеке секіру функциясының көмегімен шығу сигналын есепте
- А) 5
Б) 9
С) 2
Д) 1
Е) 0
6. Егер нейронның шығу сигналы 0,82 болса, орташа квадраттық қате неше пайыз?
- А) 2
Б) 4
С) 3
Д) 1
Е) 0
7. Мұғаліммен оқыту санаты:
- А) Жалпылау
Б) Ассоциация
С) Кластерлеу
Д) Регрессия
Е) Алгоритмдеу
8. Мұғалімсіз оқыту санаты:
- А) Жалпылау
Б) Жіктеу
С) Терен оқыту
Д) Регрессия
Е) Алгоритмдеу
9. Қурделі деректерді талдайтын, адам миын имитациялайтын және аппараттық және программалық түрғыдан іске асыруға қабілетті математикалық модель:
- А) Жасанды нейронды желі
Б) Жасанды интеллект
С) Синапс
Д) Биологиялық нейрон
Е) Машиналық оқыту
10. Нейрондарды бір-бірімен байланыстырады:
- А) Жасанды нейронды желі

- В) Жасанды интеллект
- С) Синапс
- Д) Биологиялық нейрон
- Е) Машиналық оқыту

11. Белгілі бір шарттарға негізделген дұрыс шешім қабылдауға мүмкіндік беретін үлгі

- А) Аксон
- Б) Дендриттер
- С) Нейрожелі
- Д) Медицинада
- Е) Интеллект

12. Нейрондар арасындағы байланыс, олардың әрқайсысы өз кіріс салмағының дәрежесіне ие

- А) Машиналық оқыту
- Б) Аксон
- С) Синапс
- Д) Нейрондық желі
- Е) Жасанды интелект

13. Бір-бірімен синапстар арқылы біріктірілген нейрондардың белгілі бір тізбегі

- А) Машиналық оқыту
- Б) Аксон
- С) Синапс
- Д) Нейрондық желі
- Е) Жасанды интелект

14. Нейрондық желілерде қатемен басқару процесі деп:

А) бір-бірімен синапстар арқылы біріктірілген нейрондардың белгілі бір тізбегі

- Б) олардың әрқайсысы өз кіріс салмағының дәрежесіне ие
- С) мәліметтердің дұрыс жауаптары мен нақты шығыс мәнінің айырмасы
- Д) есептеу кезінде қателерді анықтау

Е) Белгілі бір шарттарға негізделген дұрыс шешім қабылдау

15. Накты уақытта физикалық әлемді толықтыратын орта:

- А) Маркерлер негізінде кенейтілген шындық
- Б) Кенейтілген шындық
- С) Сәулет және дизайн
- Д) Ойын-сауық
- Е) Ғылым

16. Бірнеше кадрдан құралған қарапайым кенформатты фотосурет:

- А) Жазық панорама
- Б) 3D панорама
- С) Сфералық панорама
- Д) Биологиялық нейрон

E) Текшелі панорама

**17. Компьютер мониторында көрсетуге бейімделген қоршаган ортасың
360°-тық бейнесі:**

- A) Виртуалды тур
- B) Кеңейтілген шындық
- C) 3D панорама
- D) 3D баспа
- E) Виртуалды шындық

18. Панорамаларды өндөуге арналған программа:

- A) Hugin
- B) Google Photos
- C) Dermandar
- D) Photo Gallery
- E) Image Composite Editor

19. Накты уақытта физикалық әлемді толықтыратын орта:

- A) Маркерлер негізінде кеңейтілген шындық
- B) Кеңейтілген шындық
- C) Сәулет және дизайн
- D) Ойын-сауық
- E) Ғылым

**20. Арнайы программалық жасақтама көмегімен жасалған жасанды
компьютер:**

- A) Виртуалды тур
- B) Кеңейтілген шындық
- C) Виртуалды шындық
- D) 3D графика
- E) Виртуалды машина

VIII БӨЛІМ. ЗАТТАР ИНТЕРНЕТІ**8.1. «Заттар интернеті» дегеніміз не?**

«Заттар Интернеті» (IoT) сөзі екі негізгі бөліктен тұрады: Интернет косылудың негізі, ал заттар – объектілер, құрылғыларды білдіреді. «Заттар Интернеті» не екенін ұғыну үшін мысалдар келтірейік.

Заттар Интернетінде жүрек мониторының имплантанты бар адам, биочип ретрансляторы бар жануарлар, дөңгелекте қысымының тәмендігін көрсететін кіріктірілген датчиктері бар автомобилдер, ақылды тостерлер, иттерге арналған фитнес-карғыбаулар немесе IP-адрес тағайындалуы мүмкін және деректерді желі арқылы жіберуге болатын кез келген табиғи немесе жасанды заттар мысал бола алады.

Нақты уақыт режимінде қашықтықта жабдықтар мен жағдайларды бақылау үшін SCADA (диспетчерлік бақылау және мәліметтерді жинау) қолданылады. Одан кейін оның табиғи іріктеу IoT болды. SCADA жүйесінің соңғы буыны IoT жүйесінің бірінші буынына айналды. SCADA жүйесі аппараттық және программалық компоненттерден тұрады. Компьютерге мәліметтерді жинайтын және беретін аппараттық қамтамасыз етуде арнайы SCADA программалық қамтамасызыз ету орнатылған.

2010 жылы Қытай үкіметі IoT экожүйесі бесжылдық жоспарында стратегиялық басымдықка ие болатынын мәлімдеді.

Заттар Интернеті (IoT) – адамдар, жүйелер және басқа косымшалардан мәліметтерді жинау және олармен алмасу үшін біріктіретін миллиардтаған интеллектуалды құрылғылардан тұратын сенсорлық жүйе.

Зауыттарда косылған машиналар мен объектілер «төртінші индустріалды революцияға» алып келеді, сарапшылардың пікірінше 2020 жылға қарай жаңа кәсіпорындардың жартысынан астамы IoT-да жұмыс істейтін болады.

Заттар Интернетіне келесі технологиялар тән:

- база (платформа) – кез келген пәндік сала, әкпараттық, экономикалық, техникалық, басқарушылық және т.б. үшін маңызды жалпы технологиялар;
- тақырыптама (сегменттер) – мақсатты түрде анықталған белгілі тақырыптық салаға тән технологиялар;
- коршаған орта (инфрақұрылым) – басқа әкпараттық, техникалық, экономикалық немесе гуманитарлық жүйелермен бірігіп, тұтас немесе сегменттер бойынша IoT белсенді түрде пайдалануга мүмкіндік беретін технологиялар.

Қазақстандағы заттар интернеті

ТМД елдерінің арасында M2M-өзара әрекеттесу міндеттерін іске асыру саласында Қазақстан көш басында тұр.

Казіргі танда Қазақстанның төрт ірі қаласы – Нұр-Сұлтан, Алматы, Шымкент және Ақтөбе қалаларында Smart City pilotтық жобасы іске қосылған. Бұл жобаның мақсаты – тұргын үй-коммуналдық шаруашылық және қоғамдық

көліктегі IoT жүйесін енгізу. Үстүк және сұық суды, электр энергиясын және газды тұтынуды есепке алу үшін «Orion Система» Қазақстандық компаниясы өз күрылғысын ұсынды, оларда LoRaWAN (Long Range wide Area Networks) қашықтықтан әсер ететін радиуска ие энергия тиімділік желілері қолданылады. Есепке алудың нәтижелері пайдаланушыларға осы қызметтерді тұтыну барысында шығындарды табуға мүмкіндік береді.

Мобильді байланыс ел ішінде Заттар Интернетінің таралуында негізгі рөл аткарады.

«Қазактелеком» LoRaWAN технологиясы базасында отандық жабдықты пайдаланып, LPWAN сымсыз желісін құратын бірінші ұлттық байланыс операторы болды. Осындағы сымсыз желі негізінде GPS трекерлері, козғалыс датчиктері, «ақылды» жарықтандыру және «ақылды» бағдаршамдар жұмыс істейді. Бұлардың бәрі лицензияланған ауқымдаған жұмыс істейді.

5G (ағылш. fifth generation бесінші буын) Advanced стандарттарынан кейінгі телекоммуникация стандарттары негізінде колданыстағы 4G/IMT әрекет ететін мобильді байланыстың бесінші буыны. Жаңа буын байланысының телекоммуникациялық стандарты.

5G мобильді байланыстың тестілік сынектары Нұр-Сұлтан, Алматы және Шымкент қалаларында өткізілді. GSM стандарты – Global System of Mobile communication – мобильдік байланыстың бүкіләлемдік жүйесі. GSM стандартының мобильдік байланысының цифрлық жүйесі екінші буынның мобильді жүйесін білдіреді. Мобильді жүйелердің келесі буыннанда цифрлық технологияның таңдауы іргелі болады. GSM стандарты мобильді байланысының ұялы жүйесінде колданылатын цифрлық тәсілдердің артықшылықтары келесідей:

- анағұрлым тәмен жылдамдықты сөздерді цифрлық кодтау;
- тиімді өзгеретін жиілік жолағының кеңдігі;
- барынша жоғары кедергіге тұрақтылығы.

Huawei болжамына сәйкес 2020 жылы Қазақстанда 5G желілері базалық стансиялар пайда болады. Олар 16 мамырда XII Нұр-Сұлтан экономикалық форумында тестік режимде іске қосылған болатын.

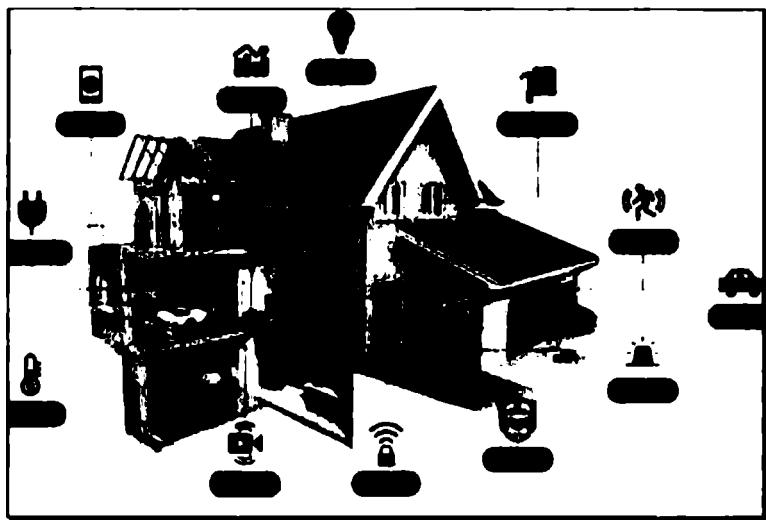
Zigbee технологиясының негізінде Smart Home платформалары іске қосылуда, бұл мобильді қосымшалар арқылы козғалыс датчиктерін, IP камерарапарын, температура мен ылғалдылық сенсорларын, тутіндеуді бақылауға мүмкіндік береді. Желілерді қамтитын IoT нарығының сегменттеріне: B2C – пәтерлер, кіреберістер мен жеke үйлер; B2B/B2G – қала көшелерінің ауқымы, әкімшілік ғимараттар, көп пәтерлі үйлер, автокөлік жолдары және өндірістік объектілер кіреді.

Price Waterhouse Coopers (PWC) бағалауы бойынша әлемде заттар Интернетінің дамуына төрт технологиялық процесс ықпал етеді: мәліметтерді беру құнын төмендету, есептеу қуатының құнын төмендету, «қосылған» күрылғылар санының жылдам өсуі, бұлттық технологияларды және Big Data жүйесін дамыту.

8.2. «Ақылды үй»

Ақылды үй – құрылғылар мен жабдықтарды кез келген жерден мобиЛЬДІ немесе басқа желілік құрылғы көмегімен Интернет байланысы арқылы автоматты түрде қашықтан басқаруға болатын үйдегі ынгайлы кондырығы. Ақылды үйдегі құрылғылар бір-бірімен Интернет арқылы қосылып, пайдаланушыға үйге қауіпсіз кіру, температура, жарыктандыру және үй кинотеатры сияқты функцияларды қашықтан басқаруға мүмкіндік береді.

Ақылды құрылғылардың мысалдары



Технологияның артықшылықтары:

- қашықтан басқару, басқа қалада бола тұра құрылғылардың жұмысын басқара алу;
- құрылғыларды пайдаланудың қарапайымдылығы, дауысты басқара алу;
- шығынды үнемдеу, бір құрылғы бірнеше жүйенің жұмысын алмастырады.

Кемшіліктері:

- тәуекелдер, әртүрлі өндірушілерге байланысты технологияларды интеграциялауда проблемалар туындауы мүмкін;
- тозған жабдықтар қауіпсіздігінде қателер туындауы мүмкін.

Ақылды үй желісінің технологияларын суретте көрсетілгендей екі негізгі түрге бөлуге болады:

Ақылды үй технологиялары

Сыншылар жүйесі	Сыншылар жүйе
Жабдық көттегі күт көзине тіреледі көсіпталған. Соңыктан олардың іске қосу немесе сандру ушын дәрістер көрмегілерге хабарлауда	Жұбада ол жеткілік жүргізет болуы мүмкін: хабарлауда және алушты. Көттегінде күршегілер баста қорасылғыштарға байланысу ушын сәнсөз технологияның колданылымы.
Мисалы: біраңғам және, оптикалық тәсілдер	Сыншылар байланыс жүйесіндегі мисал ретінде микротехнология, инфракрасный порт (IR), радиошешілдік (RF), Wi-Fi, Bluetooth жүйелерге болады.

Ақылды үйлер қалай жұмыс істейді?

Ақылды үй құрылғылары бір-бірімен байланысты және оларға нүктесмартфон, планшет, ноутбук немесе ойын консолі секілді бір орталық арқылы қол жеткізуге болады. Есіктің құлыптары, теледидарлар, термостаттар, үй мониторлары, камералар, жарықтандыру, тіпті тоназытың сияқты құрылғыларды бір ғана үйді автоматтандыру жүйесімен басқаруға болады. Ақылды үй жүйесі мобиЛЬДІ немесе басқа жөнілік құрылғыға орнатылады және пайдаланушы белгілі бір өзгерістерді қүшіне енгізу үшін ыңғайына қарай өз кестесін жасай алады.

Ақылды үй технологиясының мысалдары

Технологиялар тұрмыстың барлық салаласына (шамдар, ыдыс жуғыш машиналар және т. б.) ақылды үйдің баламаларын енгізді:

- Smart TV қосымшалар арқылы контентке, мысалы, бейнематериалға және музыкаға қол жеткізу үшін интернетке сұраныс бойынша қосылады. Кейбір ақылды теледидарлар дауысты немесе қымылдарды таниды.

- Philips Lighting Holding BV Ние сияқты ақылды жарықтандыру жүйелері белмедегі адамдардың бар-жоғын анықтап, қажет болған жағдайда жарықтандыруды реттей алады. Ақылды шамдар күндізгі жарықтың болуына байланысты өздерін реттей алады.

- Ақылды құлыптар мен гараж есіктерін ашуышылар келушілерге кіруге рұксат берे алады немесе тыйым сала алады. Ақылды құлыптар тұрғындардың жақын қалғанын анықтап, оларға есік аша алады.

- Ақылды қауіпсіздік камералары адам кетіп бара жатқанда немесе демалыста болған кезде үйді бақылауда ұстайды. Ақылды қозғалыс датчиктері тұрғындар, үй жануарлары мен қарақышлар арасындағы айырмашылықты анықтай алады және күдікті мінез-құлық анықталған кезде билікке хабарлайды.

- Үй жануарларын күтуді қосылатын коректендіргіштер көмегімен автоматтандыруға болады. Үй өсімдіктері мен ғүлзарларды қосылған таймер арқылы суаруға болады.

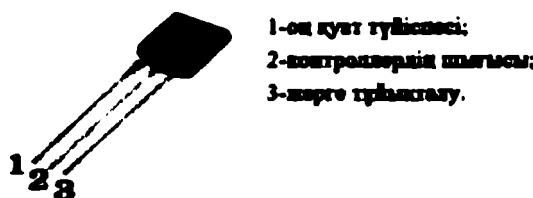
• Ас үй техникасының барлық түрі қолжетімді, соның ішінде **акылды кофекәйнатқыштар** бар, олар белгіленген уақытта автоматты түрде жана кофе қайната алады. Сонымен катар жарамдылық мерзімін қадағалайтын, сатып алу тізімдерін жасайтын, тіпті қолжетімді ингредиенттер арқылы рецептер жасайтын **акылды тоназытқыштар** да қолданыска еніп жатыр.

• **Тұрмыстық жүйенін мониторлары** кернеудің жоғарылауын аныктап, тұрмыстық техниканы өшіре алады немесе судың үзілін немесе құбырлардың катып калуын анықтай алады және жертөлөні су баспауы үшін суды өшіре алады. Олардың өздігінен білім алу дағдылары бар, үй иелерінің графигіне қарай қажетті түзетулер жасай алады.

Үй өсімдіктерін суарудың автоматтандырылған жүйесі

Белме өсімдіктеріне арналған интеллектуалды құрылғы өсімдіктерді әрдайым бакылауында ұстап, қажет болған кезде автоматты түрде суарып, күтім жасайды және судың артық құйылып кетпеуін қадағалайды. Бірнеше аптаға үйден кететін болсан, өсімдіктерінді қарауды достарыңа тапсырудың қажеті жок. Ұлғалдылық және температура датчигі арқылы топырактың ылғалдылығы, температурасы, өсімдіктерге жарықтың дұрыс түсіп түргандығы туралы нақты мәліметтерді уақытылы ала аласын. Қажет болған кезде автоматты түрде суару функциясын қосуға болады.

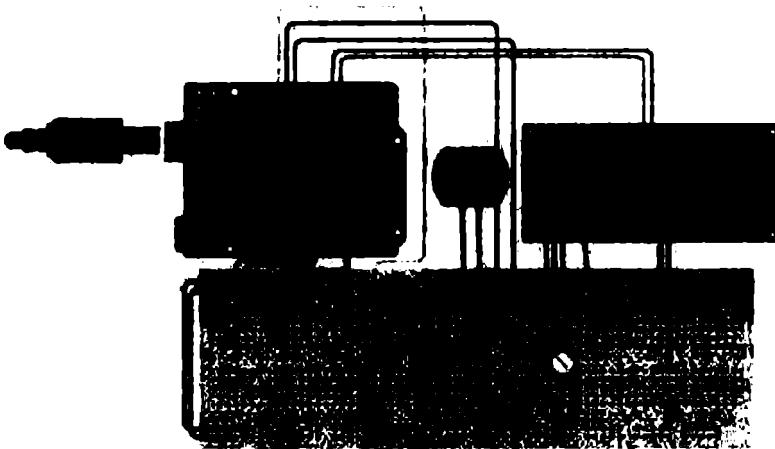
Суаруды жобалау кезінде топырактың ылғалдылық датчигін LM35 температура датчигімен ауыстыруға болады. Температура датчигінің балама үлгісі (LM35) 1-суретте көрсетілген. Оның үш түйіспесі бар. Олар он қуат түйіспесі, контроллердің шығысы және жерге түйікталу деп аталады. Тұракты ток козғалтқышы (DC motor) өсімдіктерді автоматты суаруды басқарады. NPN-транзистор тұракты ток козғалтқышын басқарады. Егер бөлмедегі жарық деңгейі темен болса, онда ол моторды қосады. Егер топырактың ылғалдылығы төмен болса, онда автоматты суаруды қосады.



Үй өсімдіктерін автоматты түрде суару үшін қажет компоненттер:

№.	Адамы	Сынады	Назареттілік
1.	U1Uno	1	Arduino Uno R3
2.	MDC motor	1	Туралы ток краганттышы
3.	NPN	1	NPN – транзистор (биполярлық)
4.	R2 R3	2	220 Ом Резистор
5.	D2	1	Жыныл түсті жарық диоды
6.	U3LCD	1	ЖК – экран (16x2)
7.	Rpot1	1	10 кОм, Потенциометр
8.	D1	1	Қызыл түсті жарық диоды
9.	R1	1	330 Ом Резистор
10.	R5	1	200 Ом Резистор
11.	U2LM36	1	Температура берігіші [LM35]

Күрылғы сыйбасы



Бұл сыйбаның бағдарламалық коды мынадай болады:

```

1 #include <LiquidCrystal.h>
2 const int LM35 = A0;
3 const int motor = 13;
4 const int LedRed = 12;
5 const int LedGreen = 11;
6 LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
7 //Жәнең жөнкүлү
8 void setup()
9 {
10   Serial.begin(9600);
11   lcd.begin(16, 2);
12   lcd.print("Automated Plant");
13   lcd.setCursor(1,1);
14   lcd.print("Watering System!");
15   pinMode(motor, OUTPUT);
16   pinMode(LedRed, OUTPUT);
17   pinMode(LedGreen, OUTPUT);
18   delay(2000);
19   lcd.clear();
20   lcd.print("Temp= ");
21   lcd.setCursor(0,1);
22   lcd.print("WaterPump= ");
23 }

```

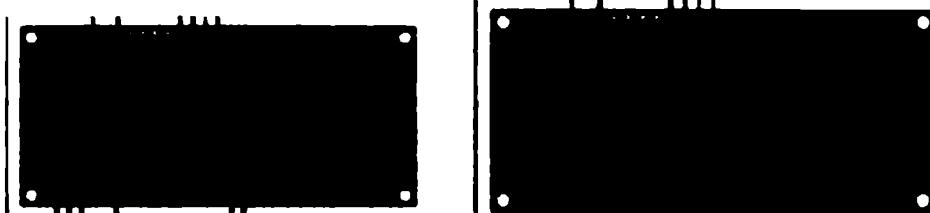
```

1 void loop()
2 {
3   // Температураны чынай
4   int value = analogRead(LM35);
5   float Temperature = value * 0.025;
6   lcd.setCursor(0, 1);
7   lcd.print(Temperature);
8   lcd.setCursor(1, 1);
9
10  if (Temperature > 25)
11  {
12    // Ағаштың тұрақтылығы
13    digitalWrite(motor, HIGH);
14    digitalWrite(LedRed, HIGH);
15    digitalWrite(LedGreen, LOW);
16    lcd.print("ON");
17
18    // Тұрақтылықтың күшін
19    digitalWrite(motor, LOW);
20    digitalWrite(LedRed, LOW);
21    digitalWrite(LedGreen, HIGH);
22    lcd.print("OFF");
23
24    delay(1000);
25  }
26 }

```

Басқару логикасы

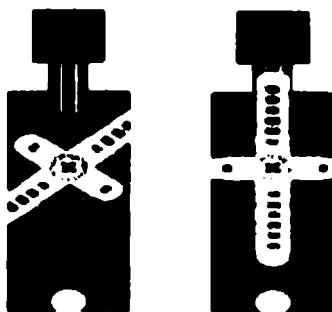
Үй өсімдіктерін суару жүйесі іске қосылған кезде LCD дисплейінде іғымдағы күй туралы хабарлама шығады:



Температура датчигі топырактың қаншалыкты ылғал екенін аныктайды. Егер топырак күргап кетсе, датчик ылғалдың деңгейі төмен екенін аныктап, өсімдікті суару үшін автоматты түрде суарғышты косады. Дисплейде *WaterPump = ON* хабарламасы шығады. Датчик өсімдік топырағына жеткілікті үй барғанын анықтаған кезде суару автоматты түрде токтайды.

