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 файлдармен жұмыс жолдарын игеру.

(сабақ тақырыбы)

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

**Дайындаған педагог** Нургисаева У.М

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

**1. Жалпы мәліметтер**

Курс, оқу жылы, топ 3 курс, 3БҚ-22

Сабақ түрі: Аралас

**2. Мақсаты, міндеттері:**

#### **Оқу:** Білім алушыларға Python программалау тілін қолдана отырып, файлдармен жұмыс істеу негіздерін үйрету. Оқу процесінде файлдарды ашу, оқу, жазу және жабу әдістерін меңгерту. Сонымен қатар, Python тілінде қателерді өңдеуді және файлдардың жолдарын дұрыс басқаруды үйрету.

#### **Дамыту:** Аналитикалық ойлау қабілеттерін дамыту, файлдармен жұмыс істеу барысында кездесетін проблемаларды шешу тәсілдерін үйрету. Бұл процессте ақпаратты құрылымдау, жүйелеу және нақты шешім қабылдай білу дағдыларын жетілдіру.

#### **Тәрбиелік:** Еңбекке жауапкершілікпен қарауды, жұмысты тиімді ұйымдастыруды және деректерді қауіпсіз өңдеуді тәрбиелеу. Олар ақпаратты дұрыс сақтау мен өңдеу арқылы деректердің құпиялылығын сақтау мәдениетін меңгеру керек.

#### **3. Оқу-жаттығу процесінде білім алушылар меңгеретін күтілетін нәтижелер және кәсіби дағдылар тізбесі:** Файлды ашу, оқу, жазу және жабу әдістерін меңгеру. Файлдарды тексттік және бинарлық режимде өңдеу дағдыларын игеру. Python тілінде файлмен жұмыс істегенде кездесетін қателерді анықтау және оларды тиімді өңдеу.

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

***5. Сабақтың барысы: (90 минут)***

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

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

***“Менен сұрақ - Сізден жылдам жауап” әдісі*** *(15 минут)*

**Python (Пайтон) программалау тілінде файлдармен жұмыс істеу**

Python бағдарламалау тілінде файлдармен жұмыс істеу — бұл мәліметтерді сақтау және қайта алу үшін маңызды қабілет. Файлдармен жұмыс істеу арқылы пайдаланушының қажетті деректерді сақтау, оқу, өңдеу және жазу процесін жүзеге асыруға болады. Python тілінде файлдармен жұмыс істеу оңай әрі тиімді, өйткені оның стандартты кітапханасы файлдармен жұмыс істеуге қажетті барлық құралдарды ұсынады.

**1. Файлды ашу**

Python тілінде файлмен жұмыс бастау үшін алдымен оны ашу қажет. Бұл үшін Python-ның стандартты open() функциясы қолданылады. Файлды ашу үшін екі параметр пайдаланылады:

* **file\_name** — бұл ашылатын файлдың аты.
* **mode** — файлды ашу режімі (оқу, жазу, қосымша және т.б.).

**open() функциясының параметрлері:**

* "r" — тек оқу режимі (файл тек оқылады).
* "w" — жазу режимі (егер файл бұрыннан бар болса, ол өшіріліп, жаңа файл жазылады).
* "a" — қосу режимі (файлға мәлімет қосылады, егер файл жоқ болса, жаңа файл құрылады).
* "rb" — бинарлық режимде оқу (мысалы, бейнемазмұнды оқу).
* "wb" — бинарлық режимде жазу.
* "x" — файлды ашқанда ғана құру (егер файл бар болса, қате шығады).

python

file = open("example.txt", "r") # example.txt файлын оқу режимінде ашу

**2. Файлдан оқу**

Python файлынан оқу үшін бірнеше тәсілді пайдалану мүмкін:

* read() — файлдың барлық мазмұнын бірден оқып шығуға мүмкіндік береді.
* readline() — файлды жолмен оқиды, әр шақыруда келесі жолды қайтарады.
* readlines() — файлды жолдар бойынша оқиды және оларды тізім түрінде қайтарады.

