Жамбыл облысы әкімдігі білім басқармасы

«Білім» кәсіби гуманитарлық-техникалық колледжі

(білім беру ұйымының атауы)

**Оқу сабағының жоспары**

(теориялық немесе өндірістік оқыту)

**Python (пайтон) программалау тілінде екі өлшемді массивте деректерді енгізу және шығару.**

(сабақ тақырыбы)

**Модуль/Пән атауы** Бағдарламалық кодты қайта өңдеу

**Дайындаған педагог** Нургисаева У.М

**20\_**25**\_ жылғы** «\_\_\_\_» \_\_\_наурыз\_\_\_

**1. Жалпы мәліметтер**

Курс, оқу жылы, топ 3 курс, 3БҚ-1-22, 3БҚ-2-22

Сабақ түрі: Аралас

**2. Мақсаты, міндеттері:**

**Оқу:** Білім алушыларға python (пайтон) программалау тілінде екі өлшемді массивте деректерді енгізу, есеп шығаруды жүзеге асыру, қолдану жолдарын ұйымдастыру, программаны өңдеу жұмыснегіздерін үйрету, шартты операторлардың жұмыс істеу принциптерін түсіндіру.

#### Дамыту: Логикалық ойлау қабілеттерін дамыту, есептердің әртүрлі шешімдерін табуға дағдыландыру, программалық кодтың құрылымын дұрыс құру қабілеттерін қалыптастыру.

#### Тәрбиелік: Жауапкершілік пен ұқыптылыққа тәрбиелеу, алгоритмдік ойлауды дамыту, шығармашылық қабілеттерін жетілдіру.

**3. Оқу-жаттығу процесінде білім алушылар меңгеретін күтілетін нәтижелер және кәсіби дағдылар тізбесі:** Python программалау тілінде екі өлшемді массивтерге есеп шығаруды жүзеге асыру, қолдану жолдарын ұйымдастыру, программаны өңдеу және пайдалану дағдыларын меңгеру. Array, матрица, рандом басқару, қолдану жолдарының жұмыс негіздерін үйренеді, код жазады.

**4. Қажетті ресурстар:** ДК немесе ноутбуктер

<https://ppt-online.org/982702>

***5. Сабақтың барысы: (90 минут)***

**5.1. Ұйымдастыру кезеңі:** *( 3 мин )*

**5.2. Үй жұмысын жан-жақты тексеру:**

***“Сұрақ-жауап” әдісі*** *(15 минут)*

Python (пайтон) программалау тілінде екі өлшемді массивте деректерді енгізу және шығару.

Екі өлшемді массив – бұл элементтері екі индекстен тұратын мәліметтер құрылымы. Ол өзінде қатарлар мен бағандардан тұратын мәліметтерді сақтайды, яғни екі өлшемді массивті көбінесе кестелер немесе матрицалар ретінде бейнелеуге болады.

Python-да массивтер **list** (тізім) деректер құрылымы арқылы жүзеге асырылады. Бірнеше тізімді бір-біріне қосу арқылы екі өлшемді массив жасай аламыз.

### Екі өлшемді массивтермен жұмыс істеу

Python-да екі өлшемді массивті жасау және оған деректер енгізу қарапайым операция болып табылады. Оны қарапайым Python тізімдері арқылы орындауға болады.

#### 1. Екі өлшемді массив жасау

Екі өлшемді массив жасау үшін тізімдерден тізімдер құра аламыз. Мысалы:

1-мысал

# Екі өлшемді массивтің мысалы

matrix = [

[1, 2, 3], # бірінші қатар

[4, 5, 6], # екінші қатар

[7, 8, 9] # үшінші қатар

]

2-мысал

import random

# Матрицаның өлшемдерін анықтау

rows = 3 # қатарлар саны

cols = 3 # бағандар саны

# Екі өлшемді массивті жасау және оны рандом мәндермен толтыру

matrix = []

for i in range(rows):

row = [] # әр қатар үшін бос тізім

for j in range(cols):

value = random.randint(1, 100) # 1 мен 100 арасында кездейсоқ бүтін сан

row.append(value) # қатарға қосу

matrix.append(row) # қатарды матрицаға қосу

# Енгізілген матрицаны шығару

print("Екі өлшемді матрица (рандом мәндермен):")

for row in matrix:

print(row)

Бұл мысалда бізде 3 қатар және әр қатарда 3 бағаннан тұратын матрица бар. Әрбір ішкі тізім (list) бір қатарды білдіреді.

