

Тақырып 7 КОРТЕЖДЕР МЕН ТІЗІМДЕР

Зертханалық жұмыс №5

7.1 Жұмыстың мақсатты

Тізімдерді пайдалана отырып, есептеу процесін әзірлеу мен бағдарламалаудың практикалық дағдыларын меңгеру.

7.2 Теориялық мәлімдемелер

Python-дағы тізімдер кортеждерге ұқсас жұмыс істейді, айырмашылығы өзгерілетін тізбек болғандықтары. Осылайша, егер жаңа элементті қосу, элементті жою, сұрыптау және т.б. сияқты әрекеттер мүмкін болмаса, онда тізіммен жұмыс істеу әдістері бұған мүмкіндік береді.

Кортеж элементтері сияқты, тізім элементтері тек объектіге сілтеме жасайды, сондықтан тізімдерге әртүрлі мәліметтер кіруі мүмкін. Бұл тәсіл дәстүрлі массивтерді басқа бағдарламалау тілдерінде өңдеуден ерекшеленеді, дегенмен Python-да тізімдерді өңдеу әдістерінде массивтерді өңдеу әдістерімен ұқсастық бар.

Тізімдерді жариялау синтаксисын қарастырайық .

Тізім атауы = [элемент 1, элемент 2, ...элемент N]

Мысаплы ,

```
spisok=[1, 2, 3, 4, 5]
```

Тізімдермен жұмыс істеу кезіндегі типтік тапсырмалар:

- оларда берілген элементтің болу фактісін анықтау;
- белгілі бір шарттарды қанағаттандыратын элементтерді таңдау.

Екі жағдайда да тізім элементтерін берілген үлгімен циклдік салыстыру қолданылады. Берілген үлгінің тізімде болу фактісін анықтау үшін бір сәйкестік жеткілікті, содан кейін одан әрі қарау тоқтатылады. Егер таңдау шарты тізімнің бірнеше элементтері үшін орындалса, онда бүкіл тізімді соңына дейін қарау керек.

Жоғарыда қарастырылған барлық әдістер тізімдер үшін жарамды, бірақ тізімдердің өзгеруіне байланысты олардың жиынтығы айтарлықтай кеңейтілуі мүмкін. Атап айтқанда, тізім элементтерін жоюға және қосуға, сұрыптауға болады.

Қолданушы тізімді құра алады, тік жақшада (жоғарыда көрсетілген синтаксиске сәйкес), оның өңдеуді қажет ететін элементтерін орналастыра отырып, немесе кездейсоқ түрде генерациялау арқылы. Бұл әдістерді толығырақ "Тізім элементтерінің қосындысын табу" мысалында қарастырайық. Бұрын осы тақырыпта біз дәл сол есептің шешімін қарастырдық, бірақ кортеждермен жұмыс істеу үшін. Мұнда және одан кейін Python-да тізбекті өңдеу әдістерін салыстыру пайдалы болады.

```
sum=0
spisok=[1, 2, 3, 4, 5]
print(" Список ")
for i in spisok:
    print(i, end=" ")
for i in range(5):
    sum=sum+spisok[i]
print("\n Сумма элементов списка = ", sum)
```

Тізімдерді кездейсоқ түрде құру мүмкіндігін қарастырайық. Мұны бірнеше жолмен жасауға болады. Біріншіден, біз білетін **range** функциясында тізімнің басталу және аяқталу мәндерін көрсетуге болады, ал **list** функциясы тізімді қайтарады. Сонда алдыңғы бағдарламаның коды келесідей болады:

```
sum=0
spisok=list(range(1, 21))
print(" Список ")
print(spisok)
for i in range(1, 20, 1):
    sum=sum+spisok[i]
print("\n Сумма элементов списка = ", sum)
```

