

Дәріс 2

Python тілінің негізгі сипаттамасы. Сызықты алгоритм есептері

Жоспар:

1. Алгоритм ұғымы және оның негізгі қасиеттері.
2. Алгоритмді жазу құралдары мен әдістері.
3. Есептеу процестерінің негізгі типтері: сызықты, тармақталған және циклді.
4. Есепті дайындау және компьютерде шығару кезеңдері

Сабақ мақсаты: студенттерге алгоритм ұғымы, жазу құралдар мен әдістері, негізгі типтері туралы түсінік беру

Оқытудың әдістемесі мен формасы: Баяндау, дәріс

Негізгі түсініктер: алгоритм; блок сызба.

Әдебиеттер: [1,3]

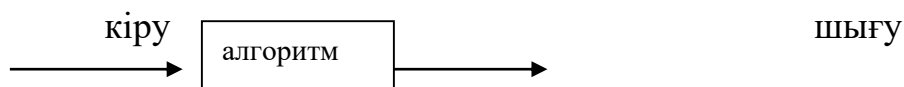
1. Алгоритм ұғымы

Әлбетте, мектепте «Информатика» курсын оқу кезінде алынған білім «Алгоритм деген не?» сұрағына жауап береді. Алгоритм түсінігі қазіргі математикадағы негізгі түсініктің бірі болып табылады. Бұл сөз IX ғасырдағы өзбек математигі Мухаммед әл Хорезмидің атынан шыққан. (арабша – «Аль Хорезми»). Оның арифметика және алгебра бойынша жұмыстары XII ғасырда латын тіліне аударылған және Еуропадағы математиканы дамытуға көп әсер етті. Онымен құрылған төрт арифметикалық әрекетті орындау ережесі «алгоризм» атын алды, соңынан «алгоритмус» сөзіне, содан соң «алгорифм» және «алгоритм» сөзіне айналған. Бірақ-та, одан бұрын Евклид алғашқы алгоритм болып саналатын екі бүтін санның ең үлкен ортақ бөлгішін табу ережесін ашқан.

ДК пайда болған кейін өте үлкен өлшемді алгоритмдерді құру мүмкіндігі туды. Мұндай жағдайда мына тәсіл қолданылады: алдымен алгоритмді қарапайым әрекеттер көмегімен емес, үлкен блок түрінде келтіріп жазады, содан соң әрбір блокты біртіндеп қарапайым әрекетке дейін кішірек түрге келтіріп ашып жазады. Мұндай әдіс кадамдық бөлшектеме әдісі деп аталады.

Программаның блокты құрылымы-ол да құрылымды программалаудың принципі.

Тағы бір көзқараспен алгоритм қарастырайық. Егер оның құрылымына терең қарамасақ, онда кезкелген алгоритмді «қара жәшік» түрінде көрсетуге болады.



Кіру – есептеу үшін алгоритмде қолданылатын айнымалылар мен оның мәндерінің жиынтығы.

Шығу – есептеме орындалған кейін алынған айнымалылар мен мәндерінің жиынтығы.

Алгоритмдік тілді сипаттайтын ереже екі бөлікке бөлінеді: мағына (семантика) және синтаксис.

Мағына – тілді сипаттаудың мағыналық бөлігі. Ол алгоритмдерді қалай түсіну (адамға) және қалай орындау (машинада) керектігін анықтайды.

Синтаксис- алгоритм тілінде алгоритмді жазудың немқұрайлы ережелер жиыны.

Алгоритм қасиеттері

- Түсінікті – барлық нұсқаулар орындаушыға түсінікті.
- Анық – әр нұсқау әрекетті анықтайды.
- Жалпы – алгоритмді әртүрлі берілгендер жиынына қолдана алу.
- Нәтижелі.

Алгоритм құрылымы. Блок-схема

1. Сызықтық немесе тізбекті алгоритм.

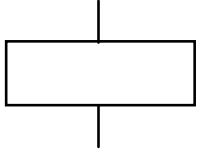
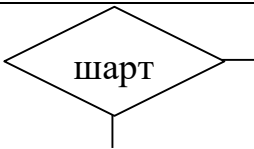
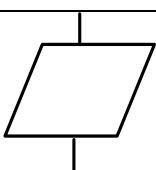
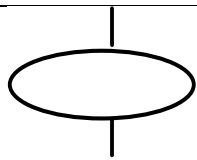
Әрекеттердің тізбектей орындалуын сипаттайтын алгоритм *сызықты алгоритм* деп атайды. Сызықтық алгоритм тізбектеле орналасқан командалардан, ал блок-схемалар бір сызық бойына орналасқан тізбекті блоктардан тұрады.

2. Тармақталу алгоритмдері.

Тармақталу алгоритмінде көбінесе арифметикалық теңсіздік (теңдік) түрінде берілген логикалық шарт тексеріледі. Егер ол орындалса, онда алгоритм бір тармақпен, ал орындалмаса екінші тармақпен жүзеге асырылады да, соңында екі тармақ қайта бірігеді. Мұндай алгоритмдерде шартты тексеру тармақталу командасы деп аталады. Оны алгоритмдік тілде өрнектегенде **егер, онда, әйтпесе** түйінді сөздері пайдалынады.

