PYTHON БАҒДАРЛАМАЛАУ ТІЛІН НЕГІЗДЕРІН» ОҚУ МЕН ОҚЫТУ ПРОЦЕСІНДЕ ОҚУШЫЛАРДЫҢ АБСТРАКТІЛІ ОЙЛАУІ ДАҒДЫЛАРЫН ҚАЛЫПТАСТЫРУ ЖӘНЕ ДАМЫТУ

Түркістан облысы
Сайрам ауданы
№56 Ю.Гагарин атындағы ЖОМ
Математика және информатика пәні мұғалімі
Ахмедов Закиржон Октамович
ahmedow@mail.ru

Аннотация:

Мақалада мектеп жасындағы балаларға білім беру жүйесінде «Python бағдарламалау тілін негіздері» бөлімдерін оқу мен оқыту процесінде оқушылардың абстарктілі ойлау дағдыларын қалыптастыру мен дамыту мәселесі қарастырылады. Мәселенің өзектілігі оқушылардің Python бағдарламалау тілінде бағдарламаларды әзірлеуге кәсіби көзқарасын қалыптастыру қажеттілігімен байланысты. Абстрактілі ойлаудың даму деңгейін бағалау әдісі сипатталған, оның мәні Л.С. Выготскийдің (синкреттер, комплекстер, нақты ұғымдар) әдісі бойынша, бағдарламаларды жазу кезінде оқушылардың Python бағдарламалау тілінде деректердің типтеріне байланысты жіберген қателіктері мен олардың абстарктілі ойлауының даму кезеңдері арасындағы байланысты анықтау болып табылады. Бұл әдістің мүмкіндіктері оқушының бағдарлама жазу процесінде жүйелік ойлауының қажетті шарты болып табылатын «абстракті деректер типі» ұғымының даму шарттары мысалдармен көрсетіледі.

Мектеп жасындағы балалардың «Python бағдарламалау тілін негіздері» бөлімдерін оқыту барысында әзірленген абстрактілі ойлауды дамыту бойынша әдістемелік ұсыныстар беріледі.

Тірек сөздер:

Информатиканы оқыту әдістемесі, бағдарламалауды оқыту, оқытуға ынталандыру, абстарктілі ойлауды дамыту.

Кіріспе

Бүгінгі таңда кез-келген оқулық мұғалім үшін оқытуда қосымша ресурс болып табылады. Ал, оқушы оқуындағы бірінші ресурс бұл оқулық болуы керек деп ойлаймын. Себебі, мұғалімнің оқушыға өз бетінше орындауға берген тапсырмаларды орындау үшін бірінші болып оқулыққа жүгінеді.

Кейде кейбір оқулықтарда техникалық қателіктердің болуы, пәндік ұғымдер мен түсініккетерге байланысты берілген өзгеше талқылаулар, кейде оқуға ұсынылған оқу материалының көлемінің үлкен болуы себепті оқушылар тапсырма орындау кезінде оқулықтан тиімді пайдалана алмайды.

Сондықтан информатика пәні бойынша әріптестеріме оқулықтарда кездесетін жоғарыда атап өтілген кемшіліктердің оқушының оқуына кедергі келтірмей айналып өту жолдарын ұсыну мақсатында тәжірибеде сынылған әдісті осы мақаланы жазу арқылы жеткізуді жөн деп таптым.

Абстракция ойлаудың бір түрі ретінде:

Абстрактілі ойлау дегеніміз - адамның нақты объектілер туралы ақпаратты таңбалық түрде таңбаларға айналдырып, оны нақты практикалық мәселелердің шешімін табуда қолдану мүмкіндігі. Ол ғалымдар - физиктер, математиктер, информатиктер және зертте саласының мамандары. Жинақталған белгілер мен белгілерді сәтті қолданып, адам көптеген табысты операцияларды үйрене алады. Абстрактілі ойлау не болып жатқанын әр қырынан көруге, оқиғалардың нәтижесін имитациялауға және дерексіз қорытынды жасауға мүмкіндік береді. Белгілі бір дәрежеде ол әр адамның бойында дамиды, дегенмен күшті ойлау аппаратын дамыту үшін уақытты, ресурстарды едәуір инвестициялау қажет, сонымен қатар