Ақылды үйдің қауіпсіздігі

Біз көп жағдайда үйдегі немесе кенседегі бөлмемізге сәлкім рұқсатсыз кіре үлмайтындағы қауіпсіз етіп, маңызды аксессуарлар мен құнды заттардың қолды болмауын, жоғалмауын қадағалауымыз керек. Казіргі кезде қауіпсіздік күйесінің көптеген түрі бар. Олардың кейбірі саусак іздерін, көз торының сканерін, нұрлы кабық сканерін, тұлғаның идентификаторын, тілдік сканерді, RFID оку күралын, құпиясөз бел PIN кодты аутентификация үшін пайдаланады. Arduino пернетектасында есікті цифрлық құпиясөзбел құлыптауга болатын жүйе бар. Осылайша цифрлық құлып үй мен шкафтың есігін сенімді түрде коргайды. Әл дұрыс құпиясөз енгізілген жағдайда ғана есікті ашады. Сервомотордың есікті салай құлыптастыны суретте көрсетілген.

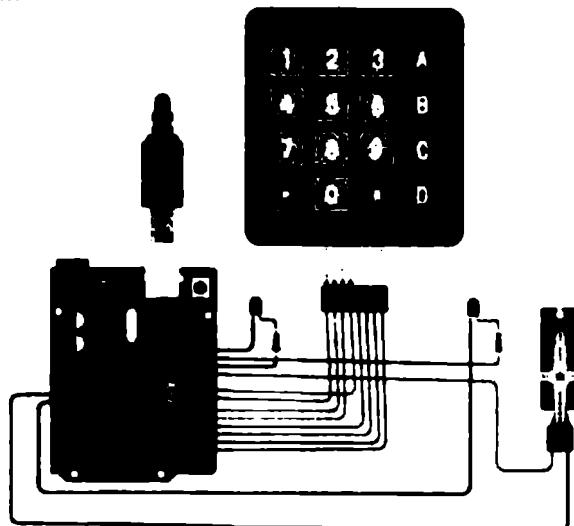


Сервомоторлар (сервопривод, серво қозғалтқыштар) накты орынға ауыса алады, сондыктан олар электроникаға сидірілген қондыргылар үшін ете қолайлы. Олардың интегралдық берілістері мен дәл басқарылатын білігі бар. Стандартты сервомоторлар білікті әртүрлі бұрыштарда, әдетте 0-ден 180 градуска дейін орналастыруға мүмкіндік береді. Үздіксіз айналмалы сервомоторлар біліктін әртүрлі жылдамдықта айналуына мүмкіндік береді.

Кауіпсіздік жүйесіне қажетті компоненттер

№	Атауы	Саны	Компоненті
1.	U1	1	Arduino Uno R3
1.	KEYPAD1	1	4 x 4 пернетақтасы
1.	SERVO1	1	Микросервомотор
1.	D1	1	Қызыл түсті жарық диоды
1.	D2	1	Жасыл түсті жарық диоды
1.	R1 R3	2	280 Ом Резистор

Қосылу сыйбасы:



Қосылу принципі:

1. сервомотордың көк сымы 11 түйіспеге қосылады;
2. сервомотордың сары сымы 5В қуат көзіне қосылады;
3. сервомотордың кара сымы GND-ге қосылады;
4. 11 және VCC (5V) түйіспелері GND микросервомоторымен байланысады;

5. 2, 3, 4, 5, 6, 7, 8, 9 түйіспелері 4x4 пернетактасына қосылады;
 6. 12 және GND түйіспелері қызыл түсті жарық диодына қосылады;
 7. 13 және GND түйіспелері жасыл түсті жарық диодына қосылады.
- Бұл сұзбаның программалық коды мынадай болады:

```

1 #include <Servo.h>
2 #include <Keypad.h>
3 Servo ServoMotor;
4 //Установки перекл
5 char* password = "4896";
6 //Конфигурация клавиш по вертикали и горизонтали
7 const byte ROWS = 4;
8 const byte COLS = 4;
9 //Настройка клавиатур
10 char keys[ROWS][COLS] = {
11     {'1','2','3','A'},
12     {'4','5','6','B'},
13     {'7','8','9','C'},
14     {'*','0','#','D'}
15 };
16 //Подключение устройства к микроконтроллеру
17 byte rowPins[ROWS] = {8,7,6,5};
18 byte colPins[COLS] = {5,4,3,2};
19 Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
20 int keypadLock = 12;
21 int screenPadLock = 13;
22 int position = 0;
23 void setup()
24 {
25     pinMode (keypadLock, OUTPUT);
26     pinMode (screenPadLock, OUTPUT);
27     ServoMotor.attach(11);
28     LockPosition(true);
29 }

```

```

30 void loop()
31 {
32     char key = keypad.getKey();
33     //Открытие двери
34     if (key == '*' || key == '#')
35     {
36         position = 0;
37         LockedPosition(true);
38     }
39     if (key == password[position])
40     {
41         position++;
42     }
43     //Закрытие двери
44     if (position == 5)
45     {
46         LockedPosition(false);
47     }
48     delay(100);
49 }
50 void LockedPosition(int locked)
51 {
52     //Настройки светодиода при НЕ правильном наборе пароля
53     if (locked)
54     {
55         digitalWrite(RedpinLock, HIGH);
56         digitalWrite(GreenpinUnLock, LOW);
57         ServoMotor.write(90);
58     }
59     //Настройки светодиода при правильном наборе пароля
60     else
61     {
62         digitalWrite(RedpinLock, LOW);
63         digitalWrite(GreenpinUnLock, HIGH);
64         ServoMotor.write(0);
65     }
66 }

```

Басқару логикасы

Arduino пернетакта модуліндегі күпиясөзді қабылдау, салыстыру, сервомотордың айналуы сияқты барлық процесті басқарады. Пернетактаны күпиясөзді енгізу үшін қолданады, ол үшін қосымша кітапхананы косу қажет:

#include <Keypad.h>

Айналу кезінде есікті ашу үшін сервоқозғалтқышты пайдалану керек. Ол кітапхана арқылы қосылады:

#include <Servo.h>

Логика пайдаланушының қауіпсіздік жүйесін іске қосудан және сервомотордың білікті 90 градуска бұрып, қызыл түсті жарық диодты косуынан, содан кейін есікті құлыштаудан турады. Жүйеде орнатылған күпиясөз: «4898». Оны енгізу үшін **char* password = "4898"** код жолын қолдану керек.

Егер күпиясөз дұрыс енгізілсе, сервоқозғалтқыш білігі 0 градусқа бұрылады, жасыл түсті жарық диоды жаңып, есік ашылады.

Үйді автоматтандырудың заманауи жүйесі

«Ақылды үй технологиясы» – белгілі бір технологиялар жынтығын қолданып, үйді автоматтандыру идеалдарының орындалуы. Бұл – жоғары технологиялық автоматты жарыктандыру, температуранны, қауіпсіздікти,

тұрмыстық техниканы бакылау және басқа да көптеген мүмкіндігі бар үй. Кодталған сигналдар үйдегі электр сымдары арқылы үйдің эр белгіндегі құрылғылар мен электронды құрылғыларды баскаруға бағдарламаланған ажыратқыштар мен розеткаларға жіберіледі. Үйді автоматтандыру жүйесіндегі қашықтықты өлшесу үшін ультрадыбыстық датчикті пайдаланады:



1 2 3

Arduino Uno-га арналған ультрадыбыстық қашықтық датчигінде 3 түйіспе бар:

- GND – жерге түйіктау;
- 5V – 5В қуат көзі;
- SIG – сандық шығыска косылатын сигнал.

PIR датчигі инфракызыл жылу сәулеленуін анықтау үшін колданылады.

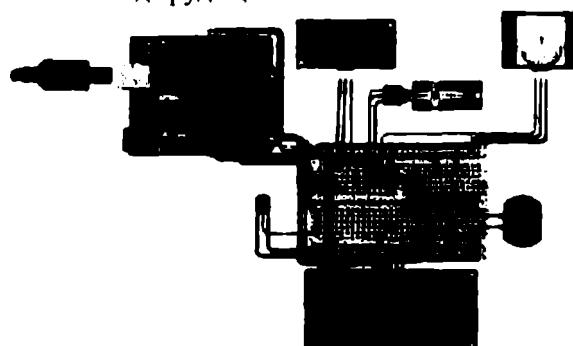
Датчикте 3 түйіспе бар:

- GND – жерге түйіктау;
- OUT – сандық шығыска косылу;
- 5V – 5В қуат көзі.

Үйді автоматтандыру компоненттері:

№	Анығы	Саны	Назарият
1.	U1	1	Arduino Uno R3
2.	U2	1	Температуралы берілші [TMP36]
3.	PING1	1	Ультрадыбыстық қашықтық берілші
4.	SERVO1	1	Микросервомотор
5.	PIR1	1	Пироэлектрілік инфрақызыл берілші
6.	U3	1	Н напірі бар электр жетекі
7.	M1	1	Туралықтаптыш
8.	D1 D5	2	Көз түсті жарық диоды
9.	D7 D2	2	Жасыл түсті жарық диоды
10.	D3 D6	2	Қызыл түсті жарық диоды
11.	R1 R2	2	220 Ом Резистор
12.	U4	1	ЖК - экран (16x2)
13.	Rpot1	1	200 кОм, Потенциометр
14.	R3	1	1 кОм Резистор

Ақылды үйді автоматтандырудың сыйбасы:



Ақылды үйдің сыйбасы

Бұл сыйбаның бағдарламалық коды мынадай болады:

```

1 #include<Servo.h>
2 #include<LiquidCrystal.h>
3 LiquidCrystal lcd(10, 9, 8, 7, 5, 4);
4 float value;
5 int tmp = A0;
6 const int pingPin = 7;
7 int servoPin = 8;
8 Servo servos;
9 void setup()
10 {
11     Serial.begin(9600);
12     servos.attach(servoPin);
13     lcd.begin(16, 2);
14     pinMode(2, INPUT);
15     pinMode(4, OUTPUT);
16     pinMode(11, OUTPUT);
17     pinMode(12, OUTPUT);
18     pinMode(13, OUTPUT);
19     pinMode(A0, INPUT);
20     pinMode(2, LOW);
21     pinMode(11, HIGH);
22 }
23 void loop()
24 {
25     long duration, inches, cm;
26     pinMode(pingPin, OUTPUT);
27     digitalWrite(pingPin, LOW);
28     delayMicroseconds(2);
29     digitalWrite(pingPin, HIGH);
30     delayMicroseconds(10);
31     digitalWrite(pingPin, LOW);
32     duration = pulseIn(pingPin, HIGH);
33     inches = microsecondsToInches(duration);
34     cm = microsecondsToCentimeters(duration);
35     servos.write(inches);
36     servos.write(cm);
37     if (inches < 10)
38     {
39         servos.write(180);
40         servos.write(180);
41         lcd.print("Door OPEN");
42     }
43     else
44     {
45         servos.write(90);
46         servos.write(90);
47         lcd.print("Door CLAMMED");
48     }
49     if (duration > 400)
50     {
51         if (tmp <= 100)
52         {
53             digitalWrite(12, HIGH);
54             digitalWrite(13, HIGH);
55             digitalWrite(4, HIGH);
56             digitalWrite(11, HIGH);
57             digitalWrite(2, HIGH);
58             delay(1000);
59             digitalWrite(12, LOW);
60             digitalWrite(13, LOW);
61             digitalWrite(4, LOW);
62             digitalWrite(11, LOW);
63             digitalWrite(2, LOW);
64         }
65     }
}

```

```

62 lcd.print ("OPP");
63 {
64 digitalWrite (4, LOW);
65 }
66 ...
67 value = analogRead (tmp) * 0.004882814;
68 value = value - 0.8)* 100.0;
69 lcd.setCursor (0, 0);
70 lcd.print ("TEMP.");
71 lcd.print (value, decimal, 1000);
72 Serial.println ("temperature");
73 Serial.print (value);
74 ...
75 if (value < 20) {
76 {
77 digitalWrite (12, HIGH);
78 digitalWrite (13, LOW);
79 }
80 ...
81 }
82 else {
83 {
84 digitalWrite (12, LOW);
85 digitalWrite (13, LOW);
86 }
87 }
88 ...
89 long microsecondsToInches (long microseconds)
90 {
91 return microseconds / 74 / 2;
92 }
93 long microsecondsToCentimeters (long microseconds)
94 {
95 return microseconds / 29 / 2;
96 }

```

Басқару логикасы

Жүйе іске қосылған кезде есік автоматты түрде жабылады, есіктің жабық екені туралы хабар және бөлменің ағымдағы температурасы LCD дисплейінде 3-суреттегідей көрінеді:



Сервоқозғалтқыш білігі бастапқыда 0 градусты көрсетеді. Егер біреу 40 см кашыктықта есікке жақындаса, онда сервомотор білікті 90 градуска бұрады және есікті "Door:OPEN" хабарламасымен 2 секундка ашады. Ол кашыктықты өлшеу үшін ультрадыбыстық датчикті және есікті ашу үшін сервомоторды колданады. Егер бөлмеде қандай да бір қозғалыс байқалса, жарық диоды автоматты түрде жанады. Ол қозғалысты анықтау үшін PIR датчигін колданады. Бөлмеге

температура анықталады, егер ол 20 градустан жоғары болса, желдеткіш автоматты түрде қосылады, керінше болған жағдайда желдеткіш өшеді. Ағымдағы күй LCD дисплейінде көрініп тұрады.

ТАҚЫРЫПТЫҚ ТЕСТ

1. Android Studio платформасы қай тілде құрылады?

- A) C++, Java, Lazarus
- B) Java, Pascal
- C) Pascal, Basic
- D) Kotlin, C++, Java
- E) Python, Basic

2. Ақылды үйдегі гаджеттер қалай басқарылатынын анықта:

- A) Интернет арқылы
- B) Bluetooth модулімен
- C) Пульт арқылы
- D) Датчик бақылау және басқару жүйесі арқылы
- E) ЖИ-дің дауыстық басқаруымен

3. Arduino тақтасы неден жасалған?

- A) Arduino IDE-ден
- B) Микропроцессорлардан
- C) Сымдардан
- D) ЖИ арқылы
- E) Bluetooth модулімен

4. Ақылды үй функциясын ата:

- A) Датчиктердің күрғак екенін көрсетеді
- B) Компьютерге USB кабель арқылы қосылады
- C) Кілемді тазалайтын роботтың болуы
- D) Есіктің кілтін ашу
- E) Поштадан хат алу

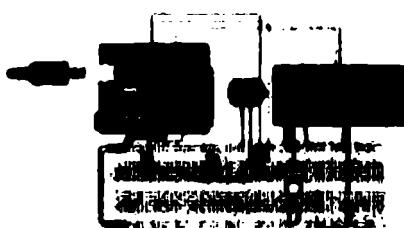
5. Ақылды үйін ішкі жүйесін анықта:

- A) Интернеттің сыртқы қорабы
- B) Жылы кофе дайындап қою
- C) Кілемді тазалайтын роботтың болуы
- D) кондиционер
- E) климатты бақылау

6. Arduino Uno датчигінің сипаттамасы:

- A) Инфрақызыл сәулелену спектрін тез сезеді
- B) 14 сандық кіріс/шығысы, 6 аналогтық кірісі бар
- C) Кедергілерді өлшеуге негізделген мәліметтер оқиды
- D) Өзі қамтитын аукымдағы адамдардың қозғалысын сезеді

- E) Адам есікке жақындағанда хабарлама береді
7. Топырактың ылғалдылығын өлшетін Arduino датчик сипаттамасы:
- Инфрақызыл сәулелену спектрін тез сезеді
 - 14 сандық кіріс/шығысы, 6 аналогтық кірісі бар
 - Кедергілерді өлшеуге негізделген мәліметтер оқиды
 - Өзі қамтитын ауқымдағы адамдардың қозғалысын сезеді
 - Адам есікке жақындағанда хабарлама береді
8. Arduino-ның өрт датчигінің сипаттамасы:
- Инфрақызыл сәулелену спектрін тез сезеді
 - 14 сандық кіріс/шығысы, 6 аналогтық кірісі бар
 - Кедергілерді өлшеуге негізделген мәліметтер оқиды
 - Өзі қамтитын ауқымдағы адамдардың қозғалысын сезеді
 - Адам есікке жақындағанда хабарлама береді
9. Arduino-ның PIR датчигінің сипаттамасы:
- Инфрақызыл сәулелену спектрін тез сезеді
 - 14 сандық кіріс/шығысы, 6 аналогтық кірісі бар
 - Кедергілерді өлшеуге негізделген мәліметтер оқиды
 - Өзі қамтитын ауқымдағы адамдардың қозғалысын сезеді
 - Адам есікке жақындағанда хабарлама береді
10. Arduino-ның су датчигінің сипаттамасы:
- Инфрақызыл сәулелену спектрін тез сезеді
 - Өткізгіштігін өлшеу арқылы датчиктің құрғақ, дымқылдығын көрсетеді
 - Кедергілерді өлшеуге негізделген мәліметтер оқиды
 - Өзі қамтитын ауқымдағы адамдардың қозғалысын сезеді
 - Адам есікке жақындағанда хабарлама береді
11. Arduino-ның газ датчигінің сипаттамасы:
- Инфрақызыл сәулелену спектрін тез сезеді
 - Өткізгіштігін өлшеу арқылы датчиктің құрғақ, дымқылдығын көрсетеді
 - Кедергілерді өлшеуге негізделген мәліметтер оқиды
 - Өзі қамтитын ауқымдағы адамдардың қозғалысын сезеді
 - Тұтінді анықтайтын балама шығыс датчигі
12. Үй өсімдіктерін суару жүйесін автоматтандыру үшін колданылған компонентті анықта.



- A) Өрт датчигі
- B) Су датчигі
- C) Фоторезистор

D) Пернетакта

E) Жасыл түсті жарық диоды

13. Блокчейн технологиясын қолданбайтын сала:

A) Оффлайн дауыс беру;

B) заттар Интернеті (IoT);

C) смарт-келісімшарттар;

D) авторлық құқықты қорғау;

E) жеке басты идентификациялауды басқару.

14. Блокчейн транзакциясы блогындағы негізгі элементті анықта

A) Тізбек

B) Аныктама енгізу

C) Хеш

D) Электронды окулық

E) Онлайн оку

15. Берілген тұжымдама, мобилді құрылғылардың қай терминіне сәйкес келеді?

• Бұл – ұялы телефонның операциялық жүйесінде жұмыс істейтін косымшаларды қолдауға арналған микросхемалық жүйе.

• Бұл жадыны басқаруга, графикалық өндеуге жүзеге асыруға және мульимедианы басқаруга мүмкіндік беретін мобилді косымшаларды қолдауға қажетті барлық мүмкіндікті қамтамасыз ететін дербес жұмыс ортасын ұсынады.

A) Мобилді процессор

B) Бейнекарта

C) Жады

D) Сенсорлы экран

E) Дербес компьютер

16. IoT қандай мағына береді?

A) Internet of Things

B) Things of Internet

C) Internet and Things

D) Things' Internet

E) Internets' Things

17. Arduino-ның қандай компоненттерін пайдаланып, есікті цифрлық кодпен құлыштатын жүйе құруға болады?

A) LCD дисплей

B) Arduino UNO

C) Пернетакта

D) Тышқан

E) Саусақ іздері

18. Олардың интегралдық берілістері мен дәл басқарылатын білігі бар. Қандай сенсор туралы айтылған?

A) Өрт датчигі

B) Су датчигі

С) Топырактың ылғалдылық датчигі

Д) Севрометр

Е) Қозғалыс датчигі

19. Ақылды үй желісінің электр сымсыз жүйесіне жататын құрылғыны ата.

А) Оптикалық талшық

В) Коаксиалды кабель

С) Бұралған жұп

Д) Есулі қосақталған

Е) Радиожиілік (RF)

20. Ақылды үй желісінің технологияларын ата

А) Жасанды интелект

В) МИПФ технологиясы

С) Сымсыз жүйе

Д) Биометрия

Е) Кодтау

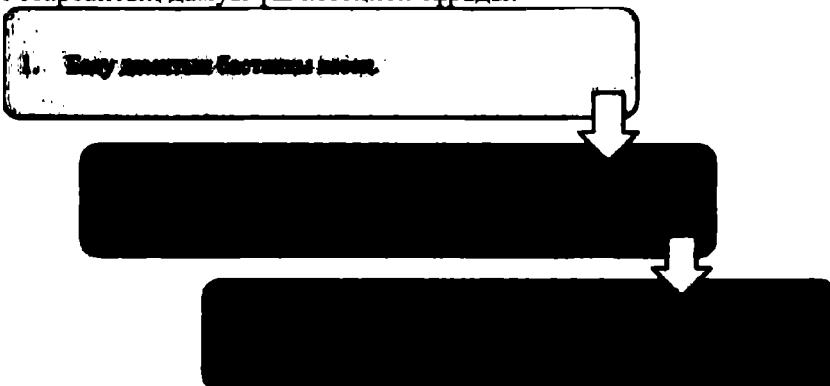
IX БӨЛІМ. IT STARTUP

9.1. Стартап қалай іске қосылады?

Startup-ты қалай іске қосады?

Startup (стартап) – жаңа инновацияларды инвестицияға қолайлы етіп дамытатын дарынды кәсіпкерлер тобы. Көбіне накты даму идеяны немесе шешүге болатын маңызды мәселені іздеуден басталады. Стартаптың мәсслеті – проблеманың бастапқы жарамдылығын және оны маңызды компанияға, тәуелсіз бизнеске дейін масштабтау алдында өнімнің сәйкестігін тексере алатын қажетті дағдылар мен қабілеттерге ие топ құру.

