

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ  
Циклова комісія інформаційних технологій

КУРСОВИЙ ПРОЕКТ  
(РОБОТА)

Програмування

(назва дисципліни)

на тему:

аналіз даних з соціальних медіа

Студента IV курсу 407-АКІТ групи  
спеціальності 151 Автоматизація та комп'ютерно-  
інтегровані технології

спеціалізації 5.151.1 Обслуговування  
інтелектуальних інтегрованих систем

Горобець М.О.

Керівник: викладач **Васильєв М.В.**

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_ (Васильєв М.В.)

(підпис)

\_\_\_\_\_ (\_\_\_\_\_)

(підпис)

\_\_\_\_\_ (\_\_\_\_\_)

(підпис)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ**

**РОЗГЛЯНУТО**

на засіданні циклової комісії  
Голова циклової комісії  
Марина ВЕЛИЧКО  
«31» серпня 2023р.

**ЗАТВЕРДЖЕНО**

Зав. відділення  
Олеся ТВЕРДОХЛІБОВА  
«31» серпня 2023 р.

**Завдання**

на курсовий проєкт студенту гр. 407-АКІТ  
спеціальність/освітньо-професійна програма 151 Автоматизація та  
комп'ютерно-інтегровані технології/5.151.1 Обслуговування  
інтелектуальних інтегрованих систем

***Горобцю Максиму Олександровичу***

Тема: Аналіз соціальних медіа.

Термін виконання роботи з 18.11.2023 р. по 4.04.2024 р.

Вхідні дані для проектування: (Варіант №5)

Розробка програми для аналізу даних з соціальних медіа. Для зберігання інформації використовується CSV-файл, де міститься інформація про публікації, лайки, коментарі та репости. Програма повинна проводити аналіз популярності та взаємодії з контентом.

**Література та посібники для проектування:**

1. Head-First Python, 2nd edition: Paul Barry. - Sebastopol, California, U.S.: O'Reilly Media, 2016. – 622 с.
2. Think Python: How to Think Like a Computer Scientist, 2nd edition: Allen B. Downey. - Sebastopol, California, U.S.: O'Reilly Media, 2015. – 292 с.
3. Clean Code: A Handbook of Agile Software Craftsmanship: Robert C. Martin. - London, England: Pearson, 2008. – 464 с.
4. Python.org: веб-сайт. URL: <https://www.python.org> (дата звернення: 02.01.2023)
5. Python Tutorial: веб-сайт. URL: <https://www.w3schools.com/python/> (дата звернення: 02.01.2023)
6. Learn to become a modern Python developer: веб-сайт. URL: <https://roadmap.sh/python/> (дата звернення: 02.01.2023)

### Зміст розрахунково-пояснювальної записки

№ п/п	Зміст	Планований термін виконання	Фактичний термін виконання	%
1.	Вступ	22.11.2023	22.11.2023	5
2.	Аналіз задачі, засобів та методів її вирішення	30.11.2023	30.11.2023	15
3.	Проектування загального алгоритму роботи програми	18.01.2024	18.01.2024	40
4.	Розробка програмного забезпечення	01.03.2024	01.03.2024	80
5.	Керівництво користувача	14.03.2024	14.03.2024	90
6.	Висновки	28.03.2024	28.03.2024	95
7.	Список використаних джерел	02.04.2024	02.04.2024	100

Керівник курсової роботи \_\_\_\_\_ Микола ВАСИЛЬЄВ

Завдання до курсової роботи

Одержав(ла) студент(ка) гр. 407-АКІТ \_\_\_\_\_ (\_\_\_\_\_)

Дата «18» листопада 2023 р.

## Реферат

Пояснювальна записка: 39 сторінок, 20 рисунків, 4 використаної літератури.

Ключові слова: ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, PYTHON, CSV, АНАЛІЗ.

Об'єктом розробки є програмне забезпечення для аналізу даних з соціальних медіа.

Мета розробки – розробка програмного забезпечення.

У процесі роботи проведено розробку блок-схеми алгоритму роботи програми та окремих функцій, розроблено програму, яка виконує алгоритм роботи.

