Жамбыл облысы әкімдігі білім басқармасы

«Білім» кәсіби гуманитарлық-техникалық колледжі

(білім беру ұйымының атауы)

**Оқу сабағының жоспары**

(теориялық немесе өндірістік оқыту)

For циклі операторымен жұмыс.

(сабақ тақырыбы)

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

**Дайындаған педагог** Нургисаева У.М

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

**1. Жалпы мәліметтер**

Курс, оқу жылы, топ 3 курс, 3БҚ-22

Сабақ түрі: Кіріктірілген

**2. Мақсаты, міндеттері:**

**Оқу:** Білім алушыларға python (пайтон) программалау тілінде For циклі операторымен 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 таңдауды ұйымдастыру және пайдалану дағдыларын меңгеру. For циклі операторымен жұмыснегіздерін үйренеді, код жазады.

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

***5. Сабақтың барысы: (90 минут)***

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

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

***“Ойлан – Жұптас – Бөліс” әдісі*** *(15 минут)*

**For циклі операторымен жұмыс.**

Көп жағдайда аргументтерінің әр түрлі мәні бойынша алгоритмнің белгілі бір бөліктерін бірнеше рет қайталауға тура келеді. Осындай процесстерді ұйымдастыру үшін циклдік құрылымды алгоритмдер – қайталау операторлары қолданылады. Олар әр түрлі заңдылық негізіндегі ереже бойынша қолданылады.

Паскаль тілінде қайталау процессін жүзеге асыратын циклдік операторлардың 3 түрі бар: арифметикалық цикл - FOR, шартын алдын ала тексеретін цикл – WHILE және шартты соңынан тексеретін цикл – REPEAT.

Егер шартын алдын ала тексеретін циклдегі операторды неше рет қайталау керек екендігі белгісіз болып, оның тек қайталану шарты берілсе, онда WHILE, REPEAT операторлары пайдаланылады. Ал FOR операторы қайталану саны алдын ала белгілі болған кезде қолданылады.

FOR операторын *параметрлі цикл операторы* деп те аталады, өйткені, қайталау саны функция аргументі сияқты циклдің параметрі қызметін атқаратын басқару айнымалысы арқылы беріледі.

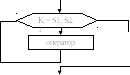
FOR операторының екі түрі бар:

FOR айнымалы:= to do оператор;

FOR айнымалы:= downto do оператор;

Мұндағы, S1 және S2 – цикл параметрінің алғашқы және соңғы мәндерін анықтайтын өрнектер; for…do – цикл тақырыбын анықтайтын түйінді сөздер; - цикл тұлғасы.

FOR – ҮШІН, TO – ДЕЙІН, DO – ОРЫНДАУ деген мағынаны беретін түйінді сөздер. Паскаль тілінде цикл параметрі міндетті түрде бүтін немесе реттелген типтегі айнымалы болуы қажет. Параметрдің өзгеру қадамына байланысты операторда TO немесе DOWNTO (DOWN – төмен, TO – дейін, DOWNTO - кері қарай) түінді сөздері пайдаланылады. Егер қадам +1 ге тең болса, онда операторда TO, ал қадам -1 –ге тең болса, онда DOWNTO сөзі қолданылады.

Параметрдің бастапқы және соңғы мәндері бүтін сан түрінде немесе өсуі, кемуі бойынша реттелетін болуы тиіс, әйтпесе оларды бүтін мен беретін арифметикалық өрнек түрінде жазуға да болады.

*FOR операторының блок-схемасы*

for айнымалы in итерацияланатын\_объект:

цикл\_денесі

Мұндағы:

* айнымалы – әрбір итерацияда (қайталау қадамында) өзіне жаңа мән қабылдайтын айнымалы;
* итерацияланатын\_объект – цикл арқылы өтуге болатын құрылым (мысалы, тізім, диапазон, жол, кортеж және т. б.);
* цикл\_денесі – әр итерация сайын орындалатын код блогы.

