Жамбыл облысы әкімдігі білім басқармасы

«Білім» кәсіби гуманитарлық-техникалық колледжі

(білім беру ұйымының атауы)

**Оқу сабағының жоспары**

(теориялық немесе өндірістік оқыту)

**Модульдермен жұмыс**

(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 модульдермен жұмыс жасауды зерттеу, екі өлшемді массивтерге есеп шығаруды жүзеге асыру, қолдану жолдарын ұйымдастыру, программаны өңдеу жұмыснегіздерін үйрету

#### **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 модульдермен жұмыс, екі өлшемді массивтермен жұмыс, есеп шығаруды жүзеге асыру, қолдану жолдарын ұйымдастыру, программаны өңдеу және пайдалану дағдыларын меңгеру.

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

Ю.Аляев, О.Козлов. Алгоритмизация и языки программирования  Python, C++, Visual Basic: Учебно-справочное пособие. – М.: Финансы и статистика, 2004

<https://docs.python.org/3/tutorial/modules.html>

***5. Сабақтың барысы: (90 минут)***

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

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

***“Алгоритм” әдісі*** *(15 минут)*

**Модульдермен жұмыс**

Python – қуатты, икемді және кеңейтілетін бағдарламалау тілі. Оның басты ерекшеліктерінің бірі – модульдік құрылымда жұмыс істеу мүмкіндігі. Python тілінде модульдер бағдарламаны бірнеше кішігірім бөліктерге бөлуге және оларды қайта қолдануға мүмкіндік береді. Бұл бағдарламаның құрылымын жақсартып, кодтың оқылуын жеңілдетеді.

**1. Python-дағы модульдер дегеніміз не?**

Модуль – бұл Python файлы, оның ішінде айнымалылар, функциялар және кластар болуы мүмкін. Python-да әрбір .py кеңейтімі бар файл модуль болып табылады. Модульдер көмегімен үлкен кодтарды бірнеше кішігірім файлдарға бөлуге және оларды басқа бағдарламаларда қолдануға болады.

**Python-дағы модульдердің артықшылықтары:**

* Кодты қайта пайдалану мүмкіндігі
* Бағдарламаның құрылымын жақсарту
* Кодтың оқылуын және жөндеуді жеңілдету
* Үлкен жобалардағы ұйымдастыруды жеңілдету

**2. Python-да модульді импорттау**

Python-да кез келген модульді импорттау үшін import түйінді сөзін қолданамыз.

**Қарапайым импорттау:**

import math # math – бұл стандартты кітапханадағы математикалық модуль

print(math.sqrt(16)) # 4.0

**Модульдегі белгілі бір функцияны импорттау:**

from math import sqrt

print(sqrt(25)) # 5.0

**Модульді қысқа атаумен импорттау:**

import numpy as np

Мұнда numpy модулі np деген қысқа атаумен қолданылады, бұл кодты ықшам әрі ыңғайлы етеді.

**Барлық функцияларды импорттау (ұсынылмайды):**

from math import \* # Барлық функцияларды импорттау

print(sin(0)) # math.sin() деп жазбай-ақ қолдануға болады

Бұл әдіс үлкен бағдарламаларда қате тудыруы мүмкін, өйткені әртүрлі модульдердегі функция атаулары қайталануы мүмкін.

**3. Өз модулімізді құру**

Python-да жеке модуль жасау өте оңай. Ол үшін .py кеңейтімі бар жаңа файл құрып, оның ішіне функциялар немесе айнымалылар жазу керек.

**Мысал:**  
mymodule.py файлын жасаймыз:

def greet(name):

return f"Сәлем, {name}!"

def add(a, b):

return a + b

Енді бұл модульді басқа Python файлына импорттауға болады:

import mymodule

print(mymodule.greet("Айжан")) # Сәлем, Айжан!

print(mymodule.add(5, 7)) # 12

Егер тек белгілі бір функцияларды импорттағымыз келсе:

from mymodule import greet

print(greet("Қайрат")) # Сәлем, Қайрат!

**4. \_\_name\_\_ == "\_\_main\_\_" және модульдерді орындалатын файл ретінде пайдалану**

Python-да әрбір модульдің өзіндік атауы бар (\_\_name\_\_). Егер модуль негізгі орындалатын файл ретінде жұмыс істесе, оның \_\_name\_\_ мәні "\_\_main\_\_" болады.

Мысал:

# mymodule.py

def greet(name):

return f"Сәлем, {name}!"

if \_\_name\_\_ == "\_\_main\_\_":

print(greet("Әлихан")) # Бұл тек тікелей іске қосылғанда орындалады

Егер mymodule.py модулін басқа файлдан импорттасақ, if \_\_name\_\_ == "\_\_main\_\_" блогы орындалмайды. Бұл негізгі кодты басқа жерлерде пайдалануға мүмкіндік береді.

**5. Python стандартты модульдері**

Python стандартты кітапханасы көптеген пайдалы модульдерді қамтиды. Кейбір танымал модульдер:

**5.1. math модулі – математикалық функциялар**

import math

print(math.factorial(5)) # 120

print(math.pow(2, 3)) # 8.0

**5.2. random модулі – кездейсоқ сандар**

import random

print(random.randint(1, 10)) # 1-ден 10-ға дейінгі кездейсоқ сан

print(random.choice(["Apple", "Banana", "Cherry"])) # Кездейсоқ элемент таңдайды

**5.3. datetime модулі – уақыт пен күнді өңдеу**

import datetime

now = datetime.datetime.now()

print(now.strftime("%Y-%m-%d %H:%M:%S")) # Ағымдағы уақыт

**5.4. os модулі – операциялық жүйе функциялары**

import os

print(os.getcwd()) # Жұмыс каталогын көрсету

os.mkdir("new\_folder") # Жаңа папка жасау

**5.5. sys модулі – жүйелік параметрлер**

import sys

print(sys.version) # Python нұсқасын көрсету

**6. Сыртқы модульдер мен кітапханалар**

Python-да көптеген сыртқы модульдер бар, оларды pip арқылы орнатуға болады.

**Мысалы, NumPy кітапханасын орнату:**

pip install numpy

Содан кейін оны импорттауға болады:

import numpy as np

arr = np.array([1, 2, 3])

print(arr)

**Тапсырмалар**

1. math модулін қолданып, берілген санның квадрат түбірін есептеу.
2. my\_module.py атты модуль құрып, екі санды қосу және көбейту функцияларын жазу.
3. random модулін қолданып, 1-ден 100-ге дейінгі кездейсоқ санды шығару және 5 кездейсоқ бүтін саннан тұратын тізім жасау.
4. datetime модулін пайдаланып, ағымдағы күн мен уақытты HH:MM:SS форматында шығару.
5. os модулі арқылы жұмыс каталогын шығару және test\_folder атты жаңа папка жасау.
6. sys модулі арқылы Python нұсқасын анықтау.
7. geometry.py модулін жасап, шеңбер мен тіктөртбұрыштың ауданын есептеу функцияларын жазу.
8. time модулін қолданып, бағдарламаны 5 секундқа кідірту.
9. converter.py модулін жасап, температураны Цельсийден Фаренгейтке айналдыру функциясын жазу.
10. calendar модулін пайдаланып, пайдаланушы енгізген жыл мен айдың күнтізбесін шығару.

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

**Рефлексия**

**Үй тапсырмасы –** Тапсырманы орындау және қорғау