өз саласына деген қатты құштарлық қажет. Абстракция - бұл белгілі бір құбылыстардың қасиеттерін жалпылаудың бір түрі, оның негізінде адам психикалық тұрғыдан ұқсас суретті «сала» алады және өз бетінше заттардың мінез-құлық моделін ойлап табады.[1]

Мектепте информатика сабақтарында оқушыларға «Python» бағдарламалау тілінде бағдарламалауды оқытудың маңызды міндеттерінің бірі-оқушылардың бағдарламаларды әзірлеуге кәсіби көзқарасын біртіндеп қалыптастыру болып табылады. Бұл міндетті орындау абстрактілі ойлауды дамытпай тұрып мүмкін емес, абстрактілі ойлау-бұл тек тапсырманы дұрыс қоюға, бағдарламалау тәсілін таңдауға ғана емес, бәлкі бағдарламашының назарын ең алдымен мәселені формалді шешуге аудару. Бұл жағдай өз кезінде таңдалған тәсілнің контекстінде белгілі бір абстракциялық жүйе арқылы ойлау процесінде қалыптасады, (деректер типі, модульдік, басқару) және есептеу процесінің модельдерін қарастыруға жетелейді.

Бағдарламалауды оқыту процесінде абстрактілі ойлауды қалыптастыру және дамытудың маңыздылығына қарамастан, бұл мәселе мектепте информатика пәнін оқыту, соның ішінде бағдарламалауды оқыту мен зерттеу әдістемесіне арналған бірқатар басылымдарда ішінара ғана қозғалады.

Мысалы:

Көптеген абстрактілі ойлауды қалыптастыруға бағытталған жұмыстарда, соның ішінде Д.М.Гребневаның «Семиотический подход к обучению программированию в школе» атты мақаласында бағдарламалауды үйренуге семиотикалық тәсіл деп аталатын контексте абстрактілік дәрежесінің жоғарылау принципі туралы айтылады.[2] Бұл принципке сәйкес, есептерді бағдарламалау кезінде, оқушылар біртіндеп абстрактілі белгілерге, символдық жүйелерге және ресми сипаттамаларға көшуі керек. Алайда бұл тәсіл абстрактілі ойлаудың даму кезеңдерінің психологиялық-педагогикалық негіздерін ескере отырып, информатиканы оқытуда бағдарламалау сабақтарын ұйымдастырудың және өткізудің нақты тәжірибесіне қатысты анық қорытынды жасауға мүмкіншілік бермейді.

Келешектегі бағдарламашының, яғни оқушының ресми ойлау стилін дамыту керек, бұл дегеніміз: оқушының назарын мәселені кодтау процесіне емес, мәселені шешу процесіне аудару керек деп ойлаймын. Осылайша, бағдарламалау тілінің синтаксисі мен семантикасын білу негізгі міндет емес, бұл оқытудың қосалқы міндеті болып табылады. Білімді игерудің негізгі шарты есептерді шешудің әртүрлі кезеңдерінде рефлексия мен сыни ойлау механизмдерін қолдану нәтижесінде пайда болатын «когнитивті-технологиялық бірлік» деп жариялануы керек. Мұғалімге келетін болсақ, оған оқушылардың жазған кодының сапасын және олардың ойлау стилін қадағалау және басшылыққа алу міндеті жүктеледі, өйткені «бағдарламалау бұл өнер, яғни ойлау өнері».

Әрине, әзірленетін бағдарламалардың сапасы мен ойлау нәтижесінің сапасы арасында байланыс бар екенін түсіну өте маңызды. Сыни ойлау мен рефлексия қажет, бірақ проблемаларды шешуге нақты, анық көзқарасты қалыптастырудың жеткіліксіз шарттары екенін түсіну де өте маңызды. Нақтысын айтқанда, балалардың сыни ойлау мен рефлексия дағдыларын бастауыш сынып оқушыларының бойында да байқауға болады. Алгоритмдік ойлау дағдыларын қалыптастырып дамытуға бағытталған көптеген әдістемелік құрал және бағдарламалық жабдықтар (LOGO, Scretch және т.б) бар. Алайда, бұл қосымшалар тек ғана алгоритмдік ойлау басында тұрған, өз ойын тек ғана көрнекілік құралдарға жүгіну арқылы білдіріп, қорытынды жасай алатын 7-11 жастағы оқушыларға арналған. Осы жастағы балалардың қызығушылығы мен бағдарламалауға деген талпынысына қарамастан, көрнекі алгоритмдер есептерді шешуге анық көзқарасты қалыптастыруға ықпал етеді деп айтуға болмайды. Бұл дағды оқушыларда тапсырманы жалпылау мен егжей-тегжейдің әртүрлі деңгейлерінде еркін қарастыруды, сондай-ақ жүйелі ойлау мен жалпылауды үйренгенде ғана пайда болуы мүмкін.