Параметрлік цикл деп те аталатын for циклі Python тілінде функционалдылыққа бай. for циклі айнымалыны және айнымалы цикл өтетін мәндер жиынын көрсетеді. Мәндер жиынын тізім, кортеж, жол немесе сандар аралығы-диапазон арқылы көрсетуге болады.

Міне, циклды пайдаланудың қарапайым мысалы, мұнда кортеж мәндер жиыны ретінде пайдаланылады:

i = 1

for color in 'red', 'orange', 'yellow', 'green', 'cyan', 'blue', 'violet':

print('#', i, ' color of rainbow is ', color, sep = '')

i += 1

Бұл мысалда color айнымалысы «red», «orange» және т.б. мәндерін ретімен қабылдайды. Циклдің негізгі бөлігінде түс атауын, яғни color айнымалысының мәнін, сондай-ақ цикл итерациясының нөмірін, алдымен 1-ге тең, содан кейін бірге көбейтетін санды қамтитын хабарлама көрсетіледі. (циклдің әрбір өтуімен i += 1 нұсқауымен.

i += 1 нұсқауы i = i + 1 конструкциясына баламалы (бұл жай ғана стенографиялық белгі). Бұл стенографиялық белгілерді барлық арифметикалық амалдар үшін қолдануға болады: \*=, -=, /=, %=...

Мәндер тізімі әртүрлі түрдегі өрнектерді қамтуы мүмкін, мысалы:

Циклдің алғашқы үш итерациясында i айнымалысы int түріндегі мәнді қабылдайды, келесі үш кезеңде ол str түріндегі мәнді қабылдайды.

for i in 1, 2, 3, 'one', 'two', 'three':

print(i)

range функциясы

Әдетте, for циклдері әрекеттер тізбегін белгілі бір рет қайталау үшін немесе циклдегі айнымалының мәнін кейбір бастапқы мәннен соңғы мәнге өзгерту үшін пайдаланылады.

Циклды белгілі бір n рет қайталау үшін, range функциясымен бірге for циклін пайдалануға болады:

for i in range(4): # for i in 0, 1, 2, 3 инструкциясымен бірдей:

# мұнда циклдік әрекеттерді орындауға болады

print(i)

print(i \*\* 2)

# цикл аяқталды, себебі шегіністі блок аяқталды

print('Цикл соңы')

n сандық тұрақты, айнымалы немесе ерікті арифметикалық өрнек болуы мүмкін (мысалы, 2 \*\* 10). Егер n мәні нөл немесе теріс болса, онда цикл денесі бір рет болса да орындалмайды.

range функциясы бір ғана емес, екі параметрді қабылдай алады. range(a, b) индекс айнымалысы a-дан b - 1-ге дейінгі мәндерді қабылдайтынын білдіреді, яғни екі параметрмен шақырылатын range функциясының бірінші параметрі индекстік айнымалының бастапқы мәнін көрсетеді, ал екінші параметр индекс айнымалысының соңғы мәнін көрсетеді, бірақ соңғы мәні орындалмайды. Егер a≥b болса, онда цикл бір рет болса да орындалмайды. Мысалы, 1-ден n-ге дейінгі сандардың мәндерін қосу үшін келесі бағдарламаны пайдалануға болады:

sum = 0

n = 5

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

sum += i

print(sum)

С++

#include <iostream>

using namespace std;

int main() {

// 1-ден 10-ға дейінгі сандарды шығару

for (int i = 1; i <= 10; i++) {

cout << "Сан: " << i << endl;

} return 0;

}

Бұл мысалда i айнымалысы 1, 2, ..., n мәндерін қабылдайды және sum айнымалысының мәні сол мәндер бойынша дәйекті түрде қосылып(артып) отырады.