#### 2. Екі өлшемді массивке деректерді енгізу

Пайдаланушыдан екі өлшемді массивке деректерді енгізу үшін біз **input()** функциясын қолданамыз. Мұнда біз пайдаланушыдан әр қатар мен баған бойынша мәндерді сұрау арқылы массивті толтыра аламыз.

Мысалы, үш қатарлы және үш бағанды матрицаға деректер енгізу:

# Пайдаланушыдан деректер енгізу

rows = 3 # қатарлар саны

cols = 3 # бағандар саны

# Екі өлшемді массивті жасау

matrix = []

for i in range(rows):

row = [] # әр қатар үшін бос тізім

for j in range(cols):

value = int(input(f"matrix[{i}][{j}] элементін енгізіңіз: ")) # әр элементті енгізу

row.append(value) # қатарға қосу

matrix.append(row) # қатарды матрицаға қосу

print("Енгізілген матрица:")

for row in matrix:

print(row)

Бұл кодтың жұмыс істеу принципі:

* Алдымен matrix атты бос тізім (екі өлшемді массив) жасалады.
* Содан соң, сыртқы цикл арқылы қатарлар санын белгілеп, ішкі цикл арқылы әр қатарға деректер енгізіледі.
* Әр енгізілген элемент ішкі тізімге қосылып, соңында барлық қатарлар басты тізімге қосылады.
* Әр қатарды жеке басып шығару арқылы екі өлшемді массивтің нәтижесін көруге болады.

import random

# Пайдаланушыдан матрицаның өлшемдерін сұрау

rows = int(input("Қатарлар санын енгізіңіз: "))

cols = int(input("Бағандар санын енгізіңіз: "))

# Екі өлшемді массивті жасау және оны рандом мәндермен толтыру

matrix = []

for i in range(rows):

row = [] # әр қатар үшін бос тізім

for j in range(cols):

value = random.randint(1, 100) # 1 мен 100 арасында кездейсоқ бүтін сан

row.append(value) # қатарға қосу

matrix.append(row) # қатарды матрицаға қосу

# Енгізілген матрицаны шығару

print("Екі өлшемді матрица (рандом мәндермен):")

for row in matrix:

print(row)

#### 3. Екі өлшемді массивті шығару

Екі өлшемді массивті шығару үшін, әрбір қатарды жеке шығару керек. Оны Python тізімдерін қайталай отырып жасауға болады:

print("Матрицаның мәндері:")

for row in matrix:

print(row)

Бұл код әрбір қатарды бір-бірден шығарып береді.

#### 4. Индекстеу және элементтермен жұмыс істеу

Python-да екі өлшемді массивтің әрбір элементін индекстеу арқылы алуға болады. Ол үшін бірінші индексте қатар, ал екінші индексте баған көрсетіледі:

print("Матрицаның бірінші қатарындағы екінші элемент:", matrix[0][1]) # бірінші қатар, екінші баған

print("Матрицаның екінші қатарындағы үшінші элемент:", matrix[1][2]) # екінші қатар, үшінші баған

Бұл мысалдарда біз екі өлшемді массивтің нақты элементтерін индекстеу арқылы ала аламыз.

#### 5. Матрицаны транспонирлеу

Матрица транспонирленгенде, оның жолдары бағандарға айналады. Python тілінде бұл операцияны келесідей орындауға болады:

# Матрицаны транспонирлеу

transposed\_matrix = list(zip(\*matrix))

print("Транспонирленген матрица:")

for row in transposed\_matrix:

print(row)

Бұл код zip(\*matrix) функциясын қолдана отырып, екі өлшемді массивті транспонирлейді.

