**ДВНЗ «ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»**

**Факультет комп’ютерно-інформаційних технологій та автоматизації**

**ЗВІТ до практичної РОБОТИ №4**

з дисципліни «Методи та системи штучного інтелекту»

за темою: «РОБОТА З ФАЙЛАМИ ТА ОРГАНІЗАЦІЯ ПРОЕКТІВ»

Варіант №7

Виконав:

Студент гр. КІ-21

Іван ДАЦЕНКО

Перевірив:

Андрій НІКІТЕНКО

Луцьк – 2024

**Мета роботи:** вивчити та засвоїти базові принципи роботи з файлами,модулями та пакетами. Ознайомитися з існуючими режимами доступу, а також з особливостями використання різних методів і функцій для роботи з файлами. Дізнатися про особливості та способи підключення модулів та організації пакетів.

**Завдання**

1) Для довідника реалізованого згідно власного варіанту у практичній роботі №3, створити програму з підтримкою функцій роботи з файлами, а саме:

 Збереження довідника до файлу.

 Читання довідника з файлу.

 Збереження змін у довіднику.

Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

2) У даній практичній роботі необхідно виконати розбиття функцій з практичної роботи №2 на модулі з наступним об'єднанням отриманих модулів водин загальний пакет. Пакет необхідно підключити в окремому файлі програмиі продемонструвати роботу з кожним модулем пакета.

Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

**Лістинг програми:**

**1 завдання - Lab4.py**

# to file  
def save\_to\_file(blogger\_list):  
 with open("blogger\_data.txt", "w") as file:  
 for blogger in blogger\_list:  
 file.write(f"{blogger}\n")  
  
# from file  
def read\_from\_file():  
 try:  
 with open("blogger\_data.txt", "r") as file:  
 blogger\_list = [eval(line.strip()) for line in file]  
 return blogger\_list  
 except FileNotFoundError:  
 return []  
  
# blg template  
blogger = {}.fromkeys(['nickname', 'channel\_name', 'link', 'age', 'subscribers'])  
blogger\_list = []  
blogger\_list = read\_from\_file()  
  
#add  
def add\_blogger(blogger\_list):  
 new\_blogger = blogger.copy()  
 new\_blogger['nickname'] = input("Enter nickname: ")  
 new\_blogger['channel\_name'] = input("Enter channel name: ")  
 new\_blogger['link'] = input("Enter link: ")  
  
 while True:  
 try:  
 new\_blogger['age'] = int(input("Enter age: "))  
 if new\_blogger['age'] < 0:  
 raise ValueError("Age cannot be negative.")  
 break  
 except ValueError:  
 print("Invalid input. Please enter a valid age.")  
  
 while True:  
 try:  
 new\_blogger['subscribers'] = int(input("Enter number of subscribers: "))  
 if new\_blogger['subscribers'] < 0:  
 raise ValueError("Subscribers count cannot be negative.")  
 break  
 except ValueError:  
 print("Invalid input. Please enter a valid number of subscribers.")  
 blogger\_list.append(new\_blogger)  
  
# delete  
def delete\_blogger(blogger\_list):  
 nickname = input("Enter the nickname of the blogger to delete: ")  
 for blogger in blogger\_list:  
 if blogger['nickname'] == nickname:  
 blogger\_list.remove(blogger)  
 print("Blogger deleted successfully.")  
 return  
 print("Blogger not found.")  
  
# edit  
def edit\_blogger(blogger\_list):  
 nickname = input("Enter the nickname of the blogger to edit: ")  
 for blogger in blogger\_list:  
 if blogger['nickname'] == nickname:  
 print("1. Edit nickname")  
 print("2. Edit channel name")  
 print("3. Edit link")  
 print("4. Edit age")  
 print("5. Edit number of subscribers")  
 choice = input("Enter your choice: ")  
 if choice not in ['1', '2', '3', '4', '5']:  
 print("Invalid choice.")  
 return  
  
 if choice == '1':  
 blogger['nickname'] = input("Enter new nickname: ")  
 elif choice == '2':  
 blogger['channel\_name'] = input("Enter new channel name: ")  
 elif choice == '3':  
 blogger['link'] = input("Enter new link: ")  
 elif choice == '4':  
 while True:  
 try:  
 new\_age = int(input("Enter new age: "))  
 if new\_age < 0:  
 raise ValueError("Age cannot be negative.")  
 blogger['age'] = new\_age  
 break  
 except ValueError:  
 print("Invalid input. Please enter a valid age.")  
 elif choice == '5':  
 while True:  
 try:  
 new\_subscribers = int(input("Enter new number of subscribers: "))  
 if new\_subscribers < 0:  
 raise ValueError("Subscribers count cannot be negative.")  
 blogger['subscribers'] = new\_subscribers  
 break  
 except ValueError:  
 print("Invalid input. Please enter a valid number of subscribers.")  
 print("Blogger edited successfully.")  
 return  
 print("Blogger not found.")  
  