Сәтті стартаптың дамуы үш кезеңнен турады:



Startup-тың 6 түрі бар:

1. **Өмір сүру стиліне орай жасалған стартаптар.** Өзін-өзі жұмыспен қамтитын кәсіпкерлер, мысалы фрилансерлер, веб-дизайнерлер құрады.
2. **Шағын бизнес стартаптары.** Иесі өз отбасына тек жайлы өмірді қамтамасыз ету үшін өршіл мәссоладарды көздейді. Мысалы, шаштараздар, азық-түлік дүкендері, наубайханалар.
3. **Масштабталатын стартаптар.** Бастапқы кезден бастап әлемді өзінің бизнес идеясымен өзгерте алымын деп сенетін, сондықтан өз бизнесін өсіру үшін инвесторлардың назарын аудару үшін масштабталатын және қайталанатын бизнес моделін табуға алаңдайтын кәсіпкерлер құрган. Мысалы, Google, Uber.
4. **Сатып алынатын стартаптар.** Олар өздерінің назарын аударатын он нәтижелерге қол жеткізгеннен кейін ірі компанияларға сату ниетінен тудады. Стартаптың мұндай түрі веб-шешімдер мен мобиЛЬДІ құрылғыларды дамытатын компанияларда ете кен тараған. Мысалы, Instagram, Facebook.
5. **Ірі компаниялардың стартаптары.** Олар әртүрлі нарыктағы жаңа колданушылар үшін жаңа инновациялық өнімдер жасауға бейім.
6. **Элеуметтік стартаптар.** Қоғамды өзгертіп, әлемді жақсарткысы келетін кәсіпкерлер. Басты мәсслеті – пайда табу емес, қоғам өміріне оң үлес косу. Соның бірі – қайырымдылық қоры.

Нарыкта сәтті жұмыс істеу үшін IT стартаптың өтетін негізгі кезеңдерін қарастырайық. Бұл кезеңдер – өз жұмысын жаңадан бастайтын компаниялардың әрекеттеріне негіз.

IT стартап даму кезеңдері

1. Академик кезең (Tech-Start Stage) - инновациялық идеяның жүргізу шарттарын аныту.

2. Акелекен кезең (Business Model) - инновациялық идеяның, оның жүргізу шарттарын аныту.

3. Операцыйлық кезең (Operational Stage) - инновацияның орталық продукттың, производствоның жүргізу.

4. Рынкылық кезең (Market Stage) - инновацияның орталық продукттың, производствоның жүргізу.

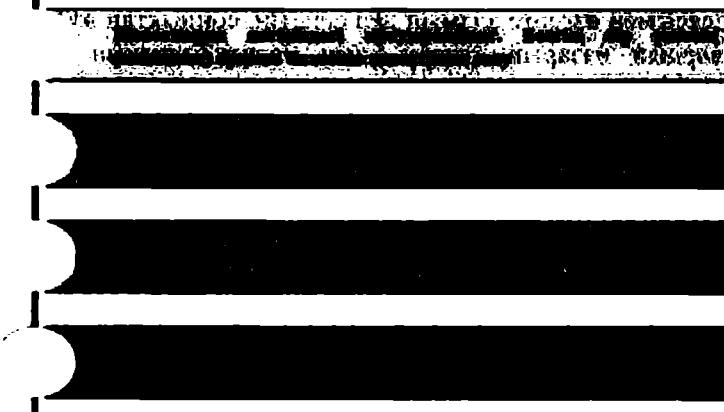
5. Стабильтік кезең (Stability Stage) - инновацияның орталық продукттың, производствоның жүргізу.

Казіргі кезде әлемде IT бизнесстің жаңа түрлері және форматтары пайда болуда. Бұл бизнесті жүйелеу және жіктеу әрекеттері үнемі табыска жету формуласын құру үшін жасалады. Электрондық қызметтер, тауарлар және қызметтің барлық түрлерінің арасында үлкен бәсекелестікке байланысты тұтынуышы үшін жаңа құндылықты іздеу процесі жүріп жатады.

Стартаптарды жүзеге асыру үшін қаржыландыру көздерін тауып, инвесторларды тарту қажет. Стартап жобаларды қаржыландыру тәсілі ретінде краудфандингті жиі пайдаланады.

Краудфандинг (crowdfunding) – ұжымдық қаржыландыру формасы. Ол потенциалды пайда немесе сыйакы алу үшін краудфандинг жобаларына қаражат салатын жеке тұлғалар немесе ұйымдар негізінде жұмыс істейді.

Краудфандингтің бірнеше түрі бар:



Краудфандингтің артықшылықтары:

- Дәстүрлі қаражат жинауға қарағанда тиімді әдіс;
- Элеуметтік белсенділікті, дәлелдерді және растауды талап ететін әдіс;
- Идеяны нақтылау үшін ұжыммен бірге ойталқы жүргізуге мүмкіндік береді;
- Бірмезгілдік маркетинг және БАҚ-да жариялау.

Кемшіліктері:

- Инвестиция салынған бизнес банкротқа ұшырауы мүмкін.
- Қайтаруға кепілдік берілмейді. Акциялар құнын қотере алмайды және пайда таба алмайды.
- Акцияларды сату қыынға соғуы мүмкін.

Барлық стартаптың негізгі қызметі – кез келген мәселеге өнімді немесе қызметті ұсыну арқылы шешім іздеу. Стартап клиенттерге, бизнестің таралуы мен каржы аспектілеріне қатысты жаңа бизнес модельдерін тұжырымдамалық тұрғыдан қарастырады және клиент тұрғысынан тұрақты немесе масштабталатынын болжау үшін бизнес модельдерді іске асырады.

9.2. Жобаны алға жылжыту

Дамыту – компания казіргі немесе әлеуетті клиенттеріне өз өнімдері немесе қызметтері жайлы хабарлау үшін пайдаланатын жарнама элементтерінің жиынтығы. Компания даму барысында барлық күш-жігерін соғы тұтынушыға және делдалдарға бағыттауы мүмкін.

Компанияның даму бағдарламасын жасау кезінде бес негізгі элемент қолданылады. Олар: жарнама, жеке сату, қоғаммен байланыс, сатуды ынталандыру және тікелей маркетинг. Жарнама – ен танымал және кең таралған даму элементі, барынша көп адамды қамтудың тиімді әдісі. Жаңа өнім немесе қызмет туралы көпшіліктің біле түсіү үшін, оның ерекшеліктерін сипаттау, пайдалану шарттарын ұсыну, бәсекелестердің ұсыныстарынан ажырату, тұтынушыларды тауарды сатып алуға итермелеу, брендтің имиджін қалыптастыру/жаксарту және т. б. үшін жарнаманы қолдануға болады. Жарнама үшін акы төленеді, сондыктan хабарламаның қандай болатынын, оның кімге, қашан және қаншалықты жиі жіберілетінін бақылай аласын. Бірақ жарнама салыстырмалы түрде кымбатқа түседі және кері байланыс алу қын.

Жарнамада қолданылатын байланыс арналары:

- **Теледидар:** дыбыстық және визуалды мүмкіндіктері бар және дыбыс, түс, кимыл үйлесімі арқылы өнім немесе қызмет түрі жайлы аппарат беруге болады.
- **Радио:** хабарламаны тек аудио арқылы жеткізуге мүмкіндік бар, бірақ максатты аудиторияға жеткізу үшін уақытты, күнді және станцияны таңдап, аудиторияны тұрақты және икемді қамтуға болады.

• **Журналдар:** көзірігі кезде ерекше қызыгуышылық тудыратын журналдардың аркасында жарнамалық хабарламаларды белгілі бір аудиторияға жеткізудін тиімді әдісі саналады.

• **Мандайша жазу:** билбордтар, техникалық қызмет көрсету станциялары, фермерлік базарлардағы мандайшалар, сауда нұктелері және супермаркеттердегі POP (сатып алу материалдары) және т.б. жатады. Сыртқы жарнама – сатып алушыларға өнімді немесе қызметті еске түсіретін өте тиімді құрал. Сыртқы жарнаманың кең таралған түрі – билбордтар. Оның қамтитын аудиториясы өте үлкен және сатып алушылар санын да көбейтеді.

• **Интернет:** тұтынушылардың көшілігі алғашқы немесе қосымша ақпарат алу үшін жүгінетін, ең жылдам өсіп келе жатқан жарнама құралы. Ол визуалды хабар алмасу мүмкіндіктерін ұсынады, сонымен қатар оның аудио/бейне мүмкіндіктері және бірегей интерактивтік мүмкіндігі де бар.

• **Жеке сату** – алға дамудың негізгі тәсілдерінің бірі. Бұл әдетте «мәмілені аяқтау» үшін сатушы мен сатып алушы арасындағы бетпе-бет байланысты білдіреді. Сатып алуға итермелебіді дамыту стратегиясының бөлігі ретінде сатушылардың рөлі – делдалдарды өнімді сатып алуға ынталандыру. Тұтынушыны тарту стратегиясының бөлігі ретінде олардың рөлі – бөлшек саудагерлерге колдау білдіру және сатудан кейінгі қызмет көрсету.

• **Жеке сатылымның негізгі артықшылықтарына** мыналар жатады: сендірудің жоғары деңгейі, жарнамалық хабарламаны баптау мүмкіндігі, жедел кері байланыс алу, ақпарат беру кезінде аудиторияны тандау мүмкіндігі.

• **Негізгі кемшіліктері** – байланыстың салыстырмалы түрде жоғары құны (сатушыларға төленетін жалакы) және сауда өкілдері жіберетін хабарламалардың өзгергіштігі.

Қоғаммен байланыс және жариялық дегеніміз – фирмалың қоғамда колайлы имидж қалыптастыру/колдау және кең аудиторияға (қызметкерлерге, бұрынғы және көзірігі клиенттерге, акционерлерге, каржы институттарына, бұқаралық ақпарат құралдарына, саясаткерлерге, т. б.) бағытталған жарнама жасау бойынша жоспарланған және тұрақты түрде өткізілетін іс-шаралар. Жарнама дегеніміз – фирмалың, өнімнің немесе қызметтің акысыз түрде ұсынылатын жалпы презентациясы.

Қажетті жарнаманы дамыту үшін колдануға болатын стратегияларға баспасөз хабарламаларын (пресс-релиз) жазу кіреді. **Баспасөз хабарламаларында** фирма, тауар/қызмет, жаңа тауарлар туралы, фирма тарихындағы маңызды оқиға, сен алған сыйлық немесе ерекше оқиға жайлы жазылады. Жарнама кітапшалары мен бейнежазбалар, сауда көрмелері, танымал адамдарға колдау көрсету және веб-сайт жасау – жарнаманы дамытудың басқа нұсқалары.

Сатуды ынталандыру дегеніміз – тауарға деген сұранысты ынталандыру үшін сонғы тұтынушыны (тарту стратегиясы) немесе делдалдарды (итермелу стратегиясы) ынталандыру тәсілі. Ол әдетте жарнамамен немесе жеке сатылыммен бірге қолданылады. Тұтынушыларға арналған жарнамалық

акцияларға бағалардың өсуі («бага женілдіктері» деп те аталады), купондар, сауда жасағандары үшін сыйлық беру, ұлгілер, байкаулар, ұтыс ойындары, ақшаны кайтару (немесе женілдіктер) және адалдықты ынталандыру шаралары жатады. Саудаға бағытталған акциялар фирмалық жарнамалық және жеке сату әрекеттерін қолдауга арналған. Сауда-саттыкты дамытудың әдеттегі құралдарына женілдіктер мен ұстемелер, бірлескен жарнама және дистрибуторлардың сату саласындағы қызметкерлерін оқытуы кіреді.

Жарнамалық акциялардың басты артықшылығы сол, ұсыныс мерзімі кезінде тауар жаксы өтіп, сатылым көлемі артады. Кемшілігі: акция аяқталғаннан кейін сатылым азайып, уақыт өте келе оның тиімділігі төмендей береді.

Тікелей маркетинг – тез дамып келе жатқан маркетингтік стратегиялардың бірі. Ол өнімді немесе қызметті ұсынатын фирма мен соңғы тұтынушы арасында тікелей карым-қатынас орнатуға, сол жердегі сатуды жүзеге асыруға және делдалдарды жоюға негізделген. Тікелей маркетинг өндіруші мен соңғы тұтынушы арасындағы тікелей байланыс арқылы фирмаларға өз нарығын тиімді түрде бағдарлауға, жаксы жауап мөлшерлемелерін алуға, қайталама сатылымдарды құруға және бәсекеге түсуге мүмкіндік береді. Тікелей маркетингтің негізгі кемшіліктері кейбір аппарат құралдарының құны, мәліметтер базасының жан-жакты және заманауи қажеттілігі, сондай-ак деректерді корғау және электронды байланыс зандарын бұзу тәуекелдері жатады. Тікелей маркетингте қолданылатын құралдар: поштаға тікелей хабарлама жіберу, кітапшалар мен үлестірме материалдар, телемаркетинг, тікелей жауап беру жарнамасы, электронды маркетинг, онлайн-маркетинг және SMS/мәтіндік маркетинг.

Өнімнің/қызметтің өмірлік циклінің кезеңдері:

- Кіріспе.
- Өсу.
- Жетілу.
- Азаю.

Потенциалды сатып алушының сатып алу туралы шешім қабылдау кезеңдері:

- Хабардар болу.
- Қызығушылық.
- Багалау.
- Сынақ және сатып алу.

Кіріспе кезеңіндегі алға жылжудың негізгі мақсаты – тұтынушылардың өнім немесе қызмет туралы хабардар болуы мен оған деген қызығушылығын арттыру. Өсу кезеңіндегі мақсат – тұтынушыны өнімді немесе қызметті сатып алуға сендіру. Жетілу кезеңіндегі мақсат – бар сатып алушыларды ұстап тұру. Жарнама тұтынушыларға өнімің немесе қызметтіңің бар екенін еске салып тұру үшін қолданылады. Өнімге деген сұраныс төмендей, сатылымнан шығуға таяған кезде, яғни азаю кезеңінде оны дамытуға, жарнамалауға аз ақша жұмсалады.

Сатып алу алдында өнім туралы хабарлар болу және оған қызығушылық таныту кезінде жарнаманың атқарар рөлі зор, себебі ол тұтынушыға сенің тауарын/қызметінің маңыздылығын тағы бір мәрте хабарлайды. Тікелей маркетинг негізінен алғашкы немесе қайталап сатып алушы ынталандыру үшін колданылады. Бұл кезеңде акция, женілдіктерді тиімді пайдалану керек. Сатып алу кезеңінде, өнім немесе қызмет бағаланып, тексеріліп жатқанда купон, мәміле, POS дисплейлері мен женілдіктер арқылы жарнамалық акцияларды ұтымды жүзеге асыруға болады. Бұл жеке сату сатып алу кезеңінде де маңызды. Жарнамалық ұсыныстарды тиімді жеткізген кезде ғана жарнама өзінің мақсатына жетіп, пайда алып келеді. Купондық және тікелей маркетингтік акцияларды өнімді/қызметті қайта сатып алуға немесе жылы қабылдауга ынталандыру үшін пайдалану керек.

9.3. IT STARTUP және жарнама

Жарнама тұтынушыға өнім туралы хабарлау және тұтынушының не алғысы келетініне назар аудару арқылы жұмыс істейді. Бұкіл әлемде жарнама корпоративтік әлемнің ажырамас бөлігіне айналды. Сондыктан компания табысының едәуір бөлігі жарнамалық бюджетке бағытталады. Жарнама сонымен катар тиімді сату үшін қажет тауардың брендін құруға қызмет етеді.

Жарнаманы қолдану мақсаты:

1. Тауарды бәсекелестердің өнімдерінен ажырату. Нарықта бір санаттағы тауарлар өте көп және олар бір-бірімен бәсекелеседі, жарнама оны берушінің өнімін бәсекелестердің өнімдерінен ажырату қызметін атқарады.

2. Өнім туралы акпарат беру. Өнімге көтісты акпарат мақсатты тұтынушыларға жеткізуі керек, ал жарнама осы функцияны орындайды.

3. Өнімді колдануға түрткі болады. Тиімді жарнама сіздін аудиторияныңда өнімге қажеттілік тудыруы мүмкін.

4. Өнімнің таралуын көңейту. Нарықта белгілі бір тауарға сұраныс артқан кезде сол өнімді сатумен айналысадын бөлшек сатушылар мен дистрибуторлар саны да көбейеді, демек өнімнің таралуы көңейеді.

Жарнамаларды функция, аймақ, мақсатты нарық, компания сұранысы, қажетті жасауд әкпарат құралдары бойынша жіктеуге болады.

Функциясы бойынша жарнама классификациясы:

- **Хабарлама-жарнама:** жарнаманың бұл түрі клиенттерге фирмадан ұйымның өнімдері, қызметтері немесе идеялары туралы хабарлама жасау үшін колданылады.

- **Сендіру жарнамасы:** жарнаманың бұл түрі потенциалды сатып алушыларды фирмадан өнімдерін немесе қызметтерін сатып алу үшін жедел әрекет етуге сендіреді немесе ынталандырады. Мысалы: «Біреуін сатып ал да, екіншісін тегін ал».

• Еске салу жарнамасы: бұл жарнама жанры бұрыннан бар клиенттерді, фирмалық өнімдерін немесе қызметтерін кем дегенде бір рет сатып алған клиенттерді орташа немесе белсенді пайдаланушылар болуға шакырады. Жарнаманың бұл түрі брендтің атауын және колданыстағы клиенттердің санасында өнімді пайдалану тәсілін сактауға көмектеседі.

Аймаққа негізделген жарнама классификациясы:

- Әлемдік жарнаманы фирма әлемдік нарықта өз орындарында жасалады. Ұлттық жарнаманы фирма ұлттық деңгейде жүзеге асырады. Бұл оның тауарлары мен қызметтеріне бүкіл ел бойынша сұранысты арттыру үшін жасалады.

- Егер өндіруші өзінің жарнамасын елдің бір аймағында қолданумен шектесе, оның жарнамалық шарасы аймақтық жарнама деп аталады.

- Жергілікті жарнама тек бір аудан немесе қала үшін жасалады.

Жарнама алатын адамдардың түрлеріне байланысты жарнамаларды бес кіші санатқа бөлуге болады:

- Тұтыну өнімдерінің жарнамасы соңғы тұтынушыны таңдандыру мақсатында жасалады.

- Өнеркәсіптік өнімдерге арналған жарнаманы іскерлік жарнама деп те атайды. Оны өнеркәсіптік өндіруші немесе оның дистрибуторы жасайды және өндіруші өндіретін өнеркәсіптік өнімге немесе қызметке сұранысты арттыру үшін арналған. Ол өндірістік тапсырыс берушіге бағытталған.

- Сауда жарнамасын көтерме және бөлшек саудагерлерді өз өнімдерін сатуға көндіру үшін өндіруші жасайды. Әрбір өндіруші өнімнің түріне, тарату арнасының сипатына және колжетімді ресурстарға негізделген әртүрлі тасымалдаушыларды таңдайды. Демек ол өнімді көтеріп сата алатын көтерме саудагерлерге және сатушыларға арналған.

- Кәсіби жарнаманы өндірушілер мен дистрибуторлар белгілі бір сауда немесе кәсіп саласындағы мамандарға ықпал ету үшін жасайды. Бұл мамандар осы өндірушілердің өнімдерін соңғы тұтынушыға ұсынады.

- Қаржылық жарнама: банктер, қаржы институттары және корпоративті фирмалар нарықтан қаражат жинау үшін жарнама шығарады. Олар проспектілер мен өтінімдердің бланкілерін басып шығарады және оларды әлеуетті инвесторлар оңай таба алатын жерлерге орналастырады.

Жарнама мақсатты тұтынушының жедел жауабын алуы немесе олардың ойында қолайлы имидж құруы мүмкін. Екі жағдайда да мақсат әртүрлі. Сонымен осы классификацияға сәйкес жарнаманың үш түрі бар:

- Тікелей жарнама клиенттерден жедел жауап алу үшін жасалады. Мысалы: маусымдық сату, журнал сатып алу купондары.

- Жанама жарнама: жарнаманың бұл түрі оқырманның немесе көрерменнін санасына жағымды эсер ету мақсатында жүзеге асырылады. Жарнаманы алып, ол тауар сатып алуға асыкрайды, бірақ оның санасында қолайлы бренд имидж қалыптастырады.

• Суррогатты жарнама: бұл – жарнаманың жана санаты. Жарнамалық күшжігердің бұл түрінде маркетолог бір тауарға жұмсап, оның негізінде басқа бір өнімді алға жылжытады. Мысалы: газдалған сусынды жарнамалau. Фирма табиғи шырындарды сатады, бірақ газдалған сусынды колдануды жарнамалайды.

Бұқаралық ақпарат құралдарына негізделген жарнаманың кең классификациясы:

- Аудио жарнамалар радио, көпшілікке хабарлау жүйелері, рикша акциялары, төрт донғалақты көліктер және т.б. арқылы таратылады.

- Визуалды жарнама РоР дисплейлері арқылы жүзеге асырылады, мәтіндік каталогтар, флейерлер, матадан жасалған баннерлер, брошюралар, электрондық билбордтар, карапайым билбордтар, жүгіру билбордтары және т.б. колданылады.

- Аудиовизуалды материалдар кинофильмдер, фильмдер, бейнеклиптер, телевизиялық жарнамалар, кабельдік теледидар жарнамалары және т.б. арқылы жүзеге асады.

- Жазбаша жарнама хаттар, факстар, мәтіні бар парапхалар, брошюралар, мақалалар мен құжаттар, газеттер және т.б. арқылы жүзеге асырылады.

- Интернет-жарнама: бүкіләлемдік желі барлық жанрдағы өнімдер мен қызметтерді жарнамалау үшін кеңінен колданылады.

- Ауызша жарнама: ауызша құралдар конференциялар, семинарлар және топтық пікірталастар кезінде ойларды, өнімдер мен қызметтерді жарнамалау үшін колданылады.

Жарнаманы құру үш кезеңде жүзеге асырылады:

- Идеялар генерациясы;
- Копирайтинг;
- Макет.

Копирайтинг – өнімді немесе қызметті сататын және потенциалды клиенттерді іс-әрекетке көндіретін көшірмені жазу өнері (веб-беттерде, жарнамаларда, жарнамалық материалдарда және т.б.). Көптеген жолмен, бұл барлық клиенттерге жету үшін бір сатушыны жалдау сиякты. Сату бөлімі клиенттермен бір-бірден байланысады; копирайтер олардың барлығына билбордтар, журнал жарнамалары, сату хаттары, блогтағы жазбалар және т.б. арқылы бірден жетеді.