Соңында, индекстің айнымалы мәнін азайтатын циклды жасау үшін үш параметрі бар range функциясын пайдалану керек. Бірінші параметр индекстік айнымалының бастапқы мәнін, екінші параметр индекстік айнымалының соңғы мәнін (оны қоспайды!) және үшінші параметр индекс айнымалысының өзгеріс мөлшерін (қадамды) анықтайды. Мысалы, range(1, 100, 2) функциясын пайдаланып 1-ден 99-ға дейінгі барлық тақ сандарды айналдыра аласыз және range(100, 0, -1) пайдаланып 100-ден 1-ге дейінгі барлық сандарды кему ретімен айналдыра аласыз.

for i in range(a, b, d) циклі, d > 0, i = a, i = a + d, i = a + 2 \* d және т.б. айнымалы индекстің мәндерін орнатады. i < b болатын барлық мәндер үшін. Егер d < 0 болса, онда цикл айнымалысы барлық i > b мәндерін қабылдайды.

### **Жолдық мәндермен жұмыс**

Python-да жол (string) **итерленетін объект**, сондықтан оны **for** циклінде әрбір символды жеке қарау үшін пайдалануға болады:

text = "Python"

for letter in text:

print(letter)

**Нәтиже:**

P

y

t

h

o

n

### **Қосарланған цикл (цикл ішінде цикл)**

Кейде бір циклдің ішінде екінші **for** циклі болуы мүмкін. Бұл көп өлшемді тізімдермен (матрицалар) жұмыс істеуде қолданылады.

matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

for row in matrix:

for num in row:

print(num, end=" ")

print()

**Нәтиже:**

1 2 3

4 5 6

7 8 9

### **Сандық массив элементтерінің қосындысын табу**

numbers = [10, 20, 30, 40, 50]

total = 0

for num in numbers:

total += num

print("Жиынтық мәні:", total)

Жиынтық мәні: 150

### **Тақ және жұп сандарды бөлек шығару**

for i in range(1, 11):

if i % 2 == 0:

print(f"{i} - Жұп сан")

else:

print(f"{i} - Тақ сан")

**Нәтиже:**

1 - Тақ сан

2 - Жұп сан

3 - Тақ сан

4 - Жұп сан

...

10 - Жұп сан

*1 – есеп*

1-ден N-ге дейінгі сандардың қосындысын табу

N = int(input("N санын енгізіңіз: "))

total = 0

for i in range(1, N + 1):

total += i

print("Жиынтық:", total)

*2 – есеп*

**N**-нің **1-ден 10-ға дейінгі** көбейту кестесін шығару

N = int(input("Көбейту кестесін шығару үшін сан енгізіңіз: "))

for i in range(1, 11):

print(f"{N} x {i} = {N \* i}")

3 – есеп

***N! (факториалын)*** *есептеңіз, мұнда N! = 1 \* 2 \* 3 \* ... \* N.*

*N = int(input("Факториалын есептеу үшін сан енгізіңіз: "))*

*factorial = 1*

*for i in range(1, N + 1):*

*factorial \*= i*

*print(f"{N}! =", factorial)*

**Тапсырмалар**

1. 1-ден 100-ге дейінгі сандарды шығару үшін for циклін қолданыңыз.
2. Тек жұп сандарды шығаратын for циклін жазыңыз (1-ден 50-ге дейін).
3. Пайдаланушы енгізген санның факториалын есептеу үшін for циклін қолданыңыз.
4. Берілген тізімдегі барлық элементтердің қосындысын табу үшін for циклін жазыңыз.  
   *Мысалы: numbers = [3, 7, 12, 5, 9] → Нәтиже: 36*
5. Кестелік көбейту (1-ден 10-ға дейінгі сан үшін көбейту кестесін шығару).
6. Пайдаланушы енгізген сөзді кері тәртіпте шығару (for циклін қолдана отырып).
7. 1-ден 100-ге дейінгі барлық жай сандарды (for арқылы) шығаратын код жазыңыз.

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

**Рефлексия**

**Үй тапсырмасы –** Пракикалық жұмыс №3 орындау және қорғау