Жиі тапсырмаларда деректермен тікбұрышты кестелерді сақтау керек. Мұндай кестелер матрицалар немесе екі өлшемді массивтер деп аталады. Бағдарламалау тілінде Питон кестені жолдар тізімі ретінде ұсынуға болады, оның әрбір элементі өз кезегінде тізім болып табылады, мысалы, сандар. Мысалы, екі жол мен үш бағаннан сандық кестені жасауға болады:

A = [[1, 2, 3], [4, 5, 6]]

Мұнда тізімнің бірінші жолы [0] [1, 2, 3] сандардың тізімі болып табылады. Яғни A[0][0] == 1, мәні A[0][1] == 2, A[0][2] == 3, A[1][0] == 4, A[1][1] == 5, A[1][2] == 6.

Тізімді өңдеу және шығару үшін әдетте екі ішкі цикл қолданылады. Бірінші цикл жол нөмірі бойынша, екінші цикл жол ішіндегі элементтер бойынша. Мысалы, екі өлшемді сандық тізімді экранға әр жол сайын шығарып, бір жолдың ішіндегі Бос орындар санын бөлісе аласыз:

for i in range(len(A)):

for j in range(len(A[i]):

print(A[i][j], end=' ')

print()

**Екі өлшемді массивті өңдеу мысалы**

N жолдар мен N бағандарынан шаршы массив болсын. Сол жақ жоғарғы бұрышынан оң жақ төменгі бұрышқа өтетін басты диагональдағы элементтерге (яғни ij олар үшін A[i][j] элементтеріне) 1 мәнін беру қажет, басты диагональдан жоғары элементтерге – 0 мәні, басты диагоналдан төмен элементтерге – 2 мәні. Яғни мұндай массивті алу (N = 4 үшін мысал):

1 0 0 0

2 1 0 0

2 2 1 0

2 2 2 1

Бұл мәселені шешудің бірнеше жолдарын қарастырайық. Басты диагональдан жоғары элементтер – бұл элементтер A[i][j], олар үшін i<j, ал негізгі диагоналдан төмен элементтер үшін i>j. осылайша, біз I және j мәндерін салыстыра аламыз және олар бойынша A[i][j] мәнін анықтай аламыз. Келесі алгоритм аламыз:

for i in range(n):

for j in range(n):

if i < j:

A[i][j] = 0

elif i > j:

A[i][j] = 2

else:

A[i][j] = 1

Бұл алгоритм жаман, өйткені әр элементті өңдеу үшін бір немесе екі if нұсқауларын орындайды. Егер алгоритмді қиындататын болсақ, біз шартты нұсқауларсыз жасай аламыз.

Алдымен біз негізгі диагоналді толтырамыз, ол үшін бір цикл қажет:

for i in range(n):

A[i][i] = 1

Содан кейін 0 мәнімен барлық элементтерді басты диагональдан жоғары толтырамыз, ол үшін I нөмірі бар жолдардың әрқайсысында j=i+1,..., n-1 үшін A[i][j] элементтеріне мән беру қажет. Мұнда біз қосымша циклдар қажет:

for i in range(n):

for j in range(i + 1, n):

A[i][j] = 0

Сол сияқты 2 мәнін j=0 үшін A[i][j] элементтеріне тағайындаамыз, …, i-1:

for i in range(n):

for j in range(0, i):

A[i][j] = 2

Сіз сондай-ақ сыртқы циклдар біріктіруге және тағы бір, ықшам шешім алуға болады:

for i in range(n):

for j in range(0, i):

A[i][j] = 2

A[i][i] = 1

for j in range(i + 1, n):

A[i][j] = 0

Бірақ бұл шешім тізімнің кезекті жолын құру үшін тізімдерді қайталау операциясын пайдаланады. тізімнің I жолы i 2 саннан тұрады, содан кейін бір Сан 1, содан кейін n-i-1 саны 0:

for i in range(n):

A[i] = [2] \* i + [1] + [0] \* (n - i - 1)