Макет дегеніміз – жарнама компоненттерінің көшірме ішіндегі логикалық орналасуы. Жарнама макеті жалпы құрылымға, жарнамалық көшірменің және иллюстрациялардың әртүрлі элементтеріне берілген позицияға жатады. Макет тақырыптардың, мәтіннің, иллюстрациялардың, маркетологияның аты, логотипі мен жарнама көшірмесіндегі бос орын көлемін орналастыруды аныктайды.

Интернеттегі жарнама түрлері:

- Элеуметтік желілердегі жарнама;
- Іздеу жүйелеріндегі жарнама;
- Электрондық пошта маркетингі;

- Мақсатты беттер;
- Мазмұнды маркетинг;
- Медиалық жарнама;
- МобиЛЬДІ жарнама.

Іздеу жүйелеріндегі жарнама – сіздің веб-беттеріңізді SERP (іздеу жүйесінің иттиханалықтары туралы беттер) деңгейіне жеткізу үшін колданылады, ол әдеттегі немесе ақылы түрде болуы керек.

Электрондық пошта маркетингі – интрузивті жарнама аланды, себебі абоненттер электронды жаңалықтар мен жарнамаларды алуға дайын; Сонымен қатар олар көз келген уақытта бас тарта алады. Электрондық пошта маркетингінің басты идеясы – жарнамалық материалдарды коммерциялық емес мазмұнмен араластыру арқылы сіздің веб-сайтыңызға жылы маркетингті тарту.

Интернеттегі жарнаманың бұл түрі – осы саладағы ежелгі әдістердің бірі және ең жауаптыларының бірі, өйткені электрондық пошта маркетологтары әрдайым жаңадан басталған интерактивті хаттар сиякты жаңалықтарды іздейді.

Әлеуметтік желілердегі жарнама – Facebook, Instagram, Twitter, Pinterest сиякты әлеуметтік желілерді жарнамалауга колданады. Компаниялар өздерінің жаңалықтары мен жарнамаларын әлеуметтік желілер арқылы мақсатты аудиториясы үшін бейімдейді.

9.4. Қазақстандағы цифрландыру

Қазір барлық адамның өмірінде цифрлық технологияның элементі кездеседі. Қунделікті және кәсіби қызметте адамдар бір-бірімен цифрлық ортада қарым-қатынас жасауға, ақпаратты орынды пайдалануға және бірлесіп жаңа идеялар мен өнімдер жасауға мәжбүр болады, сондыктan әркім цифрлық сауаттылығын дамытуы қажет.

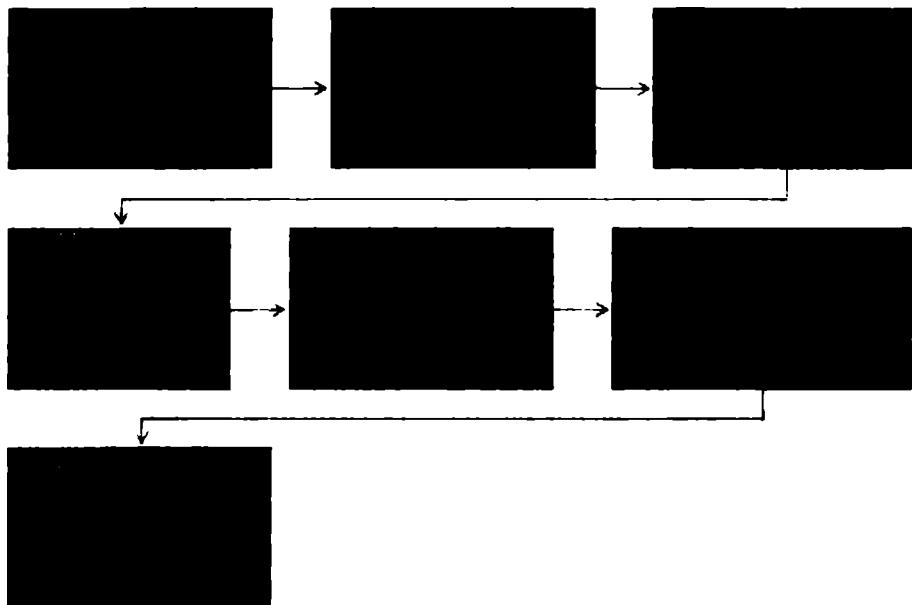
Цифрлық сауаттылық дегеніміз – цифрлық технологияларды колданып, байланыс пен ақпарат колжетімді бола түсетін қоғамда өмір сүруге, окуга және жұмыс істеуге қажетті дағдыларға ие болу.

«Цифрлық Қазақстан» (Digital Kazakhstan) мемлекеттік бағдарламасы кабылданып, қазір мемлекетте қарқынды түрде цифрландыру жұмыстары жүргіп жатыр. Бұл экономиканың жаңа моделін жасаумен, базалық салалар мен бизнесті цифрландырумен шектелмей, қоғам өміріне де жаңалық енгізді. Бағдарламаның басты бағыттары:

1. экономикадағы цифрландыру;
2. цифрлық мемлекетке көшу;
3. цифрлық Жібек жолын іске асыру;
4. адами капиталды дамытуға ықпал ету;
5. инновациялық экожүйені күру.

Мысал ретінде eGov.kz электронды үкіметтік порталын қарастыруға болады. Ол азаматтарға мемлекеттік қызметтерді жылдам алуға, тіркеуге түрү

немесе мұлікті тіркеу, лицензиялар мен сертификаттар алу қызметтерін көрсетеді. Мемлекеттік органдар, банктер мен пошта бөлімшелері де қағазбастылықтан арылып, өз жұмыстарын электрондық форматқа аудыстырып жатыр.



Қазақстан Республикасының электрондық үкіметінің даму кезеңдерінің хронологиясы көрсетілген:

Қазақстандағы цифрландыру жұмыстары

- Нұр-Сұлтанда «Сергек» атты жол ережелерінің сакталуын бақылайтын жүйе енгізіліп, жұмыс істеп түр. Егер жүйе ереже бұзылғанын анықтаса, автоматты түрде хаттама толтырылады және тәртіп бұзушының түргышықты мекен-жайына жіберіледі.

- Astana Hub – инновациялық даму платформаларын колдау, технологиялық кәсіпкерлік пен стартап мәдениетін насиҳаттау, қаржыландыруды тарту және инновацияларға деген сұранысты қалыптастыруға арналған ақпараттық технология саласындағы технопарк.

- Денсаулық сақтау министрлігі елдегі әрбір емделушіге электронды денсаулық паспортын жасайды, бұл адамның ауру тарихын интернетте сақтауға мүмкіндік береді.

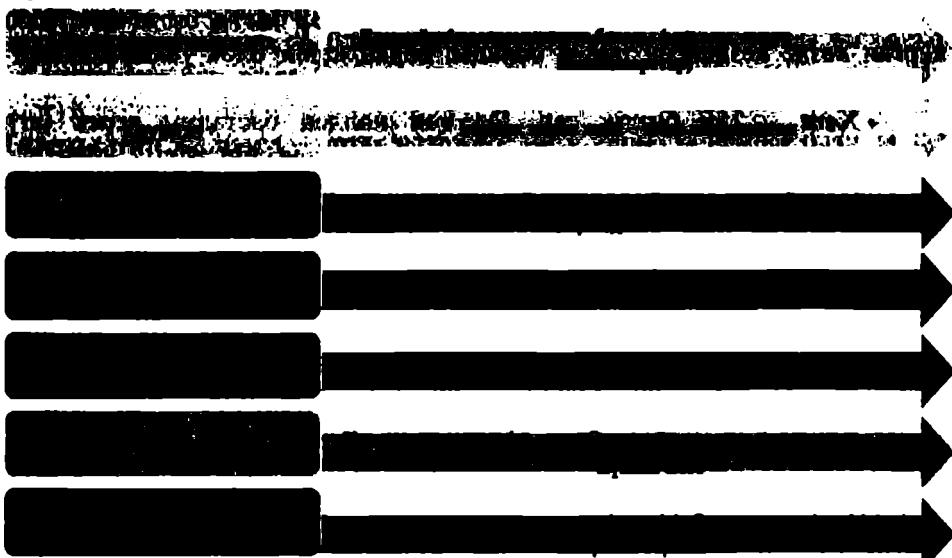
- Экономиканы цифрландыру бағдарламасы арқылы мемлекет еңбек өнімділігінің деңгейін арттыруды жоспарлап отыр. Бұл процеске жабдықта орнатылған датчик арқылы мәліметтер жинайтын, өндіріс процесінің оңтайлы болуын қамтамасыз ететін, реагенттің шығынын қадағалайтын, жабдықтың жұмыс істемейтін уақытын қысқартатын, төтенше жағдайларды тез анықтайтын және шешетін Digital Mine ақпараттық жүйесі көмектеседі.

Қазіргі уақытта электрондық үкімет – мемлекеттік қызметтерді алуға мүмкіндік беретін негізгі құрал. Оның пайдалануышылары 10 миллионнан асады. Оны дамыту үшін Ұлыбритания, Эстония, Сингапур, АҚШ және басқа да технологиясы дамыған елдердің озық шешімдері белсенді турде енгізіліп жатыр. 2020 жылы Қазақстан «электронды үкіметті» дамыту индексі бойынша БҰҰ-ның әлемдік рейтингінде 29-орынға ие болған.

9.5. Blockchain (блокчейн) технологиясы

Blockchain (блокчейн) – өзгерту, бұзу немесе алдау мүмкін емес ақпараттың жазба жүйесі. Бірнеше қатысуыш басқаратын орталықтандырылмаған деректер базасы таратылған кітаптар технологиясы (DLT) деп аталады.

Таратылған кітап технологиясының қасиеттері (DLT):



Блокчейн – транзакциялары хеш деп аталатын, өзгермейтін криптографиялық колтанбамен жазылатын DLT типі. Бұл егер сол тізбектегі бір блок өзгерілсе, оның өзгергені бірден байқалады дегенді білдіреді. Егер хакерлер блокчейн жүйесіне зиян келтіргісі келсе, тізбектің барлық таратылған нұсқасындағы барлық блокты өзгертуі керек еді.

Блокчейндермен эксперименттер 1990 жылдардың басынан бастап жүргізіліп келеді. Тек 2008 жылы Satoshi Nakamoto бүркеншік атымен жұмыс істейтін бір топ адам ресми құжат жариялағаннан кейін блокчейндер кең тарады.

Блокчейн технологиясын қолдану салалары:

- онлайн дауыс беру;
- заттар Интернеті (IoT);

- смарт-келісімшарттар;
- авторлық құқықты корғау;
- жеке басты идентификациялауды басқару.

Блокчейн технологиясын криптовалютада қолдану мысалдары.

Биткойн – бірінші рет кеңінен қолданылған орталықтандырылмаған криптовалюта. «Bitcoin: бір дәрежелі электронды ақша жүйесі» техникалық құжатының түпнұсқасы 2008 жылдың 31 казанында жарық көрді. Онда Bitcoin желісінің протоколы мен үлестірілген құрылымы бар, ал сілтеме енгізу бір жылдан кейін орындалды. Бұл құжаттар Bitcoin криптовалютасының негізіне айналды. Bitcoin баламалары – Ethereum, Litecoin, Dash, Ripple.

Блокчейнде транзакциялардың сандық кітабы көбейтіліп, компьютерлік жүйелердің желісіне таратылады. Тізбектегі әрбір блок бірқатар транзакцияларды камтиды және блокчейнде жана транзакция болған сайын әр қатысуышының кітабына осы транзакция туралы жазба косылады. Әр тізбек бірнеше блоктан тұрады және әр блокта үш негізгі элемент болады:

- **Блоктагы деректер.** Nonce деп аталатын 32 биттік бүтін сан. Nonce блок құрган кезде кездейсоқ пайда болады, содан кейін блок тақырыбының хешін тудырады.

- **Хеш** – 256 биттік сан. Ол көп нөлден басталуы керек (яғни өте кішкентай). Тізбектің бірінші блогын құрган кезде, бір реттік идентификатор криптографиялық хеш тудырады. Блоктагы деректер қол койылған болып саналады. Егер олар өндірілмеген болса, идентификаторға және хешке мәнгі байланады.

- **Түйіндер.** Блокчейн технологиясындағы маңызды ұфымдардың бірі – орталықсыздандыру. Бірде-бір компьютер немесе ұйым тізбекке иелік ете алмайды. Керінше, бұл – тізбекке қосылған түйіндер арқылы таратылған кітап. Түйіндер блокчейн көшірмелерін қолдайтын және желінің жұмысын қамтамасыз ететін кез келген электрондық құрылғы бола алады. Әр түйіннің жеке блок-тізбек көшірмесі бар және желі тізбектің жаңартылуы, сенуі мен тексерілуі үшін кез келген жаңадан алынған блокты алгоритмдік макулдауы керек.

- Барлық құрделілігіне қарамастан, блокчейннің есептің орталықтырылмаған объектісі ретінде әлеуеті шексіз. Блокчейн технологиясы қосымшалардағы қолданушылардың құпиялығы мен қауіпсіздігінің жоғары болуынан бастап өңдеу ақысы мен қателіктердің аз болғанына дейін ангарады. Сонымен қатар кемшиліктері де бар.

Блокчейннің артықшылықтары:

- адамдардың қатысуынсыз өткендіктен, тексерістердің дәлдігі артты;
- үшінші тараптың тексеруін жою арқылы шығындарды азайту;
- орталықтырылмаған өзгелердің араласуын киыннатады;
- транзакциялар қауіпсіз, құпия және тиімді;
- ашық технология;
- банктік балама және үкіметі тұрақты емес немесе дамымаған елдердің азаматтары үшін жеке ақпаратты корғау әдісін ұсынады.

Кемшіліктері:

- Bitcoin-ге байланысты технологиялық шығындары бар;
- секундына аз транзакция жасайды.

Блокчейндер ашық болғандыктан, тізілімдегі барлық әрекетті онай тексеріп, қарап шығуга болады. Әрбір катысушыга олардың транзакцияларын көрсететін бірегей нөмір беріледі. Жалпыға қолжетімді ақпаратты бакылау және тепе-тендік жүйесімен үйлестіру блокчейннің тұтастығын сактауға көмектеседі, пайдаланушылар арасында сенімділікті арттырады. Блокчейндерді технология арқылы сенімділіктің масштабталуы деп санауга болады.

Блокчейн сенімді цифрлық платформа қалыптастырады. Бұл енгізілгеннен кейін ақпаратты бұрмалаудың мүмкін емес екенін білдіреді және орталықсыздандыру, хештеу (араластыру), ашық және жабық кілттерді пайдалану деп аталатын үш негізгі қағида бойынша жұмыс істейді.

1. Орталықтандырудан таратуға дейін

Блокчейн мен дәстүрлі деректер базасының басты айырмашылығы – орталықтандыру. Деректер базасындағы қорғалған барлық жазба орталықтандырылған болса да, блок-тізбектің әрбір мүшесінде барлық жазбаның қауіпсіз көшірмесі бар және барлық қолданушы деректердің шығу тегін көре алады.

Таратылған жүйелер – ортак нәтижеге жету үшін екі немесе одан да көп түйіннің бірігіп жұмыс істейтін есептеу парадигмасы және соңғы пайдаланушылар мұны бірыңғай логикалық платформа ретінде қарастыратындағы етіп модельденген.

Орталықтандырылған жүйе

Орталықтандырылған деректер базасында клиент-сервер желісінің құрылымы қолданылады. Мұнда пайдаланушы орталықтандырылған серверде сакталатын деректерді өзгерте алады. Деректер базасын баскару дереккорға кіруге рұқсат берер алдында клиенттің тіркелгі деректерін раставтын арнайы органда қалады. Бұл орган дереккорды баскаруға жауапты болғандыктан, осы органның қауіпсіздігі бұзылған жағдайда, деректер өзгертуі немесе жойылуы мүмкін.

Орталықсыздандырылған жүйе

Блоктар көптеген орталықтандырылмаған түйіннен тұрады. Әр түйін баскаруға қатысады. Барлық түйін блок-тізбектің жаңа қосымшаларын тексереді және деректер базасына жаңа деректерді енгізе алады. Блокчейнге косу үшін түйіннің көпшілігі консенсуска келуі керек. Консенсус механизмі араласуды қындарып, желінің қауіпсіздігіне кепілдік береді.

2. Хештеу (араластыру)

Хеш – цифрлық құжатқа арналған саусак ізі ретінде қызмет ететін белгілі бір ұзындықтағы қысқа код. Хеш-генератор деп аталатын бағдарлама қолданушыға кез келген мәтін жолын жүктеуге және бірегей идентификатор құруға мүмкіндік береді. Хеш-генератор арқылы мәтіннің бірдей жолы өткен сайын бірдей құжат идентификаторын береді. Рұқсат етілмеген кіруден корғау

құралы ретінде хештеудің қосқан үлесі өте манызды. Егер құжаттағы бір әріл өзгерсе, ол автоматты түрде мүлдем басқа идентификатор жасайды.

Хештеу үлгісі:

Деректердің енгізу	Қорытынды (Хеш)
Сәлем	3639efcd08abb273b1619e82e78c29a7df02c1051b1820e99fc395dcaa3326b8
Танысқаны мақуаныштым ын	b1e91f811d3e926d194b387e41ecdb8d1adfac49c0f9747ee27dfb5cbc331c23

Хештің әртүрлі вариацияларын есептеу мына формула бойынша есептеледі:

$$P=2^i$$

Мұндағы P – ықтималдық, i – хеш битінің саны. Бұл мысалда 2^{64} түрлі ықтималдыққа ие 64 биттік хеш қолданылады. Хеш-кодтың әрбір танбасы 1 битпен кодталған.

Хештер бір жакты. Бұл хеш-генераторын құжаттан хеш жасау үшін қолдануга болады, бірақ хештен көркем жасау математикалық тұрғыдан мүмкін емес дегенді білдіреді.

Блокчейндегі транзакциялардың әрбір блогы акпараттық блоктың хешін, сондай-ақ алдыңғы блокты қоса коргайды. Бұл барлық тарарапқа ешқандай транзакцияның өзгермейтініне немесе бұзылмайтынына кепілдік береді.

3. Ашық және жабық кілттер

Ашық кілт – адамды сәйкестендіруге болатын жалпыға ортақ сәйкестендіру нәмірі.

Жабық кілт – ашық кілтпен математикалық байланысы бар пароль. Кілттерді пайдалану процесі мынадай:



Ашық/жабық кілттер жұбын пайдалану кезінде пайдаланушы жеке жабық кілт деректерін бағдарламалық жасактамаға енгізу арқылы өзінің шынымен ашық кілт «кесе» екенін растай алады; бұл өз кезегінде екі кілттің шынымен де математикалық байланысты екенін тексереді. Бұл функцияны керісінше іске қосу мүмкін емес, яғни тек қана ашық кілт туралы акпарат болса, жабық кілт жасау мүмкін емес.

ТАҚЫРЫПТЫҚ ТЕСТ

1. Сәтті Start Up-тың бірінші кезеңін көрсет

A) Инвестордың қатысуымен, өнімді нарыққа шыгару. Өнім аналогтарынан артық болуы керек

B) Идеяларды және оларды жүзеге асыру әдістерін іздеу.

C) Нарықтағы өнімге сұраныс, жоспарланған нарық орнын дамыту.

D) Компанияның даму шынына жету. Инвесторлар өз үлесін ірі компанияларға сатады.

E) Бизнес жоспарының мақсаттарына жету, жана нарықтарды игеру

2. Сәтті Start Up-тың екінші кезеңін көрсет

A) Инвестордың қатысуымен, өнімді нарыққа шыгару. Өнім аналогтарынан артық болуы керек

B) Идеяларды және оларды жүзеге асыру әдістерін іздеу.

C) Бизнес жоспарының мақсаттарына жету, жана нарықтарды игеру

D) Нарықтағы өнімге сұраныс, жоспарланған нарық орнын дамыту.

E) Компанияның даму шынына жету. Инвесторлар өз үлесін ірі компанияларға сатады.

3. Сәтті Start Up-тың үшінші кезеңін көрсет

A) Инвестордың қатысуымен, өнімді нарыққа шыгару. Өнім аналогтарынан артық болуы керек

B) Идеяларды және оларды жүзеге асыру әдістерін іздеу.

C) Бизнес-жоспарының мақсаттарына жету, жана нарықтарды игеру

D) Нарықтағы өнімге сұраныс, жоспарланған нарық орнын дамыту.

E) Компанияның даму шынына жету. Инвесторлар өз үлесін ірі компанияларға сатады.

4. Сәтті Start Up-тың төртінші кезеңін көрсет

A) Инвестордың қатысуымен, өнімді нарыққа шыгару. Өнім аналогтарынан артық болуы керек

B) Идеяларды және оларды жүзеге асыру әдістерін іздеу.

C) Бизнес жоспарының мақсаттарына жету, жана нарықтарды игеру

D) Нарықтағы өнімге сұраныс, жоспарланған нарық орнын дамыту.

E) Компанияның даму шынына жету. Инвесторлар өз үлесін ірі компанияларға сатады.

5. Сәтті Start Up-тың бесінші кезеңін көрсет

A) Инвестордың қатысуымен, өнімді нарыққа шыгару. Өнім аналогтарынан артық болуы керек

B) Идеяларды және оларды жүзеге асыру әдістерін іздеу.

C) Бизнес жоспарының мақсаттарына жету, жана нарықтарды игеру

D) Нарықтағы өнімге сұраныс, жоспарланған нарық орнын дамыту.

E) Компанияның даму шынына жету. Инвесторлар өз үлесін ірі компанияларға сатады.

6. Бірінші рет кең таралған орталықтандырылмаған криптовалюта қалай аталады?

- A) Биткойн
- B) Блокчейн
- C) IoT
- D) Ethereum
- E) Краудфандинг

7. Транзакциялар хеш деп аталатын өзгермейтін криптографиялық колтаңба арқылы жазылатын DLT түрін көрсет.

- A) Биткойн
- B) Блокчейн
- C) IoT
- D) Ethereum
- E) Краудфандинг

8. Азаматтарға мемлекеттік қызметтерді жылдам алуға мүмкіндік беретін электронды үкімет порталы қалай аталады?

- A) Astanahub.com
- B) Digital Mine
- C) Sergek.kz
- D) eGov.kz
- E) zhilfond.kz

9. Стартап құру нeden басталады?