Сіз **sample** функциясын пайдалана аласыз, ол тізім элементтерінің бастапқы реттілігінен пайдаланушы көрсеткен элементтер санын қайтарады. Бұл жағдайда **random** модульді **import** операторының көмегімен қосу арқылы қолдану керек. Келесі бағдарламада көрсетілгендей, бастапқы тізімде он элемент бар. Сонда **chislo = random.sample (spisok, 5)** бастапқы тізімнен бес элемент генерацияланады(құрылады).

```
import random
sum=0
spisok=[11, 25, 38, -5, 0, 12, -78, 27, 39, 19]
chislo=random.sample(spisok, 5)
print(chislo, end=" ")
for i in range(0, 5):
    sum=sum+chislo[i]
print("\n Сумма элементов списка = ", sum)
```

Алдыңғы кодтарды біріктірейік. Алдын ала анықталған тізімнің орнына біз оны **range** функциясын пайдаланып генерациялаймыз , содан кейін **sample** функциясын қолдана отырып(7 санын параметрге айналдырамыз), осылайша тізімдегі элементтер санын шектейміз (жетіге дейін).

```
import random
sum=0
chislo=random.sample(range(100), 7)
print(chislo, end=" ")
for i in range(0, 7):
    sum=sum+chislo[i]
print("\n Сумма элементов списка = ", sum)
```

Random модульде тізімді кездейсоқ араластыруға болатын **shuffle** функциясы бар.

```
import random
sum=0
spisok=[11, 25, 38, -5, 0, 12, -78, 27, 39, 19]
random.shuffle(spisok)
print(spisok, end=" ")
for i in range(0, 5):
    sum=sum+spisok[i]
print("\n Сумма элементов списка = ", sum)
```

Бағдарламада тізімдер жасаудың талқыланған әдістеріне қосымша динамикалық тізімді құруды қолдануға болады. Төмендегі кодта көрсетілгендей, алдымен **sp = []** операторымен бос тізім жарияланады. Содан кейін пайдаланушыдан тізімдегі элементтер саны сұралады. Циклде қолданушы тізім элементтерін тере бастайды, ал **append ()** әдісі оларды тізімге қосуға мүмкіндік береді.

```
sp=[]
n=int(input("\n Введите количество элементов списка "))
for i in range(n):
    chislo=int(input("\n Введите число "))
    sp.append(chislo) # Добавляем элементы списка
for i in range(n): # Выводим элемента списка
    print(sp[i], end=" ")
```

Осылайша, Python-да тізімдер құрудың негізгі жолдары қарастырылған. Жоғарыда талқыланған тізімдер құрудың негізгі әдістерін біріктіру арқылы тізімдерді анықтауға болады. Python-да тізімдермен жұмыс істеу әдістері қарастырылған, олардың кейбіреулері 1-ші кестеде көрсетілген.

Кесте 1 – Тізімдермен жұмыс істеуге арналған әдістер

Әдіс	Әдістің сипаттамасы
<code>spisok.append(x)</code>	Тізімнің соңына x мәнін қосады
<code>spisok.insert(i, x)</code>	i позициясына x мәнін қосады
<code>spisok.extend(spisok1)</code>	<code>Spisok1</code> тізімінің барлық элементтерін соңына қосу арқылы тізімді кеңейтеді
<code>spisok.remove(x)</code>	x мәні бар тізімдегі бірінші элементті жояды
<code>spisok.pop(i)</code>	i нөмірінің мәнін позицияға қайтарады және сонымен бірге оны тізімнен жояды. Егер аргумент берілмесе, тізімнің соңғы элементі қайтарылады және жойылады
<code>spisok.count(x)</code>	x мәні бар элементтер санын қайтарады
<code>spisok.sort([reverse = True])</code>	Элементтерді өсу бойынша сұрыптайды. Reverse параметрі міндетті емес және логикалық мәндерді қабылдайды. Егер сіз True мәнін берсеңіз, тізім кему бойынша сұрыпталады
<code>spisok.reverse()</code>	Кері тізімді қайтарады

7.3 Практикалық жұмысты орындауға арналған тапсырмалар

Есеп 1. Пайдаланушы енгізген элементті кездейсоқ құрылған тізімге қосыңыз.