Ал болады деген циклды генератор:

A = [[2] \* i + [1] + [0] \* (n - i - 1) for i in range(n)]

**Екі өлшемді массивтердің салынған генераторлары**

Екі өлшемді массивтерді жасау үшін, тізімнің генераторын жолдар үшін генератор ішінде орналастыру арқылы ішкі генераторларды пайдалануға болады. Мысалы, N жолдар мен m бағандар тізімін генератордың көмегімен жасау, әрбір элемент m нөлдер тізімі болып табылады N элементтердің тізімін жасау:

[[0] \* m for i in range(n)]

Бірақ ішкі тізімді, мысалы, генератордың көмегімен жасауға болады: [0 for j in range(m)]. Бір генераторды екіншісіне салу арқылы ішкі генераторларды аламыз:

[[0 for j in range(m)] for i in range(n)]

Бірақ егер 0 Саны i (жол нөмірі) және j (баған нөмірі) байланысты кейбір өрнекке ауыстырса, онда кейбір формула бойынша толтырылған тізімді алуға болады.

Келесі массивті орнату керек (ыңғайлы болу үшін элементтер арасында қосымша Бос орындар қосылады):

0 0 0 0 0 0

0 1 2 3 4 5

0 2 4 6 8 10

0 3 6 9 12 15

0 4 8 12 16 20

Бұл массивте n = 5 жол, m = 6 баған, және элемент i жолда және j бағанасында мынадай формула бойынша есептеледі: A[i][j] = i \* j.  
Мұндай массивті жасау үшін генераторды пайдалануға болады:

[[i \* j for j in range(m)] for i in range(n)]

1 - Кездейсоқ екі өлшемді массив жасау

import random

rows = 3 # қатарлар саны

cols = 4 # бағандар саны

matrix = [[random.randint(1, 100) for \_ in range(cols)] for \_ in range(rows)]

print("Кездейсоқ екі өлшемді массив:")

for row in matrix:

print(row)

### 2- Матрицаның барлық элементтерінің қосындысын табу

import random

rows = int(input("Қатарлар санын енгізіңіз: "))

cols = int(input("Бағандар санын енгізіңіз: "))

matrix = [[random.randint(1, 100) for \_ in range(cols)] for \_ in range(rows)]

total\_sum = sum(sum(row) for row in matrix)

print("Матрицаның барлық элементтерінің қосындысы:", total\_sum)

### 3- Матрицаның транспонирленген түрін табу

import random

matrix = [[random.randint(1, 100) for \_ in range(3)] for \_ in range(3)]

print("Бастапқы матрица:")

for row in matrix:

print(row)

transposed\_matrix = list(zip(\*matrix))

print("Транспонирленген матрица:")

for row in transposed\_matrix:

print(row)

### 4-Екі матрицаның қосындысын табу

import random

matrix1 = [[random.randint(1, 100) for \_ in range(3)] for \_ in range(3)]

matrix2 = [[random.randint(1, 100) for \_ in range(3)] for \_ in range(3)]

print("Матрица 1:")

for row in matrix1:

print(row)

print("Матрица 2:")

for row in matrix2:

print(row)

sum\_matrix = [[matrix1[i][j] + matrix2[i][j] for j in range(3)] for i in range(3)]

print("Матрицалардың қосындысы:")

for row in sum\_matrix:

print(row)

### 5-Матрицаның әр бағанының қосындысын табу

import random

matrix = [[random.randint(1, 100) for \_ in range(3)] for \_ in range(3)]

print("Матрица:")

for row in matrix:

print(row)

column\_sums = [sum(matrix[i][j] for i in range(3)) for j in range(3)]

print("Әр бағанның қосындысы:")

for i, column\_sum in enumerate(column\_sums):

print(f"Баған {i + 1}: {column\_sum}")

|  |  |  |
| --- | --- | --- |
| **Білдім** | **Білемін** | **Білгім келеді** |
|  |  |  |

**Рефлексия**

**Үй тапсырмасы –** Зертханалық жұмыс №7 орындау және қорғау