- A) Жобаны қаржыландырудан
- B) Команда атын қоюдан
- C) Идея іздеуден
- D) Өнімді дамытудан
- E) Жарнама жасаудан

10. Қазақстан экономикасын дамыту қарқынын жеделдетуге және біздің азаматтардың тұрмыс сапасын жақсартуға арналған үдеріс:

- A) Дамыту
- B) Акпараттандыру
- C) Компьютерлендіру
- D) Технологияландыру
- E) Цифрландыру

11. Краудфандинг дегеніміз:

- A) Кез келген идеяны қаржыландыру
- B) Цифрландыру
- C) Заттар интернеті
- D) Ақылды үй
- E) Жарнама жасау

12. Стартап табысының факторларына жатпайды:

- A) Идея;
- B) Команда;

- C) Уақыт;
- D) Каржыландыру;
- E) Жылдамдық;

13. Блокчейн жұмыс істейтін технологияны ата.

- A) Орталықтандыру
- B) Хештеу
- C) Биткойндар
- D) Алдыңғы блоктың хеші
- E) Цифрландыру

14. Адамды сәйкестендіруге болатын, ашық сәйкестендіру нөмірі қалай аталауды?

- A) Адрес
- B) Хеш
- C) Биткойн
- D) Ашық кілт
- E) Жабық кілт

15. Блокчейнде 8 биттік хеш қолданылса, ықтималдығы қаншага тең болады?

- A) 64
- B) 32
- C) 128
- D) 256
- E) 16

16. Блокчейнде 3 биттік хеш қолданылса, ықтималдығы қаншага тең болады?

- A) 64
- B) 8
- C) 128
- D) 256
- E) 16

17. Блокчейнде 10 биттік хеш қолданылса, ықтималдығы қаншага тең болады?

- A) 64
- B) 8
- C) 128
- D) 256
- E) 1024

18. Блокчейн транзакция блогындағы негізгі элемент:

- A) Тізбек
- B) Хеш
- C) Орталықсыздандыру
- D) Анықтамалық енгізу
- E) Қол қою

19. Блокчейнді колданатын сала:

- A) Онлайн дауыс беру
- B) Қаражат аудару
- C) Цифрландыру
- D) Онлайн сабак өткізу
- E) Ашық технология

20. Блокчейнде 5 биттік хеш қолданылса, ықтималдығы қаншаға тең болады?

- A) 64
- B) 32
- C) 128
- D) 256
- E) 16

Х БӨЛІМ. PYTHON ТІЛІНДЕ АЛГОРИТМДЕУ ЖӘНЕ ПРОГРАММАЛАУ

10.1. Python программауа тілінде тармакталған алгоритмдерді жазу

Күнделікті тұрмыста орындайтын іс-әрекеттің әрқашан сыйыктық болып ретімен орындаға бермейтін анық. Қөп жағдайда түрлі іс-әрекеттерді орындау барысында туындағын кедергілер бойынша шешім қабылдай білуіміз керек.

Тармакталған алгоритм – орындалуы керек кадам белгілі бір шарттың орындалуы немесе орындалмауына байланысты жүзеге асырылатын алгоритм.

Тармакталған алгоритмде «егер», «онда», «әйтпесе» түйінді сөздері колданылады. Тармакталған алгоритмде міндетті түрде шарт қолданылады.

Шарт дегеніміз – тексеру нәтижесінде «иә» (акыкат) немесе «жоқ» (жалған) логикалық мәндерін қабылдайтын логикалық өрнек немесе пікір.

Компьютерде программауа барысында берілген есептердің алгоритмдерін құрастыру кезінде оның мазмұнына қарай түрлі шарттар туындаиды. «Тармакталу» үғымы есептердің алгоритмін құрастыру барысында туындағын шарттар нәтижеге жеткенше бірнеше тармактарға ажырап кетуімен байланысты. Кез келген программауа тілінде міндетті түрде шартпен жұмыс істейтін операторлар болады. Сол сиякты Python программауа тілінде де шарт тексеруге арналған арнайы **if -else** операторы колданылады. Ағылшынның «if» сөзі «егер», ал «else» сөзі «әйтпесе» мағынасын береді. if-тен кейін шарт жазылады. Егер шарт орындалса, болса, онда көрсетілген команданы орындауда қажет. Шарт тексеру үшін төмендегі салыстыру операторлары қолданылады.

1-кесте. Салыстыру операторлары

Оператор	Мағынасы	Жазылды	Мысалы
=	Тен	$x = y$	$if a == b:$
!=	Тен емес	$x != y$	$k = a + b$
>	Үлкен	$x > y$	$if a >= b:$
<	Кіші	$x < y$	$k = a * b$
>=	Үлкен немесе тен	$x >= y$	$if a <= b:$
<=	Кіші немесе тен	$x <= y$	$k = a - b$

Шарт тексеру операторының туындағын шарттардың қурделілігіне қарай үш түрлі құрылымы бар. Біз әзірге екі құрылыммен ғана танысамыз.

Шарт тексеру операторының жазылу құрылымдары

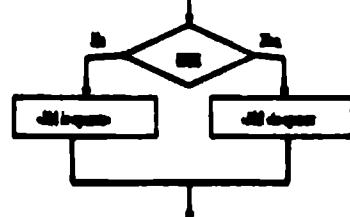
<p>1. Шарт тексеру операторының қыска жазылу түрі. if операторынан кейін шарт міндетті түрде жазылады. Шарттан кейін қос нұктес «:» койылады. 1-командалар блогы тек шарт ақыкат «иә» болғанда орындалады. Ал 2-командалар блогы кез келген</p>	<pre> graph TD D{Шарт} -- Иә --> R[Іс-әрекет] D -- Иә --> D </pre>
--	--

жагдайда орындалады.

Қысқа жазылу құрылымы

Шарт тексеру операторын алгоритм түрінде өрнектелген кезде, блок-схемада ромб фигуrasesы колданылады.

2. Шарт тексеру операторының толық жазылу түрі. Шарт тексеру операторының қысқа жазылу түрі тек туындаған шарттың ақиқаттығын тексереді. Шарт жалған болған кездегі орындалтуы керек қадамды көрсетпейді. Ол үшін шарт тексеру операторының ***if -else*** толық құрылымы колданылады.



Толық жазылу құрылымының блок-схемасы

«Егер койылған шарт орындалса, онда 1-команда немесе командалар блогын орында, эйттесе 2-команда немесе командалар блогын орында».

Тапсырмалар:

1-тапсырма

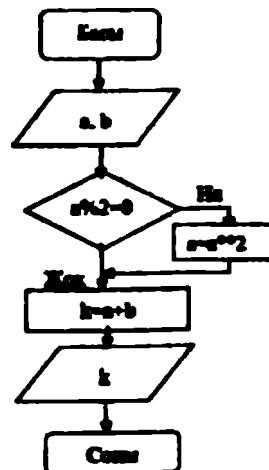
а және b бүтін сандар берілген. Егер а саны жұп болса, онда оны квадраттап, b санына қос. Ал сан тақ болса, онда санның өзін b санына қосып, нәтижені экранға шығар.

6	41	a=6; b=5; 6·6+5=41
5		a=7; b=5; 7+5=12

Тапсырманы орындағанда шарт тексеру операторының қысқа жазылу түрін қолдануға болады. а санының жұп немесе тақ сан екенін тексеру үшін, қалдықты анықтау «%» амалын қолданамыз.

```

a=int(input())
b=int(input())
if a*a+b%2==0:
    a=a*a+k=k=a+b
print(k)
  
```



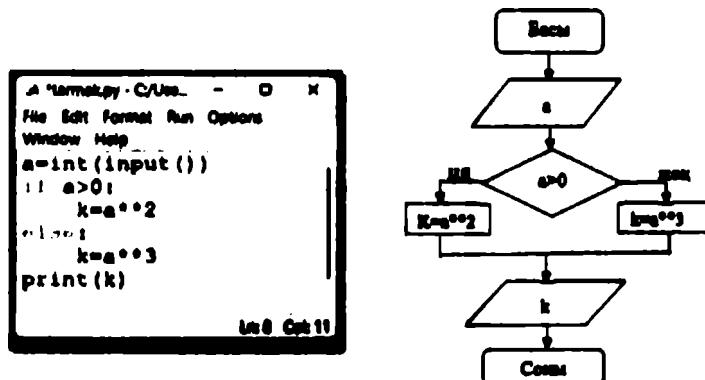
2-тапсырма

Накты а саны берілген. А саны он сан болса, квадратын, теріс сан болса, кубын есептеп, нәтижени экранға шығар.

9	81
-6	-216

Алгоритмдік шешу жолы

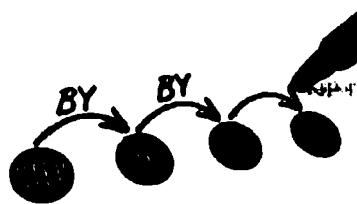
Тапсырманы орындауда шарт тексерудің толық құрылымын пайдаланамыз. Тапсырма бойынша $A > 0$, $A < 0$ шарттары туындайды. Осы екі шарттың біреуін тексеру жеткілікті. $A > 0$ шарты орындалмауына, ал $A < 0$ орындалуына алып келеді, өйткені сан не он, не теріс болады. Тапсырманы орындауда $A = 0$ жағдайы қарастырылған жок.

**10.2. Алгоритмнің трассировкасын жүзеге асыру****Алгоритм трассировкасы дегеніміз не?**

Программа жұмыс істеп тұрған кезде айнымалы мәндерді бақылай алу кодтың өзі істеу керек иәрсені істеп жатқанына көз жеткізуге және программаны бұзбайтын, бірақ күтпеген нәтижелерді беретін логикалық кателерді болдырмаяуга көмектеседі. Қолмен трассировкалау трассировка кестесін пайдалануға мүмкіндік береді:

- кодты іске қоспас бүрүн оның не істейтінін көру;
- кодтың қай жерінде кателер бар екенін табу.

Трассировка кестесі – логикалық кателердің болмауына көз жеткізу үшін алгоритмдерді тексеру үшін колданылатын әдіс.



Трассировка (жол тарту) – бағдарламаның біртінде, қадам бойынша (step-by-step) орындалу процесі.

Алгоритм трассировкасы – кездесетін катені табу және өндөу процесі.

Traceback (трассировка) — белгілі бір уақытта кодтағы орындалған функциялардан келіп түсетін хабарламалар туралы есеп.

```

Python 3.8.1 (tags/v3.8.1:1b286e, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
>= RESTART: C:\Users\Шельханова\AppData\Local\Programs\Python\Python38\wh-4.py
Traceback (most recent call last):
File "C:\Users\Шельханова\AppData\Local\Programs\Python\Python38\wh-4.py", line 6,
  in <module>
    print(a)
NameError: name 'a' is not defined
>>>

```

Python трассировканы (traceback) кодта кате болған кезде шығарады.

Қателер атаулары

NameError	Жергілікті және ауқымды айнымалылар табылмағанда шығады.	NameError: name 'a' is not defined
SyntaxError	Синтаксистік талдаушы синтаксистік катені анықтағанда шығады.	SyntaxError: invalid syntax

TypeError	Функция немесе операция типіне сәйкес емес нысанга колданылғанда шығады.	TypeError: object of type 'float' has no len()
ValueError	Функция немесе операция типі дұрыс, бірақ мәні дұрыс емес аргументті кабылдағанда шығады.	ValueError: invalid literal for int() with base 10: '45.5'
IndexError	Модульді жүктеу кезінде модуль аты дұрыс болмаған жағдайда шығады.	
ImportError	Модульді жүктеу кезінде модуль аты дұрыс болмаған жағдайда шығады.	ModuleNotFoundError: No module named 'mathem'
KeyError	Енгізілген немесе дайын кодтар жиынынан “кілт” табылмағанда шығады.	KeyError: 'b'

1-мысал:

Программа коды	Енгізу:	Нәтижесі:
<pre>a = int(input()) y = 5 if a * 2 > 100: a = a * 3 y = a * 4 print(a, y)</pre>	51	153 612

Осы алгоритм үшін трассировка кестесін құрастырайық. 1-жолда пайдаланушы «a» айнымалысының мәнін енгізеді, мысалы, 51. 2-жолда «y» айнымалысының мәні программамен орнатылады және 5-ке тең.

1	a = int(input())	Пайдаланушы енгізеді	51	мәнін	51	
2	y = 5					5

3-жолға өтейік. Енгізілген «a» саны 2-ге көбейтіледі, мәннің нәтижесі оның 100-ден үлкен болуы тексеріледі. Егер ол үлкен болса, кодтың келесі жолына көшу жүргізіледі. $51*2=102$, $102>100$, сондыктan шарт Ақиқат деп есептеледі.

1	<code>a = int(input())</code>	Пайдаланушы 51 мәнін енгізеді	51	
2	<code>y = 5</code>			5
3	<code>if a * 2 > 100:</code>	Өрнек Ақиқат деп бағаланады		

4-жолда «а» айнымалысының мәні 3-ке ұлттайтылған; «у» мәні өзгеріссіз, у = 5.

5-жолда $y = a^4$ өрнегі, себебі «а» айнымалысының жаңа мәні 153, $y = 153^4 = 612$.

6-жолда айнымалылардың мәндері экранга шыгарылады.

1	<code>a = int(input())</code>	Пайдаланушы 51 мәнін енгізеді	51	
2	<code>y = 5</code>			5
3	<code>if a * 2 > 100:</code>	Өрнек Ақиқат деп бағаланады		
4	<code>a=a * 3</code>		153	5
5	<code>y = a * 4</code>		153	612
6	<code>print(a, y)</code>			

153, 612 мәндері экранга шыгарылады

while циклі бар алгоритмнің трассировкасын қарастырайык.

2-мысал:

```
x = 5
y = 1
z = 0
while x > 3:
    x = x - 1
    y = y + 1
    z = (x + y) * 2
print(a, y)
```

Нәтижесі:

3 3 12

Бұл кодта x, y, z мәндері x > 3 болғанша өзгереді. Бастапқы x = 5 мәні және ол -1 қадамға азаяды. Соңғы нәтиже while циклінің шарты жалған болғанда басып шыгарылады. Трассировка кестесін қарастырайық:

1	$x = 5$	5			
2	$y = 1$		1		
3	$z = 0$			0	
4	while $x > 3$:				T
5	$x = x - 1$	4			
6	$y = y + 1$		2		
7	$z = (x + y) * 2$			16	
8	while $x > 3$:				T
9	$x = x - 1$	3			
10	$y = y + 1$		3		
11	$z = (x + y) * 2$			12	
12	while $x > 3$:				F
	print (x, y, z)	3	3	12	

$x > 3$ шарты Жалған мәніне бағаланбай тұрып, программа соңғы x, y, z мәндерін сактайды және оларды көрсетеді.

Трассировка кестесі кодты орындау ағынын имитациялады. Операторлар кезек-кезекпен орындалады және тағайындау операторы орындалған сайын айнымалы мәндер өзгереді.

10.3. Сұрыптау алгоритмдерін қолдану

Программалардағы массивтер деректерді сақтау, іздеу және өңдеу қызметін атқарады. Массивтер арқылы әртүрлі операцияларды жүзеге асыруға болады. Кейбір әрекеттерді жөнілдету үшін массив элементтерін белгілі бір принцип бойынша реттеу керек. Мұны сұрыптау арқылы жасай аласың.

Сұрыптау – массив элементтерін есу немесе кему ретімен жүйелеу.

- Есу ретімен сұрыпталғаннан кейін сандық деректер ең тәменгі мәннен жоғары мәнге дейін, мәтіндік мәліметтер алфавит бойынша сұрыпталады.
- Кему ретімен сұрыпталғаннан кейін сандық деректер ең үлкен мәннен ең тәменгі мәнге дейін, мәтіндік деректер кері алфавиттік тәртіппен орналасады.

Мысалы, екі массивті қарастырайық:

```
num=[8, -7, 12, 5, 9, 12, 11, 4]
```

```
city = ['Мәскеу', 'Лондон', 'Париж', 'Вена', 'Нұр-Сұлтан']
```

Өсу ретімен сұрыптау нәтижесі:

```
num = [-7, 4, 5, 8, 9, 11, 12, 12]
```

```
city = ['Вена', 'Лондон', 'Мәскеу', 'Нұр-Сұлтан', 'Париж']
```

Кему ретімен сұрыптау нәтижесі:

```
num = [12, 12, 11, 9, 8, 5, 4, -7]
```

```
city = ['Париж', 'Нұр-Сұлтан', 'Мәскеу', 'Лондон', 'Вена']
```

Сұрыптау үшін әртүрлі алгоритмді колдануға болады. Python-да әдістер мен кіріктірілген тізім функцияларын колдана отырып, сандық және мәтіндік массивтерді сұрыптауға болады.

Аталуы	Максаты
<code>sort()</code>	тізімді сұрыптау әдісі
<code>reverse()</code>	тізім элементтерін кері бағытта орналастыруға арналған әдіс
<code>sorted()</code>	тізімді өсу ретімен сұрыптауға арналған функция

`sort()` әдісі тізімді өзгертеді. Нәтижесінде тізім элементтерінің орындары ауысып, өсу ретімен орналасады.

1-мысал. Түстер массиві берілген. Түс атауларын алфавит бойынша орналастыр.

Программау коды:

```
A:\Python - C:/Users/Admin/OneDrive/Рабочий стол/баз дат шарты/таптау.py (3.10.5)
File Edit Format Run Options Window Help
color = ['blue', 'green', 'orange', 'gray', 'gold', 'red']
color.sort()
print (color)
```

Нәтижесі:

```
['blue', 'gold', 'gray', 'green', 'orange', 'red']
```

`sort()` әдісінің `reverse` косымша параметрі бар. Жакшаның ішіндегі `sort()` әдісін кему ретімен сұрыптау үшін `reverse = True` (акиқат) параметрінің сәйкес мәнін көрсету арқылы жүзеге асыруға болады.

2-мысал. Түстер массиві берілген. Түс атауларын кері алфавиттік ретпен орналастыр.

Программау коды:

```
A:\Python - C:/Users/Admin/OneDrive/Рабочий стол/баз дат шарты/таптау.py (3.10.5)
File Edit Format Run Options Window Help
color = ['blue', 'green', 'orange', 'gray', 'gold', 'red']
color.sort(reverse=True)
print (color)
```

Нәтижесі:

```
[‘red’, ‘orange’, ‘green’, ‘gray’, ‘gold’, ‘blue’]
```

Егер екі тізімді де сактау қажет болса (реттелген және реттелмеген), онда сұрыптау үшін *sorted()* функциясы қолданылады. *sorted()* функциясының айырмашылығы, ол тізімді өзгертпейді, бірақ жаңа тізім жасайды. Осылайша реттелген тізім негізінде жаңа тізім шығады.

З-мысал. Бағдарламада кездейсок 10 бүтін сандар тұратын массив жасалады. Осының негізінде элементтер өсу ретімен орналасатын жаңа массив құра.

Программалау коды:

```
Создано: Суббота Октябрь 2019, 10:10:29
File Edit Insert Options Window Help
A = [randint(10, 50) for i in range(10)]
B=sorted(A)
print (A)
print (B)
```

- O x

Lec 3 Code

Нәтижесі:

```
A = [47, 49, 15, 45, 26, 41, 12, 30, 42, 50]
```

```
B = [12, 15, 26, 30, 41, 42, 45, 47, 49, 50]
```

sorted() функциясына да *reverse=True* параметрін орнатуға болады. Мысалы, *B = sorted (A, reverse = True)* пәрмені арқылы А массивінің элементтерін кему ретімен орналастырып, жаңа В массивін құрауга болады.

10.4. Python программалау тілінде файлды оку және жазу

Python файлдарды құру, жазу және оку үшін кірістірілген функцияларды қамтамасыз етеді. Python-да өндеге болатын файлдардың екі типі бар: кәдімгі мәтіндік файлдар және екілік файлдар (екілік тілде жазылған, нөлдер және бірліктер).

<ul style="list-style-type: none"> Осы типтің файлдарында мәтіннің әрбір жолы EOL (End Of Line) арнағы символымен аяқталады, ол – Python-да әдепті жаңа жол символы ("\n"). Веб-стандарттар: html, XML, CSS Бастапқы код: c, app, js, py, java Күжаттар: txt, tex, RTF Кестелі деректер: csv, tsv Конфигурация: ini, cfg, reg 	<ul style="list-style-type: none"> Бұл типтің файлдарында жол шектеулері жок, деректер машина үшін түсінікті екілік тілге түрлендіруден кейін сакталады. Күжаттар файлдары: .pdf, .doc Кескіндер файлдары: .png, .jpg Бейнефайлдар: .mp4, .3gp, .mkv Аудиофайлдар: .mp3, .wav Дереккор файлдары: .mdb, .accde Архивтік файлдар: .zip, .rar, .7z Орындалатын файлдар: .exe, .dll
---	---

Файлға қол жеткізу режимдері

Қол жеткізу режимдері ашық файлда мүмкін болатын операциялардың типін аныктайды. Бұл файлды ашқаннан кейін қалай пайдаланылатынына байланысты. Бұл режимдер файлдағы файл дескрипторының орнын да аныктайды. Файл дескрипторы деректерді файлда қай жерден оку немесе жазу керектігін аныктайтын мәндер сияқты. Python-да файлға қол жеткізудің 6 режимі бар.

- 1. Тек оку үшін ('r'):** оку үшін мәтіндік файлды ашу. Қалам файлдың басында орналасқан. Егер файл болмаса, енгізу-шыгару қатесі орын алады. Бұл – сонымен қатар файл ашылатын әдепті режим.

- 2. Оку және жазу ('r +'):** файлды оку және жазу үшін ашу. Қалам файлдың басында орналасқан. Файл болмаса, енгізу-шыгару қатесі орын алады.

- 3. Тек жазу ('w'):** файлды жазу үшін ашу. Бар файл үшін деректер қысқартылады және қайта жазылады. Қалам файлдың басында орналасқан. Егер файл болмаса, файл жасалады.

- 4. Жазу және оку ('w +'):** файлды оку және жазу үшін ашу. Бар файл үшін деректер қысқартылады және қайта жазылады. Қалам файлдың басында орналасқан. Егер файл болмаса, файл жасалады.

- 5. Тек қосу ('a'):** файлды жазу үшін ашу. Егер файл болмаса, ол жасалады. Қалам файлдың соңында орналасқан. Жазылатын деректер бар деректерден кейін соңына қойылады.

- 6. Қосу және оку ('a +'):** файлды оку және жазу үшін ашу. Егер файл болмаса, ол жасалады. Қалам файлдың соңында орналасқан. Жазылатын деректер бар деректерден кейін соңына қойылады.