  
# show all blg  
def display\_all\_bloggers(blogger\_list):  
 for blogger in blogger\_list:  
 print(  
 f"Nickname: {blogger['nickname']}, Channel Name: {blogger['channel\_name']}, Link: {blogger['link']}, Age: {blogger['age']}, Subscribers: {blogger['subscribers']}")  
  
# sorted blg  
def top\_n\_oldest\_bloggers(blogger\_list, n):  
 if n <= 0:  
 print("Invalid input for N. Please enter a positive integer.")  
 return  
 sorted\_bloggers = sorted(blogger\_list, key=lambda x: x['age'])  
 print(f"Top {n} oldest bloggers:")  
 for i, blogger in enumerate(sorted\_bloggers[:n]):  
 print(f"{i+1}. {blogger['nickname']} (Age: {blogger['age']})")  
  
# menu  
while True:  
 print("Menu")  
 print("1. Add a blogger")  
 print("2. Delete a blogger")  
 print("3. Edit a blogger")  
 print("4. Display all bloggers")  
 print("5. Display top N oldest bloggers by age")  
 print("0. Exit")  
 choice = input("Enter your choice: ")  
  
 try:  
 choice = int(choice)  
 except ValueError:  
 print("Invalid choice. Please enter a number between 0 and 5.")  
 continue  
  
 if choice == 1:  
 add\_blogger(blogger\_list)  
 elif choice == 2:  
 delete\_blogger(blogger\_list)  
 elif choice == 3:  
 edit\_blogger(blogger\_list)  
 elif choice == 4:  
 display\_all\_bloggers(blogger\_list)  
 elif choice == 5:  
 n = input("Enter the value of N: ")  
 try:  
 n = int(n)  
 top\_n\_oldest\_bloggers(blogger\_list, n)  
 except ValueError:  
 print("Invalid input for N. Please enter a valid integer.")  
 elif choice == 0:  
 break  
 else:  
 print("Invalid choice. Please enter a number between 0 and 5.")

**2 завдання – main.py, \_\_init\_\_.py, module1.py, module2.py, module3.py.**

**main.py**

from module1 import find\_max\_positive  
from module2 import my\_sort  
from module3 import compute\_f  
  
my\_array = [1, -2, 3, -4, 5]  
result = find\_max\_positive(my\_array)  
print(f"Maximum positive element: {result}")  
  
my\_array\_to\_sort = [7, 2, 4, 5, 9, 0, 1, 8, 6, 3]  
my\_sort(my\_array\_to\_sort)  
print("Sorted array:", my\_array\_to\_sort)  
  
result = compute\_f(2, 3, 4)  
print("Result of computing function f:", result)

**\_\_init\_\_.py**

\_\_all\_\_ = ["module1", "module2", "module3", "main"]

**module1.py**

def find\_max\_positive(arr):  
 positive\_numbers = [num for num in arr if num > 0]  
 if not positive\_numbers:  
 return None  
 return max(positive\_numbers)

**module2.py**

def my\_sort(arr):  
 n = len(arr)  
 for i in range(n):  
 for j in range(0, n - i - 1):  
 if arr[j] > arr[j + 1]:  
 arr[j], arr[j + 1] = arr[j + 1], arr[j]

**module3.py**

compute\_f = lambda x, y, z: 7 \* x \* (y \*\* 3) \* (2 \* z)

**Результати:**

Зображення, що містить текст, знімок екрана

Автоматично згенерований описЗображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

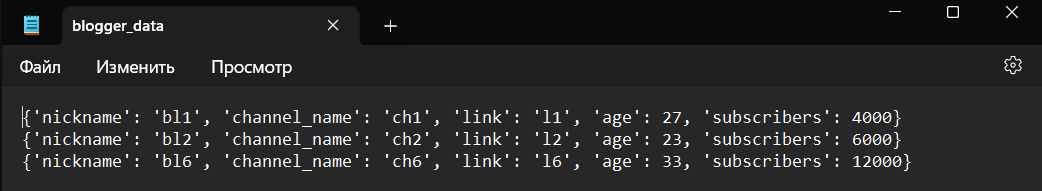


Рисунок 1 − Результати виконання першого завдання

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

Рисунок 2 − Результат виконання другого завдання

**Висновок:** У цій практичній роботі я вивчив та засвоїв базові принципи роботи з файлами, модулями та пакетами. Ознайомився з існуючими режимами доступу, а також з особливостями використання різних методів і функцій для роботи з файлами. Дізнався про особливості та способи підключення модулів та організації пакетів.