Шешімі . Төмендегі бағдарламадан көріп отырғанымыздай, алдымен тізім жасалады, содан кейін нөмір сұралады, ал тізім тізімінің соңына қосымша әдісі арқылы қосылады.

```
import random
spisok=random.sample(range(100), 7)
print(spisok, end=" ")
chislo=int(input("\n Введите число "))
spisok.append(chislo) # Добавляем введенное число в конец списка
print(spisok, end=" ")
```

Есеп 2. Пайдаланушы көрсетілген позицияға енгізген элементті кездейсоқ құрылған тізімге қосыңыз.

Шешімі. Төмендегі бағдарламада алдымен тізім жасалады, содан кейін санды қосу үшін сан мен позиция нөмірі сұралады. Содан кейін **insert ()** әдісін тиісті **poz,chislo** параметрлерімен пайдаланып, санды көрсетілген позицияға енгізеді.

```
import random
spisok=random.sample(range(100), 7)
print(spisok, end=" ")
chislo=int(input("\n Введите число ")) # Вводим число
poz=int(input("\n Введите позицию ")) # Вводим позицию для добавления
```

введенного выше числа в список

```
spisok.insert(poz, chislo) # Добавляем введенное число
print(spisok, end=" ")
```

Есеп 3. Кездейсоқ жасалған екі тізім бар. Екінші тізімнің барлық элементтерін бірінші тізімнің соңына қосыңыз.

Шешімі. Алдымен кездейсоқ құрылған екі **spisok1** және **spisok2** тізімдерін енгізейік. Бірінші тізімдегі **extend()**, әдісін қолдану арқылы біз оны екінші тізім элементтерімен кеңейтеміз.

```
import random
spisok1=random.sample(range(100), 7)
print("\n Первый список ")
print(spisok1, end=" ")
spisok2=random.sample(range(100), 5)
print("\n Второй список ")
print(spisok2, end=" ")
spisok1.extend(spisok2) # Добавляем к первому списку элементы второго
списка
print("\n Объединенный список ")
print(spisok1, end=" ")
```

Есеп 4. Пайдаланушы енгізген элемент алдын-ала жасалған тізімнен алынып тасталуы керек.

Шешімі. Бастапқы тізім бес элементтен тұрады. Пайдаланушы жойылатын элементтің мәнін енгізеді. Тізімге қолданылатын **remove()**, әдісі көрсетілген әрекеттерді орындайды.

```
spisok=[8, 9, -7, 7, 0]
for i in spisok:
    print(i, end=" ")
chislo=int(input("\n Введите число "))
spisok.remove(chislo)
print(spisok, end=" ")
```

Есеп 5. Кездейсоқ жасалған тізімді өсу ретімен сұрыптаңыз.

Шешімі. Тізімді сұрыптау үшін Python-да қолданылатын **sort()** әдісі **reverse** міндетті емес параметрі бар. Төмендегі кодта ол **False**-ге орнатылған, бұл тізімді өсу ретімен сұрыптауға мүмкіндік береді.

```
import random
spisok=random.sample(range(100), 7)
print("\n Сгенерированный список ", spisok, end=" ")
spisok.sort(reverse=False)
```

print("\n Отсортированный список ", spisok, end=" ")

7.4 Практикалық жұмыстың тапсырмалары

Бағдарламалық түрде келесі есепті нұсқа бойынша орындаңыз.