Бұл open () функциясы арқылы орындалады. Бұл функция үшін модульді импорттау қажет емес, мысалы:

```
File_object = open (г "Файлдың_аты" ("Имя_файла"), "Қол жеткізу_режимі"
("Режим_доступа"))
```

Файл Python бағдарламалық файлы орналасқан каталогта болуы керек, әйтпесе файл атауының орнына файлдың толық адресі жазылуы керек.

Python-да файлдарды оку және жазу open () әдісін түсінуді талап етеді. Бұл әдістің әмбебаптығын пайдалана отырып, Python тілінде файлдарды окуға, жазуга және жасауға болады. Файл деректері Python-ға жүктелгеннен кейін, онымен істеуге болатын заттар шексіз. Программалаушылар көбінесе файлдардың үлкен санымен жұмыс істейді, оларды автоматты түрде жасау үшін программаларды пайдаланады. Файлға деректерді жазу қажет болса, үш қадамды есте сактау жеткілікті:

- 1. Файлды ашу**
- 2. Оку/Жазу**
- 3. Файлды жабу**

Python тілінде файлды жазу мысалы келесідей:

```
f = open('example.txt', 'w')
f.write('hello world')
f.close()
```

Close () әдісі файлды жауып, сол файл пайдаланатын жадты босатады. Өткен сабакта файлды ашу қарастырылғандықтан, бұл сабакта файлдарды жазу мен оку қарастырылады.

Файлга жазу. Файлға жазудың екі әдісі бар.

1. **write ()** әдісі: мәтіндік файлға бір жолға str1 жолын қояды.

```
File_object.write(str1)
```

2. **writelines ()** әдісі: жол элементтерінің тізімі үшін әрбір жол мәтіндік файлға қойылады. Бірдей уақытта бірнеше жолдарды қою үшін қолданылады.

```
File_object.writelines(L) для L = [str1, str2, str3]
```

write () әдісінің қолданылуы:

Бұл мысал файлға деректерді жазу үшін write () әдісін пайдаланады. Файл жазу режимінде ашылады.

```
# жаңа файл объектісін жасау және файлды жазу режимінде ашу
file=open("example.txt","w")

# файлға бір жолды жазу
file.write("Welcome to Python \n")

# файлға тағы бір жолды жазу
file.write("Welcome back")

# файлды жабу
file.close()
```

write () әдісінің мысалы

Нәтижесі:

Welcome to the Python

Welcome back

Егер файлды жазу режимінде ашса және файлға көбірек жолдар жазу үшін write () функциясы қолданылса, ол алдыңғы деректерді қайта жазады және жаңа деректер мәтіндік файлға қосылады.

```
# жаңа файл объектісін жасау және файлды жазу режимінде ашу
file=open("example.txt","w")

# файлға бір жолды жазу
file.write("Hello Everyone \n")

# файлға тағы бір жолды жазу
file.write("This is the replaced string")

# файлды жабу
file.close()
```

write () әдісінің мысалы

Нәтижесі:

Нәтиже алдыңғы деректердің қайта жазылғанын және оның орнына мәтіндік файлға жаңа деректердің қосылғанын көрсетеді.

Hello Everyone

This is the replaced string

Файлда алдыңғы және жаңа деректердің сақтау қажет болса, файлды қосу режимінде ашуға болады, мысалы:

```
# жаңа файл объектісін жасау және файлды қосу режимінде ашу
file=open("example.txt","a")

# файлға бір жолды жазу
file.write("Hello Everyone \n")

# файлға тағы бір жолды жазу
file.write("This is the replaced string\n")

# файлға тағы бір жаңа жолды жазу
file.write("This is the newly added string\n")

# файлды жабу
file.close()
```

write () әдісінің мысалы

Нәтижесі:

Hello Everyone

This is the replaced string

This is the newly added string

writelines () әдісінің қолданылуды:

writelines () әдісі төменде көрсетілгендей мәтіннің бірнеше жолын бір уақытта жазу үшін қолданылады:

```

# жаңа файл объектісін хасу және фойлды жазу режимінде азы
file=open("file1.txt","w")

# айналауда бірнеше көм деректерді сақтау
str = ["Hello everyone\n", "Welcome to Python \n", "we are using writelines method\n"]

# файлға деректерді жазу үшін writelines әдісін колдану
file.writelines(str)

# файлды жабу
file.close()

```

writelines () әдісінің мысалы

Нәтижесі:

Hello everyone

Welcome to Python

we are using writelines method

Файлдан оку.

Мәтіндік файлдан деректерді оқудың үш әдісі бар.

1. **read ()** әдісі: оқылған байттарды жол ретінде қайтарады. н байттарды оқиды, егер н көрсетілмесе, бүкіл файлды оқиды.

File_object.read ([n])

2. **readline ()** әдісі: файлдан жолды оқиды және жол ретінде қайтарады. Көрсетілген н үшін ең көбі н байт оқиды. Дегенмен, н жолдың ұзындығынан үлкен болса да, бір жолдан артық оқымайды.

File_object.readline ([n])

3. **readlines ()** әдісі: барлық жолдарды оқиды және оларды тізімдегі жол элементі ретінде қайтарады.

File_object.readlines ()

Файл дескрипторының орнын өзгерту үшін seek () әдісі қолданылады. Файлдағы деректерді оку кезінде файл дескрипторы файлдың соңында орналасады. Осылайша, файл дескрипторы курсор сияқты, ал seek () әдісі курсорды жылжыту құралы ретінде қызмет етеді. Файлдан деректерді оқудың мысалы:

```

# файлды оку разжымда ашу
file=open("file1.txt","r")

# файлдан деректерді оку үшін read() елісін колдану
# жолдарды сыйында сәктау
data=file.read()

# деректерді шыгару
print("This is the output of read() function: \n")
print(data)

# позицияны файлдың басына шыгару
file.seek(0)

# файлдан деректерді оку үшін readline() елісін колдану
# жолдарды сыйында сәктау
data=file.readline()

# деректерді шыгару
print("This is the output of readline() function: \n")
print(data)

# позицияны файлдың басына шыгару
file.seek(0)

# файлдан деректерді оку үшін readlines() елісін колдану
# жолдарды сыйында сәктау
data=file.readlines()

# деректерді шыгару
print("This is the output of readlines() function: \n")
print(data)

# файлды жабу
file.close()

```

Файлдан оку

Нәтижесі:

This is the output of read() function:
Hello everyone
Welcome to Python
we are using writelines method

This is the output of readline() function:

Hello everyone

This is the output of readlines() function:

["Hello everyone\n", "Welcome to Python \n", "we are using writelines method\n"]

Файлдан оку

Көбінесе әқпаратты немесе деректерді файлда сәктау қажет. Python-да Python-ға енгізілген функцияларын пайдаланып файлдарды оцай жасауга, жазуға және оқуға болады. Файлдарды оку, жазу және жасау керек болса, программаға басқа модульдерді импорттаудың қажеті жок.

10.5. Python программалау тіліндегі кірістірілген шарттарды колдану

Python программалары if операторлары арқылы негізгі «егер солай болса, осылай жаса» әрекетін жасайды. Осылайша код шешімді орындау уақытында қабылдайды. Кірістірілген шарттар – басқа if немесе if / else операторының ішіне кірістірілген if операторы. Бұл тұжырымдамалар шарттардың ақиқат / жалған екендігін тексереді, содан кейін тиісті әрекетті қабылдайды. Қалыпты if немесе if / else операторы болған кезде шарт тексеріліп, кейбір код орындалады. Кірістірілген if операторы да алдыңғы if қалай бағаланатынына байланысты. Бұл – бірінен соң бірі тексерілетін шарттарды жүзеге асырудың бір жолы. Сондай-ақ ол программага if операторлары арасында кодты орындау мүмкіндігін береді. Кірістірілген if операторын колданудың екі негізгі әдісі бар:

- if операторын if код блогының ішіне орналастыру.
- if операторын if / else операторының else кодына орналастыру.

Шарт тексеру операторының құрамында екінші шарт тексеру операторы колданылса, онда мұндай шарт тексеру кірістірілген шарт тексеру деп аталады.

1-әдіс. Кірістірілген шарттардың құрылымы:

1-мысал: а саны берілген. А саны он немесе теріс сан екенин анықта. Егер а саны он сан болса - «1», теріс сан болса - «-1», ал сан нөлге тең болса, «0» деген жауап шығатын болсын.

```
if 1-шарт:
    if 2-шарт:
        1-командалар блогы
    else:
        2-командалар блогы
```

```
if a>0:
    b=int(input())
    c=int(input())
    k=0
    if a2==0:
        if b2==1:
            k=a+b
        else:
            k=a+b;
    print (k)
```

```
if 1-шарт:
    1-командалар блогы
else:
    if 2-шарт:
        2-командалар блогы
    else:
        3-командалар блогы
```

```
if a>0:
    print(1)
else:
    print(-1)
```

2-әдіс. Кірістірілген шарттардың құрылымы:

2-мысал: а мен b саны берілген. Егер a саны 0-ден артық болып, екі санның айырмасы да 0-ден артық болса, онда екі санның айырмасын, әйтпесе қосындысын экранға шығар.

```
If 1-шарт:
If 2-шарт:
    1-командалар блогы
else:
    2-командалар блогы
else:
    3-командалар блогы
```

```
Вашему - Старт/Админ/Онлайн
File Edit Forms View Options Window Help
a=int(input())
b=int(input())
if a>0:
    if a-b>0:
        a=a-b
    else:
        a=a+b
else:
    a=a**2
print(a)
```

3-әдіс. Кірістірілген шарттардың күрылымы:

3-мысал: мысалдың орындалуы а санының жұп не тақ болуына байланысты. Егер a саны жұп болса ғана одан кейін b-ның мәні тексеріледі. Ал а саны тақ болса, нәтиже ретінде экранға ештеңе шықпайды.

1-мысал: Пайдаланушының көлік айдауга жасы жететіндігін тексеру коды.

```
Вашему - Старт/Админ/Онлайн/Редактор старт/для пакет/Python (3.7.3)
File Edit Forms View Options Window Help
age = 19
aiaktadi = False
kualikbar = True
# Нәтижесін 10 шарттан есептей тексеру
if age>=10:
    print("Sizdin jasiniz 10-de. Kamelet jaska kelgensiz!")
    if aiaktadi:
        print('Jas tulaksiz!')
    if kualikbar:
        print('kolik aidauga boladi!')
```

Нәтижесі:

Sizdin jasiniz 10-de. Kamelet jaska kelgensiz!
kolik aidauga boladi!

Бұл код алдымен адамның ересек адам екенін тексереді. Содан кейін оның жоғары білім немесе жүргізуши күелігін алғанын білу үшін қосымша салыстыру жүргізіледі. Кодта уш айнымалы мәні жарияланады: `age = 19` мәніне ие. `aiaktadi = False` және `kualikbar = True` мәніне ие. Содан кейін екі кірістірілген if

операторы орындалады. Біріншісі `isGraduated` айнымалысын тексереді. Бұл айнымалы мән `False` болғандықтан, `if` операторының коды орындалмайды. Басқа кірістірілген `if` операторы `kualikbar`-ды іздейді. Бұл айнымалы мәні `True`, сондықтан Python өз кодын орындаиды. Оnda `print()` функциясы адамға сәтті көлік жүргізуі тілейді. Одан кейін адамның ересек екенін тексеру қажет емес екенін есте сақтау керек; бұл – бірінші `if` тұжырымдамаларының бірінде жасалған нәрсе. Осылайша «`kolik aidauga boladi!!`» хабарламасы тек ересектерге ғана көрсетілетін болады.

10.6. Python программалау тіліндегі күрделі (курама) шарттарды қолдану

Python математикадан әдеттегі логикалық шарттарды қолдайды, кодта олар келесідей жазылады:

- Тен: `a == b`
- Тен емес: `a != b`
- Кем: `a < b`
- Кем немесе тен: `a <= b`
- Артық: `a > b`
- Артық немесе тен: `a >= b`

Бұл шарттарды әртүрлі тәсілдермен қолдануға болады, көбінесе `if` операторлары мен циклдерінде қолданылады.

Кұрама шарттарды екі немесе одан да көп логикалық өрнектерді байланыстыратын **AND**, **OR**, **XOR**, **NOT** логикалық операторлары арқылы жүзеге асыруға болады. Нәтижені анықтау үшін әрбір операторға ақиқат қестесінің күрьымын түсіну қажет. Ақиқат кестесі күрделі тұжырымдамаларың ақиқаттығы немесе жалғандығы ол күрастырылған қарапайым тұжырымдамалардың ақиқаттығына немесе жалғандығына байланысты екенін көрсетеді.

Логикалық көбейту (коњюнкция-«`and`») дегеніміз – екі немесе одан да көп пікірлерді «және» жалғаулық шылауының көмегімен біріктіру амалы.

AND логикалық операторы үшін ақиқат кестелері:

1-ші логикалық өрнек	2-ші логикалық өрнек	Нәтижесі
Ақиқат	Ақиқат	Ақиқат
Ақиқат	Жалған	Жалған
Жалған	Ақиқат	Жалған
Жалған	Жалған	Жалған

Осылайша `if` операторының нәтижесі логикалық өрнектердің екеуі де ақиқат болғанда ғана ақиқат болады.

Екі шартты біріктіру үшін
AND операторының мысалы:

```
De tommy - C:\Users\Adm\OneDrive\Рабочий стол\Блокнот\3.txt
File Edit Format Run Options Windows Help
x = True
y = True
if x and y:
    print("Both x and y are True")
else:
    print("False")
```

Нәтижесі:
Both x and y are True

Үш және одан көп шартты, біріктіру үшін AND
операторының мысалы:

```
De tommy - C:\Users\Adm\OneDrive\Рабочий стол\Блокнот\3.txt
File Edit Format Run Options Windows Help
a = 15
b = 27
c = 33
d = 42
if (a < b) and (b < c) and (c < d):
    print("All 3 expressions are True")
else:
    print("False")
```

Нәтижесі:
All 3 expressions are
True

Мұндағы:

- (a < b) Ақиқат деп бағаланады, себебі 15 27-ден кем
- (b < c) Ақиқат деп бағаланады, себебі 27 33-тен кем
- (c < d) Ақиқат деп бағаланады, себебі 33 42-ден кем

Логикалық қосу (дизъюнкция-«ог») дегеніміз – екі немесе одан да көп пікірлерді «немесе» жалғаулық шылауының көмегімен біріктіру амалы.

OR логикалық операторы үшін ақиқат кестелері:

Ақиқат	Ақиқат	Ақиқат
Ақиқат	Жалған	Ақиқат
Жалған	Ақиқат	Ақиқат
Жалған	Жалған	Жалған

Осылайша түпкілікті нәтиже Ақиқат болуы үшін логикалық өрнектердің кем дегенде біреуі Ақиқат болуы керек.

Екі шартты біріктіру үшін OR операторының мысалы:

Нәтижесі:
Eki ornekten kем degende bireui True-di beredi

```
Q "менеджер - Старт/Адрес/Скачать/Редактор скриптов для Python/PyCharm (PyCharm)" - □
File Edit Run File Options Help Help, New
a = 10
b = 20
c = 30
if (b > a) or (b > c):
    print ("Егер оның таңбасы көбайында болса Тұрақты болады")
else:
    print ("False")
```

Мұнда өрнек:

- (b> a) Ақықат деп бағаланады;
- (b> c) Жалған деп бағаланады.

Логикалық теріске шығару (инверсия-«not») «Емес» сезінің көмегімен пікірді теріске шығару.

Тапсырма. Жазықтықтағы А нүктесі x, у координаталарымен берілген. Осы нүктенің координата жазықтығының қай ширегінде жатқанын анықта.

Алгоритмдік шешу жолы А нүктесінің қай ширекте жатуы x, у координаталарының таңбасына байланысты. Мысалы: А нүктесі II ширекте жатса, онда x-тің таңбасы (-), ал у-тің таңбасы (+) болуы керек. Практикалық жұмысты

компьютерде орындал, нәтижесін ал.

Мысалы	Нәтиже
-6 7	II shirek

```
Q "менеджер - Старт/Адрес/Скачать/Редактор скриптов для Python/PyCharm (PyCharm)" - □
File Edit Run File Options Help Help, New
x, y = map(int, input("X и Y координаты: ").split())
if x < 0 and y < 0:
    print ("III shirek")
if x < 0 and y > 0:
    print ("II shirek")
if x > 0 and y < 0:
    print ("IV shirek")
if x > 0 and y > 0:
    print ("I shirek")
```

Жазықтықтың координаттары

10.7. While цикл операторын пайдалану

While циклі операторлар тізбегін белгісіз мөлшерде бірнеше рет қайталау үшін қолданылады. While циклін жазғанда қанша итерация орындалатынын нақты анықтамаймыз, тек процесті жалғастыру үшін Ақықат және оны токтату үшін Жалған болуы керек шарт жазамыз. Итерация кодтың бір блогын қайта-қайта, мүмкін бірнеше рет орындауды білдіреді. Итерацияны жүзеге асыратын

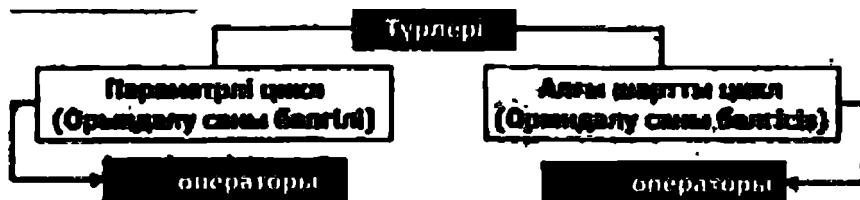
программалау құрылымы цикл деп аталады. Программалауда итерацияның анықталмаған және анықталған екі түрі бар.

Анықталмаған итерациялар

- Анықталмаған итерация кезінде циклді орындау саны алдын-ала анық көрсетілмейді. Керісінше, тағайындалған блок қандай да бір шарт орындалып жатқан кезде бірнеше рет орындалады.

Анықталған итерациялар

- Белгілі бір итерация кезінде көрсетілген блоктың орындалу саны цикл басталған кезде нақты көрсетіледі.



While циклдерін пайдалану қажеттілігінің кейбір мысалдары:

Пайдаланушы енгізуі *	Іздеу	Ойындар
Пайдаланушы енгізуін сұраған кезде енгізілген мәннің жарамдылығын тексеру керек. Программаны жалғастыру үшін пайдаланушы қате енгізуіді қанша рет енгізетінін алдын-ала білу мүмкін емес.	Деректер құрылымындағы элементті іздеу – уақытша цикл үшін басқа пайдалану жағдайы, ейткені мақсатты мәнді табу үшін қанша итерация қажет бола-тынын алдын-ала біле алмаймыз. Мысалы, екілік іздеу алгоритмі while циклінің көмегімен жүзеге асырылуы мүмкін. Екілік іздеу – элементтердің сұрыпталған тізімінен элементті табудың тиімді алгоритмі.	Ойында ойыншы жеңілгенше немесе ойын аяқталғанша ойынның негізгі логикасын сақтау үшін уақытша циклді пайдалануға болады. Мұның қашан болатынын алдын-ала біле алмаймыз, сондықтан бұл – уақытша циклге арналған басқа сценарий.

While циклі

While-Әзірше

While<шарт>

<1-оператор не өрнек>

<2-оператор не өрнек>

While («Әзір») цикл синтаксисі:

While (шарт)

{

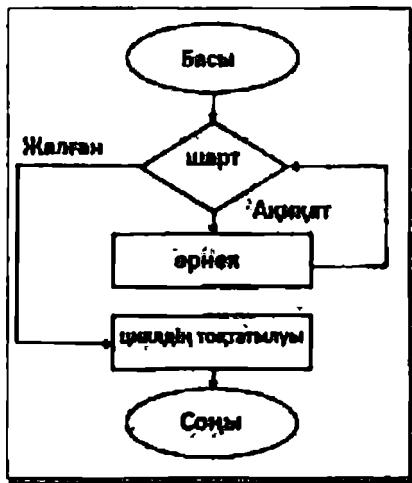
...	Цикл денесі }
<п-оператор не өрнек>	

While циклінің ерекшелігі

- Циклдің қайталау шарты цикл денесі жұмыс істемей тұрып тексеріледі

Шарт цикл цикл саны белгісіз болған жағдайларды қарастырганда қолданылады.

While циклінің блок-схемасы:

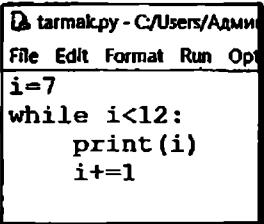


While циклінің жұмыс алгоритмі:

1. Процесс программаны орындау кезінде уақытша цикл кездескен кезде іске қосылады.
 2. Көрсетілген шарт оның Ақиқат немесе Жалған екенін тексеру үшін бағаланады.
 3. Шарт Ақиқат болса, циклге жататын операторлар орындалады.
 4. while циклінің шарты қайтадан тексеріледі.
 5. Шарт қайтадан Ақиқат болса, операторлар тізбегі қайтадан орындалады және процесс қайталаңады.
 6. Шарт Жалған мәнін шығарса, цикл тоқтап, программа циклден тыс жалғасады.
- while циклдерінің маңызды сипаттамаларының бірі – цикл жағдайында қолданылатын айнымалылар автоматты түрде жаңартылмайды. Шарт Жалған мәнінде бағаланған кезде цикл ақырында тоқтайтынына көз жеткізу үшін олардың мәндерін кодпен нақты жаңарту керек.