Основні конструктивні та техніко-економічні показники: висока надійність, зручність, простота, масштабованість.

## **Перелік використаних скорочень**

- ООП – Об'єктно-орієнтоване програмування
- ПЗ – програмне забезпечення
- CSV – Comma-separated values
- VS Code – Visual Studio Code

## ЗМІСТ

Реферат .....	2
Перелік використаних скорочень.....	3
ВСТУП.....	5
1. АНАЛІЗ ЗАДАЧІ, ЗАСОБІВ ТА МЕТОДІВ ЇЇ ВИРІШЕННЯ .....	7
1.1 Розуміння вимог .....	7
1.2 Аналіз вихідних даних .....	8
1.3 Вибір технологій.....	10
2. ПРОЕКТУВАННЯ ЗАГАЛЬНОГО АЛГОРИТМУ РОБОТИ ПРОГРАМИ	13
3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	18
3.1 Розробка головного класу Post.....	18
3.2 Класи для тестування .....	21
3.3 Клас CSV_Reader.....	22
3.4 Клас Bcolors .....	24
3.5 Сортування та клас Sorter .....	25
3.6 Основний файл виконання програми .....	26
4. КЕРІВНИЦТВО КОРИСТУВАЧА .....	28
4.1 Основний функціонал .....	28
4.3 Імпортування файлу .....	30
4.4 Робота з файлом.....	31
ВИСНОВКИ .....	33
ЛІТЕРАТУРА.....	34
ДОДАТКИ.....	35
Додаток А.....	35
Додаток Б.....	36
Додаток В .....	39

					<i>5.151.1.VK3-KП</i>		
Змн.	Арк.	№ докум.	Підпис	Дата	<p style="text-align: center;"><b>Пояснювальна записка</b></p>		
Розроб.		Горобець М.					
Перевір.		Васильєв М.					
					Літ.	Арк.	Акрушів
						4	39
					<p style="text-align: center;">ХПФК Група 407</p>		

## ВСТУП

У наш час програмування є однією з найбільш динамічно розвиваючихся галузей, що відображає світові тенденції технологічного прогресу та потреби суспільства. Однією з найактуальніших галузей програмування є аналіз даних з соціальних медіа. Соціальні платформи, такі як Facebook, Instagram, Twitter та інші, стали не просто майданчиками для спілкування, але й важливим інструментом для бізнесу, політики, науки та культури. Вони об'єднують мільйони користувачів з усього світу, дозволяючи їм обмінюватися інформацією, думками та переживаннями.

Актуальність розробки програмного забезпечення для аналізу даних з соціальних медіа визначається не лише через загальний розвиток цих платформ, але й через необхідність ефективного управління контентом та взаємодії з аудиторією. Компанії, маркетологи, політики та дослідники активно використовують дані з соціальних медіа для аналізу тенденцій, прогнозування поведінки аудиторії, а також для створення та вдосконалення своїх стратегій.

Метою нашої роботи є розробка програми для аналізу даних з соціальних медіа, спрямованої на вивчення популярності та взаємодії з контентом. Важливими завданнями програми є виявлення ключових трендів, аналіз залученості аудиторії та визначення ефективності контенту.

Галузь застосування програми включає маркетингові дослідження, наукові аналізи, стратегії управління контентом та багато іншого. Розробка стане корисним інструментом для студентів, дослідників та фахівців у сфері соціальних медіа, сприяючи їхньому подальшому професійному розвитку та науковим досягненням.

					5.151.1.VK3-KП	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

Програмне забезпечення було створене в Visual Studio Code, інструменті для розробки, редагування та налагодження сучасних веб-застосунків і програм для хмарних систем. Visual Studio Code доступний безкоштовно для платформ Windows, Linux і OS X. Microsoft презентувала Visual Studio Code у квітні 2015 на конференції Build 2015 як перший кросплатформний продукт у лінійці Visual Studio.