Тапсырма нұсқалары

Нұсқа №	Тапсырмалар
1	$X = [-1.5, 0.1, 12, 0, -2.2, 0.5, -1, 0.3]$ тізімін құрыңыз. $[0, 1]$ кесіндіге жататын тізімдегі элементтердің саны мен қосындысын шығарыңыз
2	$a = [5, -2, 0, 3, 4, 12, 7]$ тізімін құрыңыз. Тізімдегі оң элементтердің арифметикалық ортасын есептеп шығарыңыз.
3	$x = [-1.5, 0, 0.8, 2.2, 3, 0.5, 0.1]$ тізімін құрыңыз. $[-1, 1]$ кесіндісінде жататын тізім элементтерін y тізіміне қайта жазып, оны шығарыңыз.
4	Еркін сандардан тұратын $a[5]$ және $b[5]$ тізімдерінің құрып, тізімдердің қайсысының оң элементтері көп екенін анықтаңыз. A және b тізімдерінің оң элементтерінен тұратын жаңа c тізімін басып шығарыңыз
5	$Z = [-2, 0, 3.5, 7, -12, 5, -1, 3]$ тізімін құрыңыз. Алдымен R тізіміне Z тізімінің оң, содан кейін оң емес элементтерін орналастырыңыз. R тізімін шығарыңыз.
6	$A = [2.35, -4.15, 0, -3.1, 7.8, 6.3, -3.05, 1.5]$ тізімін құрыңыз. A тізімінің оң элементтерінің геометриялық ортасын және геометриялық ортаға жақын элементтің индексін табыңыз және шығарыңыз.
7	$A [10]$ тізімін құрыңыз, оның элементтері $a [i + 1] = a [i] + i^2$ ережесі бойынша құрылуы керек; $a [1] = 2$ және оның элементтерінің арифметикалық ортасын есептеңіз.
8	$Y = [2.5, 4.9, 10.2, -7.12, 3.1, -2, 6]$ тізімін құрыңыз. Одан z жаңа тізімін құрыңыз, оның элементтері y тізімнің оң элементтері болады және z тізімін өсу ретімен сұрыптаңыз.
9	$Y = [-6.3, 0.8, 12, -4, 13, 2.5, 7, 8, -9, 10]$ тізімін құрыңыз. Тізімнің максималды және минималды элементтерін тауып шығарыңыз. Оларды орындарымен ауыстырыңыз. Алынған y тізімін шығарыңыз.
10	$x = [4.1, 16, 0, -3.2, 12]$ және $y = [4, 5.1, 6]$ тізімнің құрыңыз. U тізімнің элементтерін x тізімнің үшінші және төртінші элементтерінің арасына қойып, оларды z тізіміне біріктіріңіз. Z тізімін шығарыңыз.

11	$z = [0, 1.6, 6.4, 3.8, -7, 1, -2]$ және $a = [5, 4, 6.4, 1]$ тізімнің құрыңыз. А және z тізімдері элементтерінің ішінен ең кіші индекстері бар екі бірдей элементті тауып, олардың мәндері мен индекстерін шығарыңыз.
12	$n = [3, 5, 7, 9, 11, 13, 15]$ тізімін құрыңыз. N тізімінің элементтерін кері ретімен қайта орналастырыңыз және оны басып шығарыңыз.
13	$m = [-1, 0, 10, -3, -5, 6, -2, 3, 4]$ тізімін құрыңыз. N тізімін құрыңыз және басып шығарыңыз, оның элементтері m тізімінің оң элементтерінің индекстері болып табылады.
14	$l = [13, 4, -2, 6, 7, -1, -5, 2, -3, 4]$ тізімін құрыңыз. $M[0]n[0] + m[1]n[1] + \dots + m[k]n[k]$ есептеп шығарыңыз, мұндағы $m[0], m[1], \dots, m[p]$ – 1-ші тізімнен ретімен алынған теріс элементтері; $n[0], n[1], \dots, n[q]$ - кері ретімен алынған 1-ші тізімнің оң элементтері; $k = \min[p, q]$.
15	$k = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ тізімін құрыңыз. $K[i], k[i+1]$ элементтерінің жұптарын қайта орналастырыңыз, мұндағы $i = 0, 2, 4, 6, 8$. Алынған тізімді басып шығарыңыз.
16	15 нақты элементтерден тұратын тізімде (кездейсоқ сандар генераторының көмегімен тізім элементтерінің мәндерін орнатыңыз): 1) тізімнің теріс элементтерінің қосындысын; 2) максималды және минималды элементтер арасында орналасқан тізім элементтерінің көбейтіндісін есептеңіз.
17	$X = [6, 3.8, -9.3, 2.87, 16, 5, 0.2, -3.1, 1, 10]$ тізімін құрыңыз. Y-ті ереже арқылы есептеп, Y тізімін қалыптастырыңыз және басып шығарыңыз: $\begin{cases} Y_0 = \sqrt{X_0}, Y_1 = \sqrt{X_0 + X_1}, Y_2 = \sqrt{X_0 + X_1 + X_2}, \text{ егер түбір астында теріс} \\ \text{сан болмаса басқа жағдайда} \\ Y_i = 0 \end{cases}$
18	$M = [6, 10, 7, 14, 12, 12, -2, 3, -9, 6, -10]$ тізімін құрыңыз. Тізімдегі 2-ге бөлінетін және 3-ке бөлінбейтін элементтердің саны мен қосындысын есептеп шығарыңыз.
19	$M = [14, 6, 3, 0, 7, 12, -3, 1, 5, 2]$ тізімді құрыңыз 3-ке еселі тізім элементтерінің көбейтіндісін есептеңіз және шығарыңыз.
20	$L = [7, 6, 15, 17, 12, -12, 4, 0, -10, -22]$ тізімін құрыңыз. Тізімде модуль 5-ке бөлгенде, 2-нің қалдықтарын беретіндерді нөлдермен ауыстырыңыз. Жаңадан құрылған тізімді басып шығарыңыз.
21	$K = [1, 2, 3, 4, 6, 5, 8, 9, 10]$ тізімін құрыңыз. Егер тізім элементтері өсу ретін құраса, «ИӘ» хабарламасын шығарыңыз; әйтпесе, «ЖОҚ» хабарламасы.