While циклінің мысалдары

1-мысал. Бізде k мәні 12 ($i < 12$) болғанда k мәнін шығаратын негізгі while циклі бар:	Программа коды	Нәтижесі
---	----------------	----------

 <pre>i=7 while i<12: print(i) i+=1</pre>	7 8 9 10 11
Код іске қосылғанда не болатынын көрейік:	<ul style="list-style-type: none"> • 1-итерация: бастапқыда k мәні 7, сондықтан $k < 12$ шарты Ақиқат мәнін береді және цикл басталады. k мәні басып шығарылады (7) және бүл мән 1-ге артады. Цикл қайтадан басталады. • 2-итерация: енді k мәні 5, сондықтан $k < 12$ шарты Ақиқат мәнін береді. Циклдің денесі орындалады, k мәні басып шығарылады (8) және k мәні 1-ге артады. Цикл қайтадан басталады. • 3, 4, 5-итерация: Үшінші және төртінші итерация үшін бірдей процесс қайталанады, сондықтан 9, 10 және 11 бүтін сандар басып шығарылады. • Алтыншы итерация басталғанға дейін k мәні 12-ге тең. Енді $k < 12$ кезінде цикл шарты Жалған мәніне бағаланады және цикл тоқтайды. Бірінші итерацияны бастамас бұрын while циклінің шарты Жалған болса, цикл басталмайды.
2-мысал. 1-ден 10-га дейінгі сандарды экранға шыгаратын программа құрыңыз	
Программа коды	Нәтижесі

<pre>tarmak.py - C:/Users/Админ/OneDrive/Рабочий стол/указание</pre>	File Edit Format Run Options i=0 while i<=10: print(i) i=i+1	0 1 2 3 4 5 6 7 8 9 10
--	--	--

3-мысал. Сандық салыстырудан ғори тізім пайдаланылатын while циклі:

Программа коды	Нәтижесі
<pre>tarmak.py - C:/Users/Админ/OneDrive/Рабочий стол/указание</pre>	File Edit Format Run Options Window Help baz bar foo

Тізім логикалық контексте бағаланған кезде, онда элементтер бар болса, ол ақырат болады, ал егер бос болса, жалған болады. Бұл мысалда элементтер бар болса, "a" ақырат болады. Барлық элемент .pop() әдісі арқылы жойылғаннан кейін және тізім бос болған соң, "a" жалған болады және цикл аяқталады.

4-мысал.	Нәтижесі:
<pre>tarmak.py - C:/Users/Админ/OneDrive/Рабочий стол/указание</pre>	File Edit Format Run Options Window Help ('baz', 3) ('bar', 2) ('foo', 1)

.popitem() әдісі d ішінен бір кілт-мән жүбін жояды және оны кортеж ретінде қайтарады. Осылайша while циклінің негізгі 'бөлігі' мазмунды d кортеждері ретінде көрсетеді.

Егер while циклінде көрсетілген шарт ешқашан жалған болмаса, while циклі ешқашан аяқталмайды және шексіз циклге айналады. while цикліндегі кез келген нәл емес мән әрқашан ақырат шартты көрсетеді, ал нәл әрқашан жалған шартты көрсетеді.

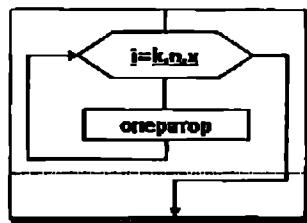
10.8. For цикл операторын пайдалану

Параметрлі цикл қай кезде қолданылады?

Бағдарламаны құру кезінде қайталану саны алдын-ала белгілі болса, онда параметрлі цикл операторы қолданылады.

Параметрлі цикл жұмысын үйымдастыру үшін for операторын range() функциясымен бірге қолдану керек.

Параметрлі цикл



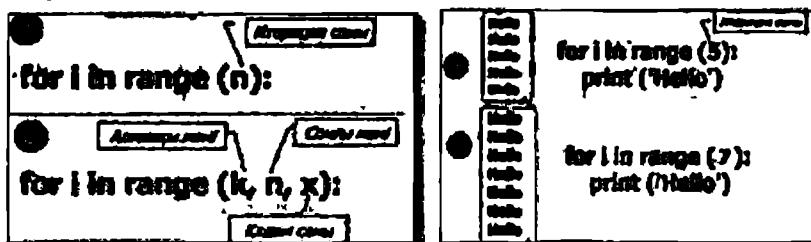
k-цикл параметрінің алғашқы мәні

n-цикл параметрінің соңғы мәні

x-қадам саны

for-үшін

range-диапазон



Цикл қатары k-дан n-1-ге дейін қадаммен өзгереді. Цикл итерацияда көрсетілген мәнге дейін жұмыс істейді.

Итератор – байланысты итеративті нысаннан ретті мәндерді шыгаратын мән өндірушісі. Кіріктірілген next() функциясы итератордан келесі мәнді алу үшін пайдаланылады.

I-мысал:	Нәтижесі:
<pre>a = ['foo', 'bar', 'baz'] itr = iter(a) itr <list_iterator object at 0x031EFD10></pre>	<pre>next(itr) 'foo' next(itr) 'bar' next(itr) 'baz'</pre>

Бұл мысалда [V]a – қайталанатын тізім, itr – iter() арқылы алынған байланысты итератор. Әрбір келесі қонырау (itr) одан келесі мәнді алады. Итератор ішкі күйді қалай сақтайтынына назар аудару керек. Ол қандай

мәндердің қабылданғанын біледі, сондықтан `next()` шақырылғанда, келесі қандай мәнді қайтаруды біледі. Итератордан алынған барлық мән қайтарылған болса, `next()` келесі шақыру `StopIteration` ерекші жағдайын шақырады. Итератордан мәндерді алудың кез келген басқа әрекеттері сәтсіз болады.

Итератордан барлық мәнді бірден алу қажет болса, кіріктірілген `list()` функциясын пайдалануға болады. Басқа ықтимал пайдаланулардың қатарында `list()` итераторды дәлел ретінде қабылдайды және итератор қайтарған барлық мәннен туратын тізімді қайтарады.

2-мысал:	Нәтижесі:
<pre>a = ['foo', 'bar', 'baz'] itr = iter(a) list(itr)</pre>	['foo', 'bar', 'baz']

Сол сиякты кіріктірілген `tuple()` және `set()` функциялары итератор алған барлық мәннен сәйкесінше кортеж бен жиынды қайтарады:

3-мысал:	Нәтижесі:
<pre>itr = iter(a) set(itr)</pre>	{'baz', 'foo', 'bar'}

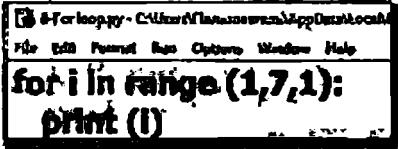
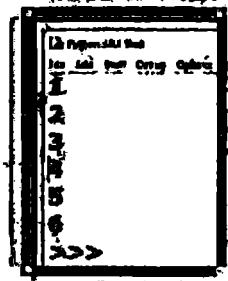
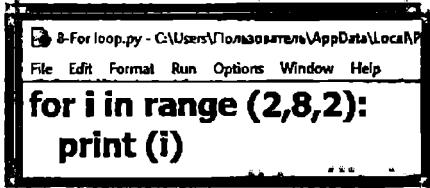
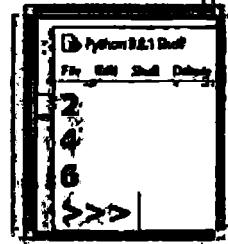
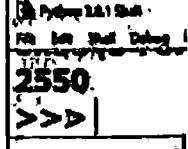
`list()`, `tuple()` және т.б. пайдаланған кезде сен итераторды оның барлық мәнін қайтару үшін бірден жасауга мәжбүрлесің. Итератор қайтарған нысандардың жалпы саны өте үлкен болса, ол көп уақыт алтуы мүмкін.

Негізгі сөздер	Мағынасы
Итерация	Жинақтағы нысандар немесе элементтерді қайталау процесі
Итерабельді	қайталанатын нысан (немесе объекттің сипаттау үшін қолданылатын сын есім)
Итератор	байланысты итератордан ретті элементтерді немесе мәндерді шығаратын нысан
<code>iter()</code>	қайталанатын нысаннан итераторды алу үшін қолданылатын кіріктірілген функция

Шындығында, Python тілінде генератор функциялары мен `itertools` көмегімен нысандардың шексіз сериясын қайтаратын итераторды жасауга болады. Егер шексіз итератордан барлық мәнді бірден алуға тырысса, программа қатып қалады.

Мысалдар:

1 ден 10-ға дейінгі сандарды кему бойынша экранға шығару кодын жазыңыз.

Программа коды	Нәтижесі
<pre>for i in range (10,0,-1): print (i)</pre> <p>алғашқы мәні = 10 соңғы мәні = 0 қадам саны = -1</p>	
	
	
<p>2-ден 100-ге дейінгі жүп сандарды шығаратын бағдарлама кодын жазыңыз.</p> <pre>for i in range (2,101,2): print (i)</pre> <p>алғашқы мәні = 2 соңғы мәні = 101 қадам саны = 2</p>	<p>2 4 6, 100</p>
<p>2-ден 100-ге дейінгі жүп сандардың косындысын табатын бағдарлама кодын жазыңыз.</p> <pre># Python 3.1.1 IDLE File Edit Format Run Options Window Help for i in range (2, 101, 2): print(i)</pre>	
Қабаттасқан циклдер	

```

Документ - C:\Users\Aman\OneDrive\Рабочий стол\for_continue.py
Настройки для Python Windows Help
for i in range(1, 6, 2):
    print(i, '- тақсан')
for i in range(2, 7, 2):
    print(i, '- жупсан')

```

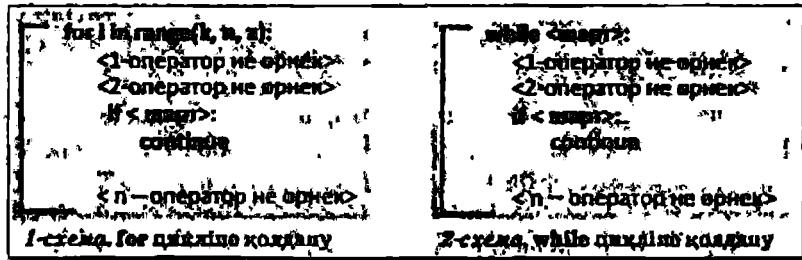
- 1 - тақсан
3 - тақсан
5 - тақсан
2 - жупсан
4 - жупсан
6 - жупсан

10.9. Цикл басқару нұсқаулығын қолдану (continue, break, else)

Continue циклін басқару

Программалаудағы циклдер команданы қайта-қайта қайталау үшін қолданылады. Белгілі бір итерацияны өткізіп жіберу немесе белгілі бір шарт орындалған кезде циклден шығу сияқты осы циклдердің ағынын басқару үшін Python бірнеше басқару операторын береді. Бұл тапсырмаларды үзу (break) немесе жалғастыру (continue) операторлары арқылы орындауга болады.

Continue операторы шарт орындалған кезде белгілі бір итерацияны өткізіп жіберу үшін пайдаланылады. Ол тоқтап, ағымды циклдің басына қайтарады. Continue операторы белгілі бір итерация үшін кодтың қалған белігін өндемейді және келесі итерация үшін ағынды қайта іске қосады.



1-мысал:

Бізде бүтін сандар тізбегі бар делік. Мән 3-ке тең болса, өндеуді өткізіп жіберу керек. Бұл сценарийді for циклі мен continue операторы арқылы орындауга болады.

```

Документ - C:\Users\Aman\OneDrive\Рабочий стол\for_continue.py
Настройки для Python Windows Help
numbers = (1, 2, 3, 4, 5)
for k in numbers:
    if k == 3:
        continue
    print('Output numbers [k]')
print("End")

```

Нәтижесі:

Output numbers 1
Output numbers 2
Output numbers 4
Output numbers 5
End

Нәтижеде көрініп тұргандау, Python $k = 3$ үшін итерацияны өткізіп жіберіп және кодтың қалған бөлігін орындаپ, жұмысын жалғастырады. Цикл аяқталмайды, келесі итерациямен жалғасады. Continue операторын for және while циклдерімен бірге қолдануға болады.

2-мисал:

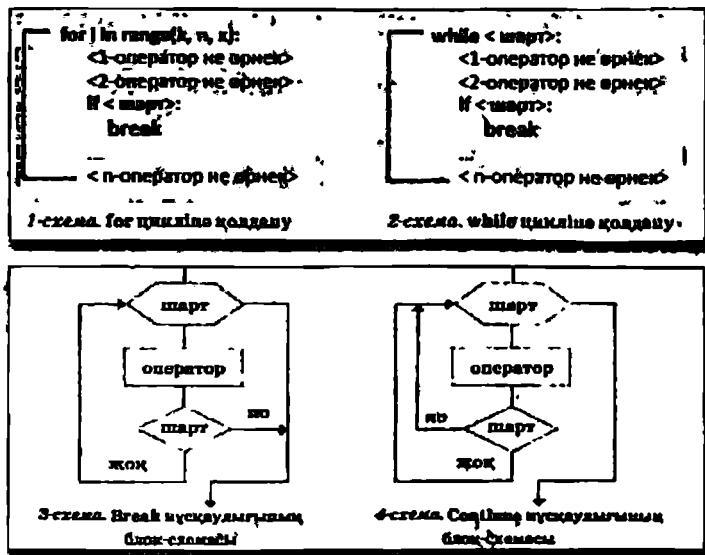
<pre>ta tarmak.py - C:/Users/Админ/OneDrive/Рабочий стол/все для юнити/tarmak.py (3.10.5) File Edit Format Run Options Window Help var=10 while var>0: var=var-1 if var==5: continue print ("Current variable value : ", var) print ("Good bye!")</pre>	<p>Нәтижесі:</p> <p>Current variable value : 9 Current variable value : 8 Current variable value : 7 Current variable value : 6 Current variable value : 4 Current variable value : 3 Current variable value : 2 Current variable value : 1 Good bye!</p>
--	---

Python-да break және continue операторлары қалыпты цикл ағынын өзгерте алады. Циклдер код блогын сынақ өрнегі жалған болғанша тере береді, бірақ кейде сынақ өрнегін тексермей, ағымдағы итерацияны немесе тіпті бүкіл циклді аяқтау қажет. Бұл жағдайларда break және continue операторлары қолданылады. Келесі сабактарда break операторы қарастырылады.

break циклін басқару

for циклі белгілі бір рет қайталанады және шарт ақырат болғанша біраз уақыт жұмыс істейді. Дегенмен егер пайдалану жағдайы циклді алдын-ала бұзуды талап етсе, Python-дағы break операторы кемекке келеді. break операторы қамтитын циклді аяқтайды. Программаны басқару цикл деңесінен кейін бірден операторға беріледі. Егер break операторы кірістірілген цикл ішінде болса, break операторы ең ішкі циклді тоқтатады.

break операторлары – циклді тоқтатып, келесі операторды орындауды сұрау тәсілі. Python үзіліс операторлары көбіне if операторымен бірге қолданылады. Себебі көп қолдану жағдайларында белгілі бір шарт орындалғандағанда ағымды ұзу талап етіледі.

**1-мисал:**

```
File Edit Format Run Options Window Help
for val in "string":
    if val == "i":
        break
    print(val)
print("The end")
```

Нәтижесі:

```
s
The end
t
The end
r
The end
```

Бұл программа «string» элементтерінің тізбегін қайталайды. і әрпінің баржыны тексеру жүргізіледі, содан кейін циклден шығады. Сондыктан шығыста і-ге дейінгі барлық әріп басып шыгарылады. Осыдан кейін цикл тоқтатылады.

break операторы ағымдағы циклді тоқтатады және С тіліндегі дәстүрлі break сияқты келесі операторда орындалуды жалғастырады. Break-тің ен жиі қолданылуы циклден тез арада шығуды талап ететін кейбір сыртқы жағдай іске қосылған кезде болады. break операторы for және while циклдерінде қолданылады.

2-мисал:

```
File Edit Format Run Options Window Help
var=10
while var > 0:
    print ('Current variable value :', var)
    var=var-1
    if var == 3:
        break
    print ("Good bye!")
```

Нәтижесі:

```
Current variable value: 10
Current variable value: 9
Current variable value: 8
Current variable value: 7
Current variable value: 6
Good bye!
```

Жоғарыда көрсетілгендей, цикл ақиқат болғанда 5 рет орындалады, ол ақиқат болғанда break операторы орындалады және циклден шығады. break үшін

жіңі қолданылатын жағдайлардың бірі – «output» сөзін терген кезде циклді табу және одан шыгу.

`break` және `continue` циклдердің ағынын басқаруға мүмкіндік береді. `break` операторы езі тұрган циклді тоқтатады, сондықтан кірістірілген циклді пайдаланған кезде `break` операторы бар цикл ғана тоқтатылады.

Else циклін басқару

Python цикл операторларымен байланысты `else` операторын қолдайды. Егер `else` операторы `for` циклімен бірге пайдаланылса, цикл тізімнің итерациясын аяқтаған кезде `else` операторы орындалады. Else блогы циклде үзіліс болмаған кезде ғана орындалады.

Жалпы дизайн циклді іске қосу және элементті табу болады. Егер элемент табылса, цикл `break` операторы арқылы шығарылады. Цикл аяқталуы мүмкін екі сценарий бар:

1. зат табылып, `break` орын алғанда;
2. цикл `break` операторын кездестірмей аяқталғанда.

Қайсысының циклдің аяқталуына әкелетінін анықтаудың бір жолы – жалаушаны орнату және цикл аяқталғаннан кейін оны тексеру. Екіншісі – `else` операторын пайдалану. `For/else` циклінің негізгі құрылымы келесідей:

for элемент in қойма:

if іздеу(элемент):

процесс(элемент)

`break`

`else:`

қоймада_табылмады()

1-мысал:

Программа коды

```
В окнеantu - C:/Users/Admin/Desktop/Рабочий стол/Python для начинающих
Реактивные языки Python для Windows 10
for n in range(2, 10):
    for x in range(2, n):
        if n % x == 0:
            print( n, 'делит', x, '=', n/x)
            break
        else:
            print(n, 'бүл жай сан')
```

Нәтижесі:

ВЕСТАЛЫК:
3 бүл жай сан
4 төң 2 + 2.0
5 бүл жай сан
6 бүл жай сан
7 бүл жай сан
8 бүл жай сан
9 бүл жай сан
10 бүл жай сан
11 бүл жай сан
12 бүл жай сан
13 бүл жай сан
14 бүл жай сан
15 бүл жай сан
16 бүл жай сан
17 бүл жай сан
18 бүл жай сан
19 бүл жай сан
20 бүл жай сан

Бұл код 2-ден 10-га дейінгі сандар үшін көбейткіштерді табады. Бөлгіштері жоқ сандарды табатын `else` блогын қосса, ол «бүл жай сан» хабарламасымен сандарды басып шығарады. Else сөйлемі итерация аяқталғаннан кейін `for` циклі аяқталғанда орындалады.

2-мысал:

Программа коды	Нәтижесі:
<pre> # Python - C:\Windows\system32\cmd.exe' настолько медленно # Вызов меню для Options Windows Help leaders = ["Аннали", "Илья", "Кастан"] for i in leaders: if i == "Диляра": print("Диляра - красавица!") break else: print("Проверка не удалась!") </pre>	Dilara tabilmadi!

Жоғарыда көрсөтілгендей, leaders тізімінде «Dilara» жоқ, сондықтан for циклі үзілмейді және тиісті ақпаратты шығаратын «else» орындалады. Dilara тізімге кіргендегі кодты қарастырайық. Белгілі бір элементті табу үшін тізімді анындауды – цикл үшін негізгі пайдалану жағдайы. Әдетте, элемент табылған кезде қайталауды жалғастырудың қажеті жоқ және циклден шығу керек.

3-мысал:

Программа коды	Нәтижесі:
<pre> # Python - C:\Windows\system32\cmd.exe for i in [1, 2, 3, 4, 5]: if i == 3: continue print(i) else: print("For zikli osindaldid!") print("For ziklinin esitligi") </pre>	<pre> ----- RESTART: C:/Users/... 1 2 4 5 for zikli osindaldid! for ziklinin esitligi </pre>

Бұл мысалда бағдарлама 1-ден 5-ке дейінгі сандарды шығарады, ал continue операторы $i = 3$ болғанда элементті өткізіп жібереді.

4-мысал:

Программа коды	Нәтижесі:
<pre> # Python - C:\Windows\system32\cmd.exe for num in range(10, 11): for i in range(2, num): if num % i == 0: print(f'{num} дел. на {i} ') break else: print(f'{num} якшы') </pre>	<pre> ----- RESTART: C:/... 10 ten 2 * 5 </pre>

Бұл код 10-нан 11-ге дейінгі сандар үшін көбейткіштерді табады. Белгіштері жоқ сандарды табатын else блогын қосса, ол «бұл жай сан» хабарламасымен сандарды басып шығарады. range() функциясы 11 санын қамтыймайды.

Else функциясын цикл қалыпты түрде аяқталғандаған пайдалануға болады. Цикл мәжбүрлі түрде токтатылған жағдайда, интерпретатор else операторын елемейді, сондықтан оның орындалуы өткізіліп жіберіледі.

while/else синтаксисінде шарт әрбір итерацияның басында тексеріледі. while операторының ішіндегі код блогы шарт ақиқат болғанша орындалады.

Шарт жалған болғанда және цикл қалыпты орындалса, else сейлемі орындалады. Дегенмен цикл break операторымен мерзімінен бұрын аяқталса, else сейлемі мүлдө орындалмайды.

5-мысал.

Программа коды	нәтижесі
<pre>#include <StdAio.h> #include <Conio.h> #include <iostream.h> #include <Windows.h> int i = 0; while (i < 5) { cout << i; i++; if (i == 3) break; cout << endl; cout << "Element 5-ten kesh emez" }</pre>	<pre>==== RESTART: C:/Users/... 0 1 2 3 4 element 5-ten kesh emez</pre>

while циклі $i < 5$ шарты жалғанша орындалады. else сейлемі шарт жалған болғаннан кейін орындалады. while операторы өрнек ақиқат болған кезде қайта орындау үшін пайдаланылады, ал else операторы while циклімен бірге пайдаланылғанда, шарт жалған болған кезде else операторы орындалады. else операторын елемеу үшін break операторын пайдалану керек.

Программа коды	нәтижесі
<pre>#include <StdAio.h> #include <Conio.h> #include <iostream.h> #include <Windows.h> int i = 0; while (i < 5) { cout << i; i++; if (i == 3) break; cout << endl; cout << "Element 5-ten kesh emez" }</pre>	<pre>>>> 0 1 2</pre>

Бұл кодта break операторы циклді аяқтайды және else сейлемі орындалмайды. Берілген жиында орындалатын break операторы else сейлемдер жинағын орындалмай-ақ циклді тоқтатады. Бұл циклде 5-тен кіші барлық санды үзіліссіз көрсету үшін continue операторы қолданылады.

Бірінші жиында орындалған continue операторы жиынның қалған белгін еткізіп жібереді және өрнекті тексеруге оралады, else операторы келесі элемент болмаса орындалады.

Программа коды	нәтижесі
<pre>#include <StdAio.h> #include <Conio.h> #include <iostream.h> #include <Windows.h> int i = 0; while (i < 5) { cout << i; i++; if (i == 3) continue; cout << endl; cout << "Element 5-ten kesh emez" }</pre>	<pre>==== RESTART: C:/Users/... 0 1 2 3 4 element 5-ten kesh emez</pre>

`continue` операторы `i = 3` шарты ақиқат болғанда жиынның бір бөлігін өткізіп жібереді және басқару элементі `while` цикліне қайта оралады. `else` операторы да орындалады.