Python - інтерпретована, об'єктно-орієнтована мова програмування з високорівневою синтаксичною структурою. Це мова загального призначення, яка найчастіше використовується для розробки веб-застосунків, наукових досліджень, штучного інтелекту та інших задач. Python має простий синтаксис, що дозволяє розробникам швидко писати чистий та зрозумілий код. Він підтримує модульність та розширюваність, що дозволяє легко інтегрувати з іншими мовами та бібліотеками.

					5.151.1.BK3-КП	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		



# 1. АНАЛІЗ ЗАДАЧІ, ЗАСОБІВ ТА МЕТОДІВ ЇЇ ВИРІШЕННЯ

## 1.1 Розуміння вимог

### Вхідні дані:

Програма для аналізу даних з соціальних медіа призначена для роботи зі структурованими даними у форматі CSV. Вона приймає дані через командний рядок де користувачем вказується шлях до відповідного файлу. Передбачається, що формат CSV-файлу має стандартизовану структуру, де зберігається інформація про публікації в соціальних мережах, таку як текст публікації, кількість лайків, коментарів та репости.

### Обробка даних:

Після отримання вхідних даних у форматі CSV програма проводить їх обробку для подальшого аналізу. Цей процес включає зчитування даних з файлу та їх трансформацію у об'єкти класу. Після завершення обробки даних формується масив, що містить інформацію про кожну публікацію, включаючи текст, кількість лайків, коментарів та репости.

### Аналіз популярності:

Одним із основних функціональних елементів програми є аналіз популярності контенту, що дозволяє визначити відгуки аудиторії на публікації. Програма використовує метрики, такі як загальна кількість взаємодій (лайки, коментарі, репости разом) або сортування постів для оцінки популярності контенту. Такий аналіз дозволяє користувачеві краще розуміти, який контент найбільш цікавий для аудиторії та як взаємодіється з нею.

					5.151.1.BK3-KП	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1.2 Аналіз вихідних даних

CSV-файл містить дані про різні публікації, що є повідомленнями або оголошеннями на платформі соціальних медіа чи іншому веб-сайті. Ось основні стовпчики що буде мати CSV-файл:

1. **Назва публікації (Post title):** Це перше поле в кожному рядку і містить заголовок або опис публікації.
2. **Лайки (Likes):** Це кількість користувачів, які виразили своє задоволення публікацією, натиснувши кнопку "лайк".
3. **Коментарі (Comments):** Це кількість коментарів, які були залишені під публікацією.
4. **Репости (Repasts):** Це кількість разів, коли публікація була поділена іншими користувачами на їх власних сторінках чи профілях.
5. **Відвідування профілю (Profile visits):** Це кількість разів, коли користувачі переходили на профіль автора публікації, можливо, після перегляду публікації.

Ці дані можуть бути використані для аналізу популярності публікацій, їхньої взаємодії з аудиторією та впливу на сторінку або профіль у соціальних мережах. Наприклад, високі цифри лайків та репостів можуть свідчити про успішні публікації, які викликали зацікавленість аудиторії. Коментарі також можуть дати уявлення про взаємодію з аудиторією та відгуки на контент.

### Загальні відомості про CSV:

CSV (Comma-Separated Values) - це текстовий формат для зберігання даних у вигляді таблиці, де кожен рядок представляє один запис, а кожний стовпець в цьому записі містить значення різних полів, розділені комами або іншими розділовими знаками. CSV файл може мати розширення .csv.

Принцип роботи CSV файлу досить простий:

					5.151.1.BK3-KП	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

1. **Розділення даних:** Дані у CSV файлі зазвичай розділяються комами, хоча іноді використовуються інші розділові знаки, такі як крапки з комою або табуляція.
2. **Рядки і стовпці:** Кожен рядок у CSV файлі відповідає окремому запису або рядку даних, а кожен стовпець в рядку містить певну інформацію, яка відноситься до цього запису. Наприклад, якщо CSV файл містить дані про співробітників, кожен рядок може містити прізвище, ім'я, посаду тощо, а кожен стовпець у рядку буде відповідати конкретному полю (прізвище, ім'я, посада тощо).
3. **Кодування даних:** Дані у CSV файлах можуть бути закодовані в різних форматах, таких як UTF-8 або ASCII, залежно від потреби і використаних програм.
4. **Заголовок:** У деяких випадках перший рядок CSV файлу може містити заголовки стовпців, які надають інформацію про те, які дані містяться в кожному стовпці. Це полегшує читання та обробку даних.
5. **Зберігання даних:** CSV файли легко зберігаються у текстовому форматі і можуть бути відкриті та редаговані з використанням різноманітних програм, включаючи електронні таблиці, текстові редактори та спеціалізовані програми для обробки даних.

CSV файли широко використовуються для обміну даними між програмами, так як вони є простими у використанні та зрозумілими як для комп'ютерів, так і для людей.

					5.151.1.ВКЗ-КП	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

### 1.3 Вибір технологій

Python - це високорівнева, інтерпретована мова програмування загального призначення з простим синтаксисом та динамічною типізацією.

#### Загальна інформація:

Python був розроблений Гвідо ван Россумом та вперше випущений у 1991 році. Він швидко став популярним завдяки своїй простоті та читабельності коду. Python підтримує багато парадигм програмування, таких як процедурне, об'єктно-орієнтоване, функціональне та аспектно-орієнтоване програмування.

#### Основні особливості Python:

1. **Простота читання та написання коду:** Python має чітку та лаконічну синтаксичну структуру, що дозволяє легко читати та розуміти код.
2. **Динамічна типізація:** Змінні в Python не пов'язані з конкретними типами даних під час оголошення, а типи визначаються автоматично під час виконання програми.
3. **Широкий спектр бібліотек та фреймворків:** Python має велику кількість стандартних бібліотек та сторонніх модулів, які полегшують розробку різноманітних проектів.
4. **Портативність:** Python працює на різних операційних системах, таких як Windows, macOS, Linux та інших.
5. **Швидкість розробки:** Благодаря простоті та потужним інструментам, розробка програм на Python зазвичай займає менше часу порівняно з іншими мовами програмування.
6. **Широке застосування:** Python використовується у багатьох галузях, включаючи веб-розробку, наукові обчислення, штучний інтелект, обробку даних, машинне навчання та багато інших.

					5.151.1.ВКЗ-КП	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

### Сторонні бібліотеки:

У цьому проєкті використовується лише модуль `operator` та функція `attrgetter`. Модуль `operator` є частиною стандартної бібліотеки Python і містить функції, які дозволяють виконувати операції з типовими операторами. Функція `attrgetter` з цього модулю використовується для отримання значень атрибутів об'єктів, що дозволяє здійснювати сортування за цими значеннями.

Це рішення обрано для того, щоб зберегти проєкт простим та не додавати надмірної складності за рахунок використання сторонніх бібліотек.

### Редактор коду:

Як редактор коду, в цьому проєкті був обраний Microsoft Visual Studio Code. **Visual Studio Code (VS Code)** - це безкоштовний редактор коду, який надає широкі можливості для розробки програмного забезпечення. Дане середовище розробки засноване на платформі Electron, що дозволяє використовувати вбудований веб-браузер і забезпечує можливості розширення за допомогою плагінів. VS Code підтримує різні мови програмування, включаючи Python, та надає багато функцій, які сприяють продуктивності розробника.

Де використовують Visual Studio Code:

- Розробка веб-застосунків (Frontend та Backend).
- Розробка мобільних додатків.
- Розробка хмарних додатків.
- Розробка штучного інтелекту, машинного навчання та аналізу даних.
- Робота з системами керування версіями (наприклад, Git).
- Робота з Docker і Kubernetes.

					5.151.1.BK3-КП	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

### Плюси Visual Studio Code:

1. **Безкоштовність:** VS Code є безкоштовним для використання та має відкритий вихідний код.
2. **Підтримка різних мов програмування:** Включаючи Python, JavaScript, TypeScript, C++, Java, і багато інших.
3. **Розширюваність:** За допомогою розширень можна розширити функціональність редактора.
4. **Широкі можливості налаштування:** Велика кількість налаштувань та можливість налаштування редактора під власні потреби.
5. **Зручний інтерфейс користувача:** Інтуїтивно зрозумілий та легкий у використанні інтерфейс.