**Мысал:**

python

# Файлды ашып, оның мазмұнын оқу

file = open("example.txt", "r")

content = file.read() # Барлық файл мазмұнын оқу

print(content)

file.close() # Файлды жапқан дұрыс

**3. Файлға жазу**

Python файлға жазу үшін write() немесе writelines() әдістерін пайдаланады.

* write() — файлға бір жолды жазады.
* writelines() — бірнеше жолды бірден жазады.

Егер файлды "w" немесе "a" режимінде ашсаңыз, онда файлға жазылған деректер автоматты түрде файлға қосылады.

**Мысал:**

python

# Файлды ашып, оған жазу

file = open("example.txt", "w")

file.write("Бұл жаңа жол.\n")

file.write("Python тілінде файлдармен жұмыс істеп жатырмыз.")

file.close() # Файлды жапқан дұрыс

**4. Файлды жабу**

Файлды оқығаннан немесе жазғаннан кейін міндетті түрде жабу қажет. Бұл ресурстарды босату үшін және файлдың дұрыс сақталуын қамтамасыз ету үшін маңызды. Бұл үшін close() әдісі қолданылады.

file = open("example.txt", "r")

content = file.read()

file.close() # Файлды жабу

**5. Файлмен жұмыс істегенде контекстті менеджерді пайдалану**

Python тілінде файлдарды ашу мен жабу процесін жеңілдету үшін контекстті менеджерлерді (with операторы) қолдануға болады. Бұл файлды ашқанда оның автоматты түрде жабылуын қамтамасыз етеді. Бұл әсіресе үлкен жобаларда немесе бірнеше файлмен жұмыс істегенде пайдалы.

**Мысал:**

python

# Контекстті менеджерді пайдаланып файлды ашу

with open("example.txt", "r") as file:

content = file.read()

print(content)

# Файл автоматты түрде жабылады

**6. Файлмен жұмыс істеу кезінде қателермен жұмыс**

Файлдармен жұмыс істегенде қателер туындауы мүмкін, мысалы файл жоқ, файлды ашу мүмкін емес және т.б. Python тілінде қателерді өңдеу үшін try-except блоктарын қолдануға болады.

**Мысал:**

python

try:

file = open("non\_existent\_file.txt", "r")

except FileNotFoundError:

print("Файл табылмады!")

**7. Бинарлық файлдармен жұмыс**

Python бинарлық файлдармен де жұмыс істей алады. Бұл бейнемазмұнды немесе басқа деректерді өңдеуге пайдалы. Бинарлық файлдарды ашу үшін "rb" (оқу үшін) және "wb" (жазу үшін) режимдері қолданылады.

**Мысал:**

python

# Бинарлық файлды оқу

with open("image.jpg", "rb") as file:

binary\_data = file.read()

print(binary\_data[:20]) # Бастапқы 20 байтты шығару

# Бинарлық файлға жазу

with open("new\_image.jpg", "wb") as file:

file.write(binary\_data)

**8. Файлдардың қозғалысын бақылау**

Файлмен жұмыс істегенде файлдың жолын анықтау, оның бар-жоғын тексеру және басқа да ақпараттар маңызды болады. Python тілінде бұл операцияларды орындау үшін os және shutil модульдері қолданылады.

python

import os

# Файл бар-жоғын тексеру

if os.path.exists("example.txt"):

print("Файл бар.")

else:

print("Файл жоқ.")

# Файлды көшіру

import shutil

shutil.copy("example.txt", "backup\_example.txt")

Python тілінде файлдармен жұмыс істеу өте ыңғайлы және қарапайым. Файлдарды оқу, жазу және оларды өңдеу үшін әртүрлі әдістер мен құралдар бар, және оларды пайдалану арқылы ақпаратты тиімді басқаруға болады. Файлдармен жұмыс істегенде қауіпсіздік талаптарына және файлдарды дұрыс жабуға ерекше назар аудару керек. Python-ның бай кітапханасы осы жұмысты жеңілдетіп, бағдарламашыларға жоғары деңгейде өнімді жұмыс істеуге мүмкіндік береді.

file = open('C:\\Users\\Acer\\OneDrive\\Рабочий стол\\3БҚ\\ff.txt', 'r')

content = file.read()

print(content)

file.close()