Сонымен, бүгінгі таңда абстрактілі ойлауды дамыту проблемаларын қандай да бір жолмен қозғап тұрған әдістемелік жұмыстарға мыналар тән деп айтуға болады:

- абстрактілі ойлауды дамытудың қажеттілігі айтылады және негізделеді, бірақ бұл дағдыны дамытуды ұйымдастыруға болатын теориялық модель жоқ;

-мұғалімдерге бағдарламалау негіздерін оқытуда тәжірибесінде тікелей қолдануға болатын абстрактілі ойлаудың даму деңгейін бағалаудың сыналған тәсілдері жоқ;
- эмпиристикалық жолмен алынған практикалық тәжірибе нәтижелері рефлексия мен сыни ойлаудың белгілі бір деңгейде дағдысы қалыптапаған оқушылардың бойында абстрактілі ойлауды қалай дамыту керек деген сұраққа жауап бермейді.

Абстрактілі ойлауды қалыптастырып, дамытудың әдіс-тәсілдерін әзірлеу үшін осы жұмыста Л.С.Выготскийдің теориялық моделін қолдануды ұсынамын. Бұл таңдауды қазіргі уақытта педагогикалық практикада кеңінен қолданылатын жақын арадағы даму аймағы (ЖАДА) тұжырымдамасының құрамдас бөліктерінің бірі болып табылатындығымен негіздеуге болады.

Л.С.Выготский «түсінікті» символдық және семантикалық (мағыналық) жағының бірге жүреді деп ұсынды, ал түсініктің дамуы оны пайдалану процесінде түсініктің мағынасын өзгертуді білдірді деп айтады [3]. Абстрактілі ұғымдар дамудың белгілі бір кезеңінде пайда болады және «символдық – семантикалық», яғни «белгі - мағына» қатынасын түрлендірудің ұзақ тізбегінің нәтижесі болып табылады. Л.С.Выготскийның айтуы бойынша символдық және семантикалық «түсінік» екі жолмен даумуы мүмкін: а) өздігінен («күнделікті жағдай»), б) оқу мен оқыту нәтижесінде ("ғылыми ұғымдар").

Бағдарламалау негіздерін оқыту тәжірибесінде мұғалімдер кейде күнделіктігі қолданылып жүрген механизмдер ұғымы қателіктерге алып келу себептеріне тап боламыз. Оқушылар есепті бағдарламалу кезінде жинақталған білімі, түсінігі және тәжірибесіне жүгінеді. Ал есепті шешу жолдарын қажетті сыни қайта қарастыруға мән бермейді, тікелей білімі мен түсінігін пайдалануға тырысады. Бұл жағдайда Л.С.Выготскийның теориясындағы ойлаудың дамуының алғашқы екі кезеңін байқауға болады, атап айтқанда а) синкреттік ойлау, б) комплекстік ойлау.

Синкреттік ойлау кезеңі

Ойлау дамуының синкреттік ойлау кезеңі ұғымдардың белгісі мен мағынасы арасындағы байланыстардың субъективті орнатылуымен сипатталады. Бұл жағдайда субъективтілік оқушының ең алдымен өз алған білімі мен тәжірибесіне жүгінуі және оны сыни тұрғыдан қайта қарастырмай, мәселелерді шешу үшін пайдалануға тырысуы болады.

Мысалы, қай уақытта оқушының «айнымалы» туралы білімі синкрет кезеңде? Бұл жағдай есептің бағдарламасын жазу барысында пайдаланылатын айнымалылардың типі басқа, нәтижеге шығатын айнымалының типі басқа болатынына оқушылар мән бермейді. Негізінен оқушылардың айнымалы туралы білімі бар, өйткені олар алгебра пәнінен айнымалыларды пайдаланып есептер шығарған. Алайда алгебра пәнін оқу барысында айнымалының типі туралы айтылмайды. Соның нәтижесінде:

sum = x + y + z жазу орнына ***x + y + z*** жазуды байқау мүмкін.