22	$Q = [1.5, 2, 3.1, 4.2, 6, 7.5, 8.3, 9]$ тізімін енгізіңіз. Тізімнен 4-ші индексіндегі элементті алып тастаңыз және енгізген s мәнімен элементтің ретін бұзбай орналастырыңыз . Алынған тізімді басып шығарыңыз.
23	15 нақты элементтерден тұратын тізімде (кездейсоқ сандар генераторының көмегімен тізім элементтерінің мәндерін орнатыңыз): 1) тізімнің оң элементтерінің қосындысы; 2) модуль бойынша максимум мен модуль элементтеріндегі минимум арасында орналасқан тізім элементтерінің көбейтіндісі есептеу керек.
24	$B = [15.1, 0.8, 32.3, 7.5, -1.5, 2.4, 6.3, 15.5]$ тізімін құрыңыз.Тізім элементтерінің арифметикалық ортасын және орташа арифметикалықтан үлкен элементтердің санын санап және басып шығарыңыз
25	20 нақты элементтерден тұратын тізімде (кездейсоқ сандар генераторы көмегімен тізім элементтерінің мәндерін орнатыңыз): 1) тізімнің минималды элементінің нөмірі; 2) бірінші және екінші теріс элементтер арасында орналасқан тізім элементтерінің қосындысын есептеп. Тізімді алдымен модулі 1-ден аспайтын барлық элементтерді, содан кейін қалғандарын орналасатындай етіп түрлендіріңіз.
26	20 бүтін элементтерден тұратын тізімде (кездейсоқ сандар генераторының көмегімен тізім элементтерінің мәндерін орнатыңыз): 1) тізімнің максималды элементінің нөмірі; 2) бірінші және екінші нөлдік элементтер арасында орналасқан тізім элементтерінің көбейтіндісі. Тізімді бірінші жартысында тақ позицияларда болатын элементтерді, ал екінші жартысында жұп позицияларда болатын элементтерді орналастырып шығарыңыз ..
27	$X = [-6, 0.5, 0.12, 13, -10.1]$ және $y = [13, 2.1, 14, 6, -2]$ тізімдерін енгізіңіз. $R [10]$ тізімін құрыңыз, онда тақ санды элементтері x тізімде орналасады, ал жұп сандар y тізімінде болады. Тізім r -ді басып шығарыңыз.
28	$A = [-2, 0, 3.1, 4.6, 10]$, $b = [4, 7, 9.1, 12, 63]$ тізімдерін енгізіңіз. A және b тізімдерінің элементтерінен c тізімін құрыңыз. Оны өсу ретімен орналастырышып шығарыңыз.
29	20 бүтін элементтерден тұратын тізімде (кездейсоқ сандар генераторының көмегімен тізім элементтерінің мәндерін орнатыңыз): 1) жұп сандары бар тізім элементтерінің көбейтіндісін есептеп; 2) тізімде алдымен тізімнің барлық оң элементтері, содан кейін барлық теріс элементтерін орналастырып түрлендірген тізімді шығар(нөлге тең элементтер, оң деп сана).
30	Бүтін типтегі 20 элементтен тұратын тізімде (кездейсоқ сандар генераторының көмегімен тізім элементтерінің мәндерін орнатыңыз):