`while/else` циклінің денесі бір немесе бірнеше айнымалының мәнін өзгертуі керек, осылайша шарт сонында жалған болып, цикл аяқталады. Әйтпесе цикл әрдайым қайталанады, ол шексіз цикл деп аталады. Әрқашан циклдегі `else` операторының шегінісін оның цикл негізгі сөзімен (`for/while`) дұрыс тураланғанын (шегініспен) тексеру керек. Егер `else` циклінің сөйлемі басқа операторлармен теңестірілсе (шегініспен), код жұмыс істемейді.

10.10. Бір өлшемді массивтер пайдаланып Python программалау тілінде программалар жасау

Қоғамда бізді әртүрлі заттар (нысандар) қоршайды, біз олармен әртүрлі іс-әрекеттер орындаімyz. Сонымен қатар компьютерлік бағдарламалар нысандарды немесе мәліметтерді басқарады. Python-да мәліметтер құрылымының негізгі түрі – тізім. Басқа бағдарламалау тілдерінде мұндай құрылымының түрі – массив. Python-да массивлен жұмыс істеу үшін «`array`» модулі енгізіледі, бірақ көбінесе тізім қолданылады. Тізімдер мен массивтердің айырмашылығы: олардың орындаітын функциялары. Массивте барлық элементтердің типтері бірдей, ал тізімде әртүрлі болуы мүмкін.

Тізім – еркін типтегі элементтер жиынтығы.

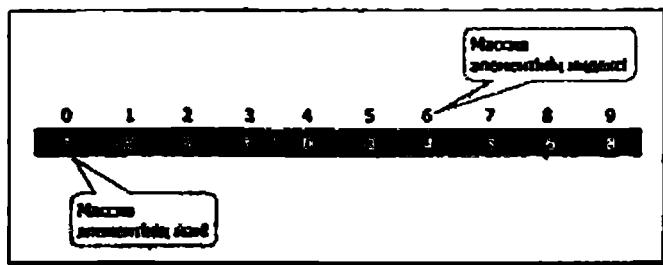
Массив – белгілі бір атпен жазылған бірдей типтегі мәліметтердің реттелген жиынтығы. Массив эр түрлі мәліметтер көлемін қамтуы мүмкін.

Массивтің ұзындығы (өлшемі) – массивтегі элементтер саны.

Индекс – массивтегі элементтің реттік номірі.

Бірөлшемді массив – элементті көрсету үшін бір индексті қажет ететін массив.

Python-да элементтерді номірлеу 0-ден басталады. Элемент индексі тік жақшада көрсетіледі.



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

```
for i in range (0,10):
    testArray.append(i)
```

```
for i in range (0,10):
    Print (testArray[i])
```

Массивтер өлшемінің келесідей түрлері бар:

- бірөлшемді;
- екіөлшемді немесе матрица;
- үшөлшемді.

Массивті (тізімді) колданудың бірнеше әдісін қарастырайық.

1-мысал. Бағдарламада массивті (тізімді) көрсету үшін оның атын көрсетіп, тік жақшага үтірмен бөлінген элементтер тізімін енгізуге болады.

Жолдық элементтердің мәндерін апострофпен жазылады.

```
імде = [45, 78, 3, 9, 23, -12, 67, -15, 7, 54]
неділдегі = ['monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday']
```

2-мысал. Массивті бос түрде сипаттауга болады, ал элементтерін келесідей пернетақтадан енгізуге болады:

```
A=[]
A = [ input() for i in range(5) ]
```

Бұл фрагментте 5 бүтін саннан тұратын массивтің енгізілуі үйымдастырылған. `int` функциясы пайдаланушы енгізген жолды бүтін сандарға айналдырады. `i` цикл параметрі көрсетілетін элементтің индексін анықтайды. `range(5)` функциясы `i` параметрінің мәндерін 0-ден 4-ке дейін орнатуға мүмкіндік береді.

Массивті толтыру үшін тізім генераторын пайдалануға болады. Бұл жағдайда массив белгілі бір ережеге сәйкес толтырылады.

3-мысал. 5 элементтен тұратын `a` массивін ондық сандармен толтыру.

`a = [10]*5`

Нәтижесі: `a = [10, 10, 10, 10, 10]`

4-мысал. 0-ден 9-га дейінгі сандардан `b` массивін күрү.

`b = [i for i in range(10)]`

Нәтижесі: `b = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`

5-мысал. Жолдан массив күрү.

`str= [str *2 for str in 'plan']`

Нәтижесі: `str = ['pp', 'll', 'aa', 'nn']`

6-мысал. Кездейсок сандармен толтыру.

`from random import randint`

```
c = [randint(10,20) for x in range(5)]
```

Нәтижесі: массив 10-нан 20-ға дейінгі 5 кездейсоқ санды береді.

Массивтің жеке элементіне жүгіну үшін оның индексін көрсету керек.

7-мысал. Массив элементтерін өзгертуге болады.

```
Num = [45, 78, 3, 9, 23, -12, 67, -15, 7, 54]
Num[0]=98
Num[7] = - 8
```

Нәтижесі: Num = [98, 78, 3, 9, 23, -12, 67, -15, 7, 54]

8-мысал. Массивтің жеке элементтерін шығару

```
Num = [45, 78, 3, 9, 23, -12, 67, -15, 7, 54]
print(Num[2])
print(Num[5])
```

Нәтижесі: 3 -12

9-мысал. Массивтің барлық элементтерін шығару үшін параметрі бар цикл қолданылады:

```
Weekday = ['monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday']
print('Анта күндері')
for i in range(7): print(Weekday[i])
```

Массивтің ұзындығын len() функциясы арқылы анықтауга болады.
Массивтің соңына жаңа элемент косу үшін append() әдісі қолданылады.

10-мысал. Түстер тізімі берілген. Соңына қызыл түсті қосу керек. Тізімді экранға шығар.

```
color = ['Blue', 'Green', 'Orange', 'Gray']
color.append('Red')
for i in range(len(color)): print(color[i])
```

Нәтижесі:

blue
Green
orange
Gray
Red

11-мысал. Массивтерді «+» таңбасы арқылы қосуға (біріктіруге) болады.

```
m1=['a', 10, 'b']
m2=[2.5, 'number', 65]
m=m1+m2
print (m)
```

Нәтижесі: $m=['a', 10, 'b', 2.5, 'number', 65]$

12-мысал. Массивте элементтерді қайталауға болады.

```
a=[0, 1, 2]*3
print (a)
```

Нәтижесі: $a=[0, 1, 2, 0, 1, 2, 0, 1, 2]$

10.11. Екі өлшемді массивтер пайдаланып Python программалай тілінде программалар жасау

Кейбір есептерді шешу кезінде мәліметті тікбұрышты кесте түрінде ұсыну қажет. Математикада мұндай кестелер матрица деп аталады. Программалай кезінде кестелермен жұмыс істеу үшін екіөлшемді массивтер қолданылады.

Екіөлшемді массив – әрбір элементті орнату үшін екі индексті қажет ететін массив. Екіөлшемді массивтің индекстері элемент орналасқан жол немірі мен баған нөмірін көрсетеді. $a[i][j]$ белгісі і жолы мен j бағанының киылышында тұрған элементті билдіреді.

Сандардың тікбұрышты кестесін қарастырайық.

				j бағаны		
						i жолдар
23	4	8	75			
60	72	45	12			
6	63	94	3			

Бұл кесте 3×4 өлшемдегі екіөлшемді массив түрінде ұсынылуы мүмкін. Мұндай массивте 12 элемент болады.

Массив элементтеріне сәйкес келетін индекстерді қарастырайық. Жолдар мен бағандардың индекстері 0-ден басталып, тік жақшага жазылады.

$a[0][0]$	$a[0][1]$	$a[0][2]$	$a[0][3]$
$a[1][0]$	$a[1][1]$	$a[1][2]$	$a[1][3]$
$a[2][0]$	$a[2][1]$	$a[2][2]$	$a[2][3]$

Бұл массивте $a[0][0]=23$ және $a[1][2]=45$ болады.

Python-да екіөлшемді массивтермен жұмыс істеу үшін кіріктірілген тізімдерді пайдалануға болады. Мысалы, кестедегі массивтің программасын $a = [[23,4,8,75], [60,72,45,12], [6,63,94,3]]$ түрінде жазуға болады.

а тізімі әрқайсысы төрт саннын тұратын және белгілі бір массив мерзіміне сәйкес келетін үш кіріктірілген тізімнен тұрады. Екіншемді массивтің барлық жолымен жұмыс істегендегі кіріктірілген тізімнің индексін көрсету керек.

Екіншемді массивтің барлық элементтерін кіріктірілген циклдер арқылы көрсетуге болады. Сыртқы цикл жол индекстерінің, ішкі цикл баған индекстерінің мәнін көрсетеді.

1-мысал. Екіншемді массив элементтерінің қосындысын табу
Бағдарламалық коды:

```
Э мәннелу - C:/Users/Asus/Desktop/Python/Математикалық операторлар (1).py
Рас 101 Консоль Вый Опции Выход Назар
a=[[23,4,8,75], [60,72,45,12], [6,63,94,3]]
s=0
for i in range(len(a)):
    for j in range(len(a[i])):
        s+=a[i][j]
print(s)
```

Нәтижесі:

375

Ескерту. Массивтегі жолдар саны len(a) функциясы арқылы есептелетін а массивінің ұзындығына тең. Массивтегі баған саны тізімдегі элементтің ұзындығына тең, ал ол len(a[i]) функциясы арқылы табылады.

Екіншемді массивті кесте түрінде экранға шығару үшін кіріктірілген циклдерді қолдану керек.

2-мысал. Екіншемді массив берілген. Массивтің әрбір элементін 2-ге белгеннен кейінгі қалдықтарымен аудыстыру және жаңа массивті экранға шығару керек.

Бағдарламалық коды:

```
Э мәннелу - C:/Users/Asus/Desktop/Python/Математикалық операторлар (1).py
Рас 101 Консоль Вый Опции Выход Назар
a=[[23,4,8,75], [60,72,45,12], [6,63,94,3]]
for i in range(3):
    for j in range(4):
        a[i][j]=a[i][j]//2
        print(a[i][j],end=' ')
print()
```

Ескерту. i параметрі бар сыртқы цикл 0-ден 2-ге дейінгі жолдардың индексін іріктеуге, ал j параметрі бар ішкі цикл 0-ден 3-ке дейінгі бағандардың индексін іріктеуге мүмкіндік береді. Ишкі цикл екі операторды орындаиды. Біріншісі – элементтерді аудыстыру, екіншісі – бос орындарға бөлінген қатар элементтерін көрсету. print() бос операторы сыртқы циклге сілтеме жасайды және бір жол элементтерін көрсеткеннен кейін жаңа жолға өтуге мүмкіндік береді.

Нәтижесі:

100100100101

Массив элементтерін пернетақтадан енгізуге болады.

3-мысал Оқушы төрт алға бойы ауа температурасын жазып отырды. Ол нәтижелерді 4×7 матрицасына жазды. Әр алтадағы максималды температураны көрсету керек.

Бағдарламалық коды:

```
n=4
t = []
for i in range(n):
    row = input().split()
    for i in range(len(row)):
        row[i] = int(row[i])
    t.append(row)

for i in range(n):
    print (max(t[i]))
```

Ескерту. Массив элементтері жолма-жол енгізіледі. Жолды оку үшін `input()` функциясын қолдану керек. `split()` әдісі енгізілген жолды бөлек элементтерге бөледі. Екінші цикл жол элементтерін сандарға айналдырады.

Массив элементтерін белгілі бір ережеге сәйкес жасауға болады.

ТАҚЫРЫПТЫҚ ТЕСТ

1. Сұрыптау кезінде массив қалай өзгереді?

- A) Массивтегі элемент индекстерінің реті өзгереді
- B) Массивтегі элементтердің мәні өзгереді
- C) Массивтегі элементтердің орналасуы өзгереді
- D) Бірдей элементтер жоылады
- E) Массив элементтеріне мән қосылады

2. `fruit=['kivi', 'orange', 'banana', 'apple', 'lime', 'avocado']` массиві берілген.

- A) `fruit=['orange', 'kivi', 'banana', 'apple', 'lime', 'avocado']`
- B) `fruit=['kivi', 'banana', 'apple', 'orange', 'lime', 'avocado']`
- C) `fruit=['lime', 'avocado', 'orange', 'kivi', 'banana', 'apple']`
- D) `fruit=['apple', 'banana', 'lime', 'avocado', 'orange', 'kivi']`
- E) `fruit=['apple', 'avocado', 'banana', 'kivi', 'lime', 'orange']`

3. `B=[-6,3,7,-8,1,2]` массиві берілген. Массивті кему ретімен сұрыпта.

- A) `B=[-6,3,7,-8,1,2]`
- B) `B=[7,3,2,1, -6,-8]`
- C) `B=[-8,-6, 1,2,3,7]`
- D) `B=[1,2,3,7,-8,-6]`
- E) `B=[1,2,3,7, -6,-8]`

4. Программада берілген массив:

`city=['Rim', 'Parizh', 'Moscow', 'Nur-Sultan', 'Tashkent'].`

Программаны орындау нәтижесінде шыққан массив:

`city=['Tashkent', 'Rim', 'Parizh', 'Nur-Sultan', 'Moscow'].`

Массивті орындау үшін қандай команда орындалды?

- A) `city.sort()`
- B) `city.sort(reverse=True)`
- C) `sorted(city)`
- D) `city.sort(reverse)`
- E) `city.sort(reverse=False)`

5. Python-да тек оқуға арналған файл дескрипторын көрсет.

- A) t
- B) w
- C) r
- D) a
- E) b

6. Бұл типтің файлдарында жол шектеулері жок, деректер машина үшін түсінкіті екілік тілге түрлендіруден кейін сақталады. Не жайлыштыралады?

- A) Графикалық файлдар
- B) Дыбыстық файлдар
- C) Екілік файлдар
- D) Мәтіндік файлдар
- E) Командалық файлдар

7. Файлдағы алдыңғы және жаңа деректерді сактау үшін қай режимді пайдалануға болады?

- A) Оқу режимі
- B) Түзету режимі
- C) Жазу режимі
- D) Қосу режимі
- E) Файлды ашу

8. Код жолы берілген: `file.seek(0)`. Код файлдың бөлігіне шығады?

- A) Файлдың ортасы
- B) Файлдың басталуы
- C) Файлдың ортасы
- D) Файлдың соны
- E) Файлдың жалғасы

9. Python программалашу тілінде «тән емес» логикалық шартын қолдану мысалын көрсет.

- A) `a<=b`
- B) `a<b`
- C) `a!=b`
- D) `a==b`
- E) `a>=b`

10. Python программалау тілінде «кем немесе тең» логикалық шартын қолдану мысалын көрсөт.

- A) $a \leq b$
- B) $a < b$
- C) $a \neq b$
- D) $a == b$
- E) $a \geq b$

11. Python программалау тілінде «картық немесе тең» логикалық шартын қолдану мысалын көрсөт.

- A) $a \leq b$
- B) $a < b$
- C) $a \neq b$
- D) $a == b$
- E) $a \geq b$

12. Бірінші итерацияны бастамас бұрын while циклінің шарты Жалған болса, циклге не болатынын анықта.

- A) Цикл басталмайды
- B) Нәтиже формула бойынша есептелетін болады
- C) Шексіз цикл
- D) Нәтижесі 0 болады
- E) Цикл басталады

13. Циклдің бір орындалуын атайтын дұрыс термин қандай?

- A) Есептегіш
- B) Бастау
- C) Функция
- D) Итерация
- E) Қайталау

14. Итерация неліктен маңызды?

А) Бұл қайталанатын кадамдарды жою арқылы кодты жеңілдетуге мүмкіндік береді

- B) Бұл программада бірнеше жолды қолдануга мүмкіндік береді
- C) Ол нұсқаулардың ретін анықтайды
- D) Бұл кодтың дұрыс жұмыс істеуін қамтамасыз етеді
- E) Бұл қайталау санын көбейтеді

15. Итератордан барлық мәнді бірден алу қажет болса, қандай кіріктірілген функцияны пайдалану керек?

- A) Next()
- B) List()
- C) Iter()
- D) Stopiteration()
- E) set(itr)

16. Берілген мәлшерде қайталау үшін қандай функция қолданылады?

- A) For

- B) If
 C) While
 D) Else
 E) Break

17. Массив ұзындығының анықтамасын таңда.

- A) Бұл массивтің ең үлкен элементі
 B) Бұл массивтің соңғы элементтің нөмірі
 C) Бұл массивтегі элементтер саны
 D) Бұл массив элементтің көрсету үшін қажет индекстер саны
 E) Бұл массивтің ең кіші элементі

18. Элементтерді көрсету үшін бір индексті қажет ететін массив қалай аталады?

- A) Көпелшемді
 B) Екіөлшемді
 C) Бірөлшемді
 D) Тізім
 E) Жол

19. $b=[2,3,8,12,9,1]$ массиві берілген, $b[4]$ элементтің мәнін таңда.

- A) 2
 B) 3
 C) 8
 D) 12
 E) 9

20. Пернетақтадан 10 бүтін сандар массивін енгізу үшін дұрыс жолды таңда.

- A) $k=[(input() \text{ for } i \text{ in range}(10))]$
 B) $k=[\text{int}(input()) \text{ for } i \text{ in range}(10)]$
 C) $k=[\text{int}(input(10))]$
 D) $k=[\text{for } i \text{ in range}(10)]$
 E) $k=[(input(10))]$

21. Индексті 4 болатын жол мен индексті 2 болатын бағанның қиылысында орналасқан массив элементі қалай жазылады?

- A) $a[2][4]$
 B) $a(4)(2)$
 C) $a[4][2]$
 D) $a(2)(4)$
 E) $a(4x2)$

22. Әр элементті орнату үшін екі индексті қажет ететін массив түрін анықта:

- A) Көпелшемді
 B) Екіөлшемді
 C) Бірөлшемді
 D) Жолдық

E) Бағандық

23. 6x9 өлшемді массив берілген. Массив қанша жолдан тұрады?

- A) 9
- B) 54
- C) 6
- D) 10
- E) 0

24. Екі өлшемді массив берілген:

45	5	12
7	13	75
52	9	2

A[2][1] элементінің мәнін тап.

- A) 45
- B) 7
- C) 13
- D) 75
- E) 9

25. Массивтің барлық жолымен жұмыс істеуге мүмкіндік беретін индексті белгіле . Программада $f=[45,12,3],[4,7,52],[9,1,65]$ массиві берілген.

- A) f(i)
- B) f
- C) f[]
- D) f[i]
- E) f()

26. split() әдісінің мақсатын белгіле.

- A) Енгізілген жолды бөлек элементтерге бөледі
- B) Екіөлшемді массивтің барлық элементін оқиды
- C) Жол элементтерін сандарға айналдырады
- D) Массивтің әрбір элементін оқиды
- E) Массив элементтерін біркітреді

27. Python тіліндегі бағдарлама үзіндісінің орындау нәтижесіндегі с массивінің элементтерінің мәндерін анықта.

a=[23,45,7,9,1]

b=[0,5,87,16,3]

c=a+b

A) c=[23,0,45,5,7,87,9,16,1,3]

B) c=['23+0','45+5','7+87','9+16','1+3']

C) c=[23,50,94,25,4]

D) с массивінің элементтерін есептеу мүмкін емес

E) c=[23, 45, 7, 9, 1, 0, 5, 87, 16, 3]

28. Программа фрагментінің нәтижесі

```
k=6
for a in range(1,k):
    print(a*5)
```

- A) 4 6 8 10
- B) 3 5 7 9 11
- C) 4 8 12 16 20
- D) 5 10 15 20 25
- E) 1 2 3 4 5

29. Программа фрагментінің нәтижесі

```
k=6
for a in range(1,k):
    print(a**2)
```

- A) 1 4 9 16 25
- B) 3 5 7 9 11
- C) 4 8 12 16 20
- D) 5 10 15 20 25
- E) 1 2 3 4 5

30. Программа фрагментінің нәтижесі

```
k=6
for a in range(1,k):
    print(a*1)
```

- A) 1 3 5 7 9
- B) 1 2 3 4 5 6
- C) 3 6 9 12
- D) 1 2 3 4 5
- E) 1 3 5 7

Пайдаланылған әдебиеттер тізімі

1. Язык программирование Python. Сузи Р.А. Учебное пособие. –М.: Интернет Университет информационных технологий, 2007 г.
2. Марк Лутц. Программирование на Python. 2011 г.
3. Доусон М. Программируем на Python. Питер, 2014 г.
4. <http://pythonworld.ru/>.
5. Д.Н. Исабаева, Г.А. Абдукалкаримова, Л.Б. Рахимжанова, М.А. Әубекова Информатика. 11-сынып
6. Қадырқұлов Р.А., Рысқұлбекова Ә.Д., Нұрмұханбетова Г.К. 2021 ж. Информатика. 7-сынып, 2021 ж.
7. Салғараева Г.И., Калымова К.А., Орынтаева Ж.А. Информатика. 9-сынып, 2019 ж.
8. Д.Н. Исабаева, Е. Киселева, Н. Құрманғалиева, Л.Б. Рахимжанова, М.А. Әубекова. Информатика. 10 сынып
9. Салғараева Г.И., Бекжанова А.А., Базаева Ж.Б. Информатика. 8-сынып, 2019 ж.
10. Саммерфилд М. Программируем на Python. Перевод с англ. Киселев А., 2009 г.

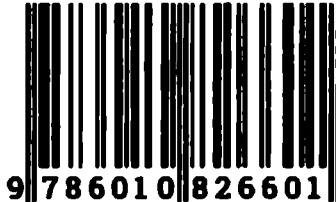
**ИНФОРМАТИКА
ҰБТ
II бөлім**

ОҚУ-ӘДІСТЕМЕЛІК ҚҰРАЛ

Оқу құралы «QAZBILIM» редакциясында қаралды

Директоры: **Гүлбану ҚАНАЙ**
Кұрастырушылар: **Гүлмира ЖАСҰЛАНОВА**
Корректоры: **Алтыншаш ӘБҮТӘЛІП**
Көркемдеуші: **Тоқтар ЖЕТПІСБАЙ**

ISBN 978-601-08-2660-1



*Баспаға 01.08.2022 ж. берілді. Формат В5.
Ұсыныс, тікірлеріңізді kazbilim@gmail.com электронды
мекенжайына жіберулеріңізді сұраймыз.
Таралымы 5000 дана.*