					5.151.1.BK3-КП	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2. ПРОЕКТУВАННЯ ЗАГАЛЬНОГО АЛГОРИТМУ РОБОТИ ПРОГРАМИ

Для зручності рекомендується розбити роботу на два етапи.

**Перший етап** – створення коду, який буде переводити кожен рядок CSV-файлу в об'єкт класу Experiment для подальшої роботи.

У цьому етапі розроблюється клас CSV\_Reader, який матиме два методи:

- Метод імпортування, що приймає задане користувачем ім'я файлу та відкриває його стандартними засобами Python.
- Метод, який розділяє дані, що знаходяться в файлі, та одразу заносить їх до ініціалізатора об'єкта класу Post.

**Другий етап** – сортування масиву перетворених об'єктів та виведення їх в консоль.

На другому етапі необхідно здійснити сортування об'єктів за їх атрибутами, наприклад, за датою або іншими параметрами. Після сортування об'єкти можуть бути виведені в консоль або оброблені іншим чином, залежно від потреб програми.

Ці два етапи дозволяють розділити роботу на більш зручні компоненти, що полегшує процес розробки та збереження коду в майбутньому.

### План першого етапу:

Нижче наведені блок-схеми роботи алгоритмів імпортування CSV файлу та переведення його в об'єкти класу Post, що дасть змогу полегшити роботу з сортуванням кожного запису або посту соціальних мереж та обчислювати загальний критерій зацікавленості аудиторії.