	<p>1) тізімнің минималды элементін анықтап;</p> <p>2) бірінші және соңғы оң элементтер арасында орналасқан тізім элементтерінің қосындысын есептеу керек.</p> <p>Алдымен нөлге тең барлық элементтер орналасатындай етіп тізімді түрлендіріңіз, содан кейін қалған элементтер.</p>
31	X = [-1, 2, -3, 4, -5, 0, 6, 8, 9] тізімін құрыңыз. Y тізіміне x тізімінің оң элементтерін бір қатарға қайта жазыңыз. Y тізімін шығарыңыз.
32	G = [-3.1, 2.8, 5, 7.7, -7.5, 7.6, 3, 0] тізімін құрыңыз. Тізімнің минималды элементін және оның номерін анықтаңыз және шығарыңыз.
33	B = [7.35, 0.12, -7, 3.12, 2.87, -4.12, 5.32, 0, 6.5] тізімін құрыңыз. Тізімнің максималды элементін және оның номерін анықтаңыз және шығарыңыз.
34	<p>20 нақты элементтерден тұратын тізімде (кездейсоқ сандар генераторы көмегімен тізім элементтерінің мәндерін орнатыңыз):</p> <p>1) тізімнің максималды элементін;</p> <p>2) соңғы оң элементтің алдында орналасқан тізім элементтерінің қосындысын есептеп шығар.</p>
35	<p>20 нақты элементтерден тұратын тізімде (кездейсоқ сандар генераторы көмегімен тізім элементтерінің мәндерін орнатыңыз):</p> <p>1) тізімдегі тақ сандар элементтерінің қосындысын ;</p> <p>2) бірінші және соңғы теріс элементтер арасында орналасқан тізім элементтерінің қосындысын есептеп шығар.</p>

7.5 Тәжірибелік жұмыс бойынша есепті дайындауға қойылатын талаптар

Тәжірибелік жұмыс туралы есепті дайындаған кезде келесі құрылым мен элементтердің кезектілігі ұсынылады:

- бастапқы бет;
- зертханалық жұмыстың атауы;
- зертханалық жұмыстың мақсаты;
- зертханалық жұмыстарға арналған жеке тапсырма (нұсқаларға сәйкес);
- жеке тапсырманы орындау туралы қысқаша түсініктемелер және есепті шешу алгоритмінің блок-схемасы;
- жеке тапсырмаға қажетті бағдарламалық код; - бағдарламаның нәтижелері; - тұжырымдар.

Зертханалық жұмыстарға арналған жеке тапсырма оқытушыдан алынған жеке тапсырманың толық мәтінін, жеке тапсырманы орындау алгоритмінің сипаттамасын және есепті шығарудың алгоритмінің блок-схемасын қамтиды.

Жеке тапсырмаға қажетті бағдарлама кодында студент әзірлеген бағдарлама кодының толық мәтіні болады. Бағдарлама жұмысының нәтижелерінде әдетте бағдарламаның жұмыс терезелерінің көшірмелері болады.

7.6 Зертханалық жұмысты қорғауға арналған сұрақтар

7.6.1. Тізімдерді жариялауға арналған синтаксисті жазыңыз.

7.6.2. Тізімдер жасау үшін Python тілінің қандай ерекшеліктері қолданылады?

7.6.3. Тізімдермен жұмыс жасаудың негізгі әдістерін атап өтіңіз.

7.6.4. Тізімнің әр элементіне оны өңдеген кезде қалай қол жеткізуге болады.

7.6.5. Тізім элементтерінің мәндері қалай алмасатынын түсіндіріңіз.

Мысалдар келтіріңіз.