3. Циклдік алгоритмдер.

Көптеген алгоритмдерде белгілі бір әрекеттер тізбегі бірнеше рет қайталанып орындалып отырады. Осындай есептеу процесі бөліктерінің қайталап орындауын *ЦИКЛ* деп атайды, ал қайталанатын бөлігі бар алгоритмдер тобы циклдік алгоритмдерге жатады. Қайталану командасын алгоритмдік тілде жазу үшін **әзірше (әзір), цикл басы (цб) және цикл соңы (цс)** түйінді сөздері қолданылады; **әзірше** сөзінен кейін қойылатын **шарт**, ал **цикл басы** мен **цикл соңы** түйінді сөздерінің арасына қайталанатын командалар жазылады.

Іс-әрекеттің аталуы	Блок-схема түрі	Негізгі әрекеті
Есептеу процесі		Математикалық өрнектерді есептеу.
Логикалық блок. Таңдау.		Есеп шығару жолын таңдау
Енгізу, шығару		Мәліметті енгізу (шығару)
Басы, аяғы		Алгоритмдердің басы, соңы

2. Алгоритмді жазу құралдары мен әдістері.

Алгоритмді жазу әдістері:

- Сөзбен (табиғи тілде).
- Құрылымдық-стилдік (алгоритмдік тілде).
- Графикалық (блок-схема).
- Программалау тілдерінде жазу.

3. Есептеу процестерінің негізгі типтері: сызықты, тармақталған және циклді.

Алгоритм типтері

- *Сызықты* – амалдар қатаң түрде бірінен кейін бірі орындалатын алгоритм.
- *Тармақталған* – шарттың орындалуына байланысты бір немесе басқа командалар тізбегі орындалатын алгоритм.
- *Циклдік* – белгілі бір командалар тобы шекті рет орындалатын алгоритм.

4. Есепті дайындау және компьютерде шығару кезеңдері

1. *Есептің математикалық қойылымы:*

- берілгені – бастапқы шарттарды тізу;
- нені алу керек – нәтижелерін тізу;
- бастапқы берілгендердің жарамдылық шарттары.

2. *Математикалық моделін құру* – нәтижесін алуға қажетті ережелер мен шарттар.

3. *Шешу әдісі* – математикалық модельді оптималды қолдану.

4. *Есептің алгоритмделуі* –математикалық сипатталуы негізінде шешімнің алгоритмін құру.

5. *Программалау* – алгоритмді компьютерде орындау мүмкіндігін қамтамасыз ететін программа құру.

Негізгі операторлар мен амалдар

Операторлар және оларды қолдану

Оператор	Аты	Түсіндірмесі
+	Қосу	Екі объектіні қосады.
-	Азайту	Екі санның айырмасын табады.Егер бірінші сан жазылмаса, онда оны нөлге тең деп есептейді.
*	Көбейту	Екі санды көбейтеді немесе жолды берілген сан бойыншақайталайды.
**	Дәрежеге шығару	x санын у дәрежесіне шығарады.

Тағайындау операторы

Тағайындау операторы = таңбасы болып табылады. Оператор стандартты түрде орындалады: алдымен теңдік белгісінің оң жағындағы өрнектер есептеледі, содан кейін алынған мән теңдік белгісінің сол жағында көрсетілген айнымалыға жазылады.

```
3  A = 3.14
1  print(type(A)) # float
2  A = 'Hello'
3  print(type(A)) # str
4  a = b = c = 0
5  a += 1          # a = a + 1
6  c = 5//2        # int
7  d = 5/2         # float
8  b = c**2        # b = c^2 (дәрежесі)
1  a, b = b, a     # мәндер алмасу a=b, b=a
```

Python-дағы түсініктемелер

Бір жолдық түсініктеме # -дан басталады

'''

Сіз осындай блоктық түсініктеме пайдалана аласыз(жалпы,
Python-да блоктық түсініктеме жоқ

'''

Python тілінің синтаксисі. Негізгі ережелері[2]:

- Жолдың соңы интрукцияның соңы болып табылады (нүктелі үір қажет емес);

- Ендірілген инструкциялар шегіністер өлшемі (көлемі) бойынша блоктарға біріктіріледі. Шегініс кезкелген болуы мүмкін, бастысы бір ендірілген блокқа бірдей шегініс өлшемі қолданылса болғаны. Кодтың оқылымы туралы да ұмытпау керек. 1 бос орын шегініс - ол ешқандай шешім болып табылмайды. Тым болмағанда 4 бос орын шегініс немесе табуляция белгісін қолданған дұрыс болады.

- Python тілінде әр инструкциялар бір шаблонға сәйкес жазылады, негізгі инструкция қос нүктемен аяқталған соң, оның соңынан инструкцияның ендірілген блогы шегініс арқылы орналасады. Келесі суретте жазылуы бейнеленген:

Бірнеше арнайы жағдайлар бар. Олар:

1. Кейбір жағдайда бірнеше инструкцияны бір жолға нүктелі үтір арқылы жазуға болады:

```
a = 1; b = 2; print(a, b)
```

Бұндайды көп қолдануға болмайды, әрқашан да оқылым туралы естесактау керек.