Немесе, алгебра пәнінен айнымалының орны өзгерген мен өрнектің мәні өзгермейді деген түсінік оқушыда бар. Сол тәжірибеге жүгініп төмендегі жазуды байқау мүмкін: ***random.randint(1,10) = x*** қате жазудің орнына ***x=random.randint(1,10)*** жазу дұрыс болып табылады.

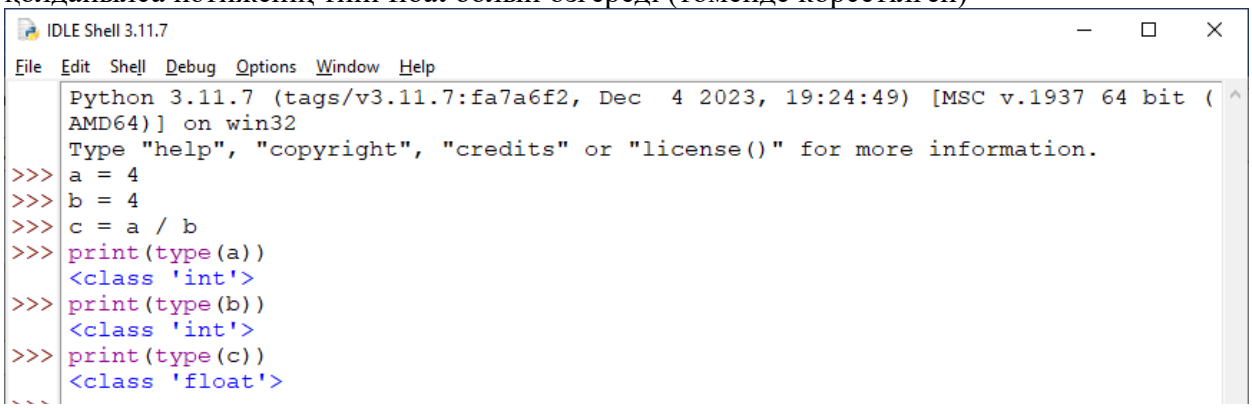
Комплексті ойлау кезеңі

Ойлаудың одан әрі дамуы бастапқы білім мен тәжірибені сыни тұрғыдан қайта қарауға негізделуі мүмкін және сондай болуы керек. Бұл қайта ойлау, ең алдымен, практикалық іс-әрекет процесінде бір немесе бірнеше объективті белгілерді бөліп алумен байланысты, олардың негізінде синкреттік сатыдағы субъективті белгілерден өзгеше жаңаша ойлау қалыптасады.

Мысалы, оқушылар алғашқы жазған бағдарламаларында айнымалыларды қолдана отырып, біртіндеп айнымалыға мәндерді меншіктеуге болатындығын түсіне бастайды, алайда айнымалының мәнін сақтау үшін бөлінген жад көлемінің шектеулілігі сияқты маңызды сипаттама назардан тыс қалады. Бұл жағдай кейде түсініксіз, бақыланбайтын және оқушылар білмейтін бірқатар қателіктерге алып келеді. Мысалы, екі өте үлкен бүтін (int) санды қосу кезінде нәтиже теріс таңбалы сан болып шығуы мүмкін, себебі айнымалының типі үшін жадынан бөлінген келемнің жетіспеу.

«Айнымалы» туралы комплексті ойлау кезеңінде «айнымалы» туралы түсінік жоғала бастайды. Оқушылар әр түрлі типті айнымалылармен арифметикалық амалдар орындауға тырысады. Str типіне list типін қосуға немесе кейде бөлуге де ұмтылып көреді. Себебі сандық типтермен (int, float) орындау амалдары басқа типтерге де тән болады деп ойлайды. Көбінесе өрнек арқылы берілген есептеу нәтижесінің типі ретінде анықталатын оның «түрі» қолданылған «өрнек» ұғымының белгісі болып табылатыны оқушылардың ойына келмейді.

Мысалы, берілген a,b бүтун сандарымен қосу, азайту немесе көбейту амалдарын оқушылар орындайтын болса нәтиженің типіде бүтін яғни int болады. Алайда бөлу амалы қолданылса нәтиженің типі float болып өзгереді (төменде көрсетілген)



Бағдарлама жазу кезінде оқушылардың структуралы айнымалыларды пайдаланады. Алған білімдері жеткіліксіз болып толтыруды талап ететін комплексті ойлауды қажеттілігі байқала бастайды. Бұны былай түсіндіруге болады. Кейбір оқушылар жүйелік блокті процессор деп атайды. Ал бағдарлама жазуды үйрену процесінде оқушылар массив, массив элементі, элемент индексін егжей-тегжей түсіне бастайды. Ал массив мен тізімнің айырмашылығын мүлдем түсінбейді. Бұл мәселеге нақтырақ тоқталайық **Массив** -бір атаумен біріктірілген, бір типтегі деректерден тұратын айнымалылардың жиынтығы.[4]
Python-да тізім -бір атаумен біріктірілген ерікті типтегі деректердің реттелетін, өзгертілетін жиынтығы.

Жалпы білім беретін мектептің 9-сыныбына арналған оқулықтың 107 бет § 22 массив элементтерін кіріктіру және жою практикум сабағында Python программалау тілінде массив элементтерін жою және массивтерге жаңа элементтер қою үшін пайдалануға болатын функциялар бар деп **append(), extend(), insert(), del(), remove(), pop()** функцияларға теориялық нақты түсінік берілген.[4] Алайда, атап өтілгендер Python бағдарламалау тілінің стандартты функцияларына жатпайды. Олар объектіге қолданылатын әдіс болып табылады.

Python бағдарламалау тіліндегі функциялар мен әдістерді ажырату өте маңызды, өйткені олар әртүрлі мақсаттарда қолданылады. Функциялар кодтың кез келген жерінен шақырылуы мүмкін және жалпы тапсырмаларды орындауға арналған, ал әдістер объектілермен байланысты және белгілі бір объект типіне тән амалдарды орындауға арналған.

Оқушы қолданылатын әдістер туралы териялық білімді толығымен игерген жағдайда да бұл практикалық жұмысты өз бетінше орындай алмайды. Кейде мұғалім көмегімен де орындай алмауы мүмкін. Себебі оқулықта берілген бағдарлама кодында жарияланған **mass = arr.array('i')** массивіне элементер енгізіп, экранға шығару үшін **def printArray(a):** пайдаланушы функциясы қолданылған. Ал, оқушы «Python бағдарламалау тіліндегі пайдаланушы функциялар» туралы білім мен түсінікті 10 сынып жаратылыстану-математикалық бағытында 2-тоқсанда оқыған кезде алады.[5]

Бұл жағдайда оқушының абстарктілік ойлау кезеңлері жүрмейді. Себебі оқушы пайдаланылған ұғымдарды толығымен түсінбейді, талап етілген нәтижеге қол жеткізе алмағандықтан оқушы бойнында эмоционалдық-психологиялық алаңдаушылық байқалуы мүмкін. Оқулықтағы берілген кодтың синтаксис қателіктері жөнделген кезде де оқулықтағы көрсетілген нәтижені бағдарлама бермейді. Себебі, **def printArray(a):** пайдаланушы функциясында жазылған

print("Initial array: ", end=" ") жазуында print функциясына аргумент берілмеген.

print("Initial array: ", mass, end=" ")-деп жазылуы керек.

Сонымен оқушыны синкреттік және комлекстік ойлауға бағыттау үшін берілген практикалық тапсырманы құрамдас бөліктерге бөліп кодты төмендегідей жазуды ұсынуға болады.

1.py - C:\Users\user\AppData\Local\Programs\Python\Python311\1.py (3.11.7)

File Edit Format Run Options Window Help

```
import array as arr #модульді импорттау

n = int(input("Number of elements: ")) #массив элементтер санын енгізу
mass = arr.array('i') #массивті жариялау
#массив элементтерін клавиатурадан енгізу
for i in range(n):
    print(i+1, end='-i')
    mass.append(int(input('element: ')))
print('Initial array: ', mass, end=' ') #массивті экранға шығару

m = int(input("Enter the number of element for delete: ")) #массивтен жойылатын элемент
del mass[m-1] #элементті массивтен жою
print(mass, end=' ')
print('\n')
mass.insert(0,1) #элементті массивтің басына кірістіру
print(mass, end=' ')
print('\n')
mass.append(9) #элементті массивтің соңына кірістіру
print (mass, end=' ')
print('\n')
print("extend([5, 6, 7]) appends iterable to the end of the array")
mass.extend([5, 6, 7]) #элементтерді массив соңына кірістіру 5, 6, 7
print ('Extended array: ', mass, end=' ')
print('\n')
print('Remove 3d element')
mass.pop(2) #3-ші элементті өшіру
print (mass, end=' ')
print('\n')
print('delete an element equal to 7 in the array')
mass.remove(7) #массивтегі 7-ге тең элементті жою
print('Final : ', mass, end=" ")
```