2. Бір инструкцияны бірнеше жолға да бөліп жазуға болады. Ол үшін бірнеше доғал, квардатты немесе жүйелі жақшаны қолдану керек.

```
if (a == 1 and b == 2 and c == 3 and  
    d == 4):  
    print('spam' * 3)
```

3. Құрамдас интсрукцияның денесі сол негізгі дененің жазылған жолына орналасуы мүмкін, егер негізгі инструкцияның денесі құрамдас денені қамтымаса. Мысалы:

```
if x > y: print(x)
```

Бағдарламалау тілінің толық синтаксисін түсіну үшін көптегенмысалдарды қарастыру керек.

Деректерді енгізу және шығару

Деректерді шығару print операторының көмегімен жүзеге асырылады:

```
1      a = 1
```

```

2     b = 2
3     print(a)
4     print(a + b)
5     print('сумма = ', a + b)

```

Нұсқауларды бір жолға жазу мүмкіндігі бар. Ол үшін оларды ; арқылы бөлу керек. Алайда, мұндай әдісті жиі қолдануға болмайды, бұл оқу барысында ыңғайсыздықтар тударады:

```

1     a = 1; b = 2; print(a)
2     print(a + b)
3     print('сумма = ', a + b)

```

Print функциясы үшін шығару элементтері арасында сепаратор – айырғыш берілуі мүмкін:

```

1     x=2
2     y=5
3     print ( x, "+", y, "=", x+y, sep = " ")

```

Нәтижесі элементтер арасында бос орын қалдырылып көрсетіледі: 2 + 5 = 7
Өзгертіліп шығару үшін format қолданылады:

```

1     x = 11
2     print ( "{:4d}".format(x) )

```

Нәтижесінде 11 саны шығарылады, ал оның алдында екі бос орын болады, себебі шығару үшін төрт белгі орнын қолдану керектігі көрсетілген.

Немесе бірнеше аргументтермен:

```

1     x = 2
2     print ( "{:4d}{:4d}{:4d}".format (x, x+x, x*x) )

```

Нәтижесінде әр мән белгісі 4 белгі орны есебінен шығарылады.

Деректерді енгізу input операторының көмегімен жүзеге асырылады:

```

1     a = input()
2     print(a)

```

Функцияның жақшасында енгізілген деректерге пікір-хабарлама көрсетуге болады:

```
a = input ("Введите количество: ")
```

Input функциясы енгізу деректерін таңбалар ретінде қабылдайды. Сондықтан, бүтін мәнді қабылдау үшін int() функциясын пайдалану керек:

```
a = int (input())
```

Арифметикалық амалдар

x + y Қосу

$x - y$	Азайту
$x * y$	Көбейту
x / y	Бөлу
$x // y$	Бүтін бөлу
$x \% y$	Бүтін бөлуден қалған қалдық $x ** y$
	Дәрежеге шығару
$-x$	Сан белгісін ауыстыру

Python-да әртүрлі математикалық операциялар бар. Бұл сабақта біз ең негізгі амалдарды қарастырамыз (қосу, азайту, көбейту және бөлу). Ең алдымен, қосу мен азайту амалдарын қарастырайық. Қосу мен азайту амалдарын орындау үшін, біз тиісінше $+$ пен $-$ амалдарын қолданамыз. Мысалы,

```
print(2 + 2)
print(10 - 6)
```

Көріп тұрғаныңыздай, бұл операторлар орындалуда қарапайым болып табылады. Енді, көбейту мен бөлу амалдарын қарастырайық. Python-да көбейту

* символы арқылы белгіленеді, ал бөлу / символы арқылы белгіленеді.

Мысалы,

```
print(15 *
2),
print(6 / 3).
```

Python-да осы негізгі амалдардан басқа, тағы бірнеше амалдар бар. Python-дағы барлық математикалық амалдар математикалық заңдарға толық бағынады. Мысалы, жақшалар амалдың қай бөлігін бірінші кезекте орындау керектігін көрсету үшін қолданылады. Python әрқашан ең алдымен жақшаның ішіндегі амалдарды орындайды, тек одан соң келесі амалдарға көшеді.

Мысалы,

```
print((5 + 5) *
2)
print(3 + (4 * 5))
print(10 / (6 - 1)).
```

Осыған қоса, біз сандық мән берілген айнымалылармен математикалық операцияларды орындай аламыз. Бірақ, мәтінді айнымалыларды біз бір-біріне тек қоса аламыз. Бұл түсінік – конкатенация деп аталады.

Python тілінде математикалық функциялардың жазылуы

Математикалық функция	Python тілінде жазылуы түрі
Sin x	sin(x)
Cos x	cos(x)
Ln x	ln(x)
$\sqrt{x}$	sqrt(x)
Arctg x	arctan(x)
e^x	exp(x)
x^2	sqr(x)
x	abs(x)