Сонда практикалық жұмыс нәтижесі төмендегідей болады:

```
Number of elements: 5
1-ielement: 1
2-ielement: 2
3-ielement: 3
4-ielement: 4
5-ielement: 5
Initial array:  array('i', [1, 2, 3, 4, 5]) Enter the num
ber of element for delete: 2
array('i', [1, 3, 4, 5])

array('i', [1, 1, 3, 4, 5])

array('i', [1, 1, 3, 4, 5, 9])

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

Remove 3d element
array('i', [1, 1, 4, 5, 9, 5, 6, 7])

delete an element equal to 7 in the array
Final :  array('i', [1, 1, 4, 5, 9, 5, 6])
```

Қорытынды:

Бағдарламалауға байланысты абстрактілі ойлауды дамуту заңдылықтарын талдай отырып, бұл процестің алғашқы шарты оқушылардың есептер тізбегін шешудің практикалық тәжірибесін үнемі жинақтап отыруы болып табылады, оның барысында олар кейде қиындықтарға тап болады және оларды жеңу үшін жаңа тәсілді қолдануға немесе бұрын қалыптасқан бірқатар ұғымдарды жалпылауға және қолдануға мәжбүр болады. Бұл ретте ұсынылып отырылған тәсілмен оқушыларға бағдарлауға берілген оқулықдағы кейбір тапсырмаларды қайта құрастыру арқылы ғылыми ұғымдардың барынша жылдам білу және түсіну нәтижесінен синкрет және комплекстік ойлау сатыларын қысқартуға жол ашылады.

Абстрактілі ойлауды дамытудың жоғарыда аталған принциптері, бір қарағанда, айқын көрінеді, бірақ оны жүзеге асыру кезінде бірқатар себептерге байланысты көптеген проблемалар туындатады.

Алғашқы және маңызды қиындықтар бағдарламалауды оқу курсының басында пайда болады, мұнда оқушыларға өте күрделі тапсырмалар беруге болмайды, бұл, біріншіден, оларға үлкен қызығушылық тудырады деп айтуға болмайды, екіншіден, олар бағдарламалау тілі синтаксисі және семантикасы тұрғысынан емес, бар білетін және түсінетін ұғымдардың көмегімен шешілетін жобаларды алдын-ала формалды қарастыра алама? Деген мәселені мұғалім ескера алуы керек.

Үлкен көлемде код жазуды талап ететін тапсырмаларды құрамдас бөліктерге боліп тапсырмалардың күрделігін азайту тәсілі бағдарламалауды үйрену тұрғысынан да, абстрактілі ойлауды дамыту тұрғысынан да бірден бірнеше артықшылықтар береді. Біріншіден, бұл оқушыларға жаңа бағдарламалау құралын үйрену әрқашан оның жаңа функционалдылықтың пайда болу қажеттілігімен байланысты екенін түсінуге мүмкіндік береді, бұл қиындықты жеңуге мүмкіндік береді. Екіншіден, күрделі мәселені ойдағыдай шешу білу оқушылардың оқуға деген ынтасын арттырады. Сонымен, үшіншіден, жаңа тәсіл абстрактілі ойлауды қалыптастырып дамыту жолында біртіндеп алға жылжып, білім және дағды пирамидасын «аяқтауға» мүмкіндік береді.

Қолданылған әдебиеттер:

1. https://kk.psychicslog.com/10609322-what-is-abstract-thinking
2. **Гребнева Д.М**. Семиотический подход к обучению программированию в школе [Электрондық ресурс] // URL: https://science-pedagogy.ru/ru/article/view?id=1495.
3. **Выготский Л.С**. Лекции по психологии. Мышление и речь. М., 2019. 432 бет.
4. **Информатика:** Жалпы білім беретін мектептің 9-сыныбына арналған оқулық. / Г.И.Салғараева және т.б Нұр-Сұлтан: «Ар ман-ПВ» баспасы, 2019. 86-108бет.
5. **Информатика:** Жалпы білім беретін мектептің 9-сыныбына арналған оқулық. / Г.И.Салғараева және т.б Нұр-Сұлтан: «Ар ман-ПВ» баспасы, 2019. 66-79 бет.