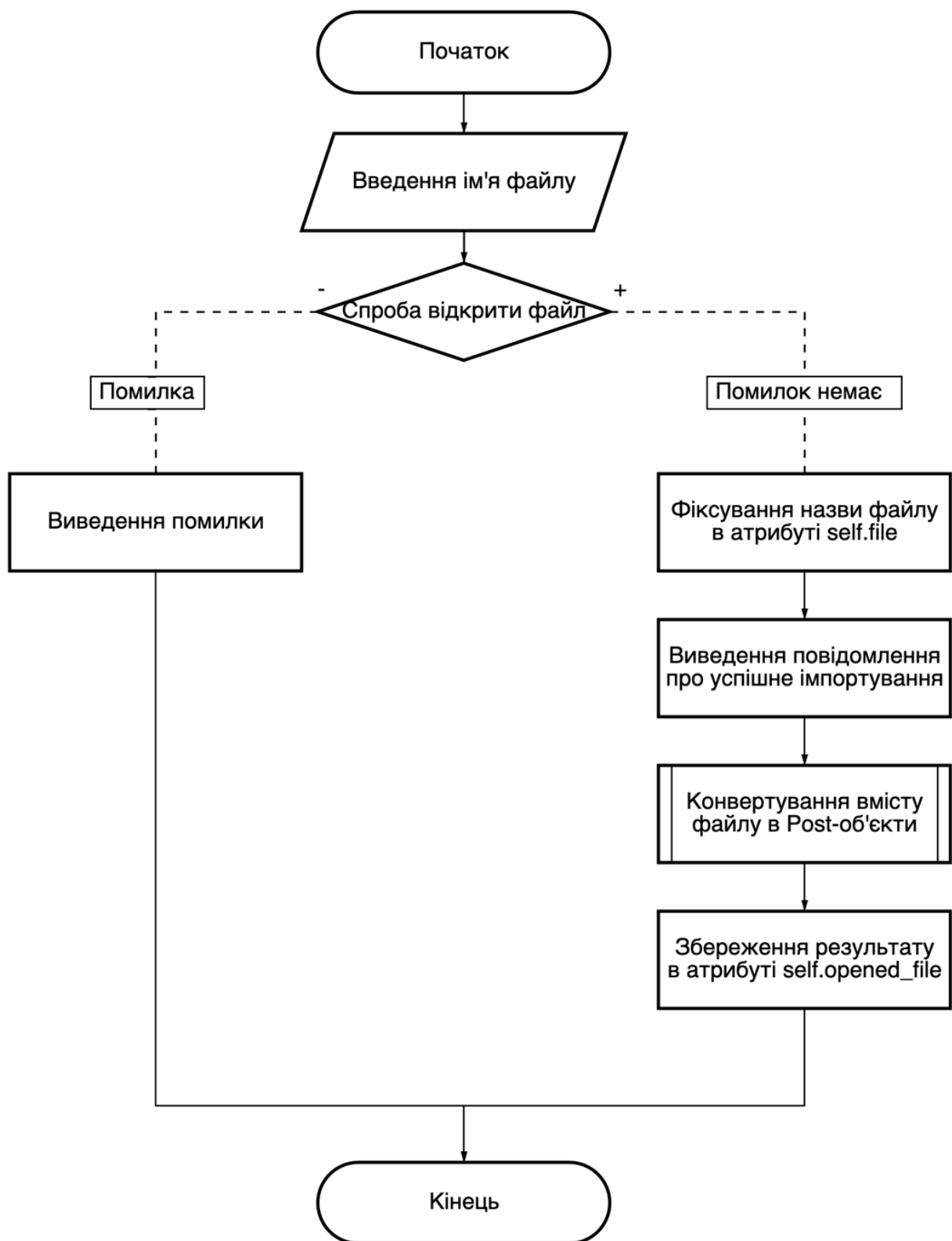


Рис. 2.1 – Схема функції імпорту файлу.



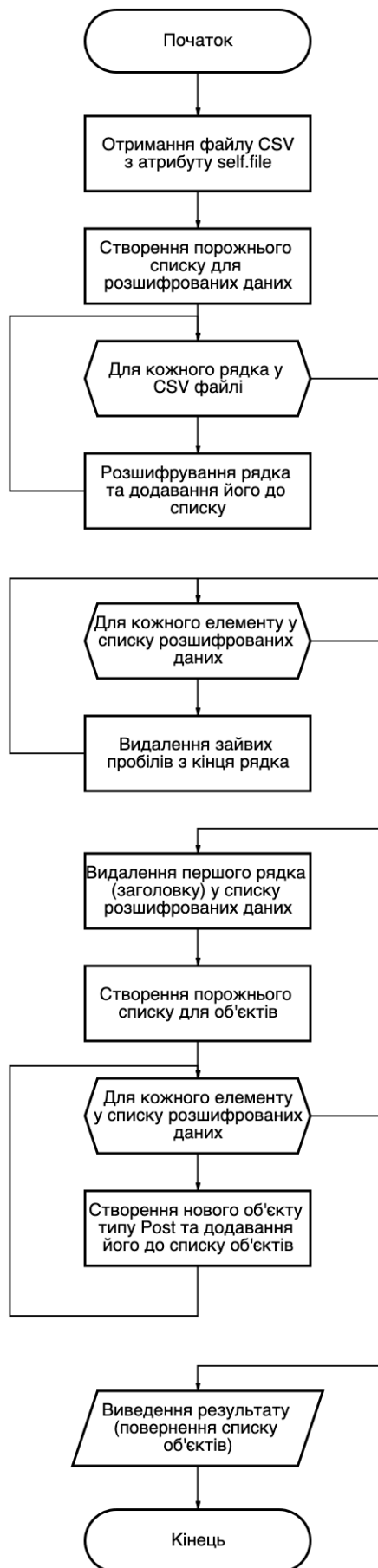


Рис. 2.2 – Схема методу переводу даних CSV файлу в об'єкти Post.

Змн.	Арк.	№ докум.	Підпис	Дата

5.151.1.BK3-KП

Арк.

15

## План другого етапу:

Нижче наведені блок-схеми роботи алгоритмів сортування об'єктів класу Post та виведення загального значення популярності посту, це допоможе мати більш краще уявлення про взаємодію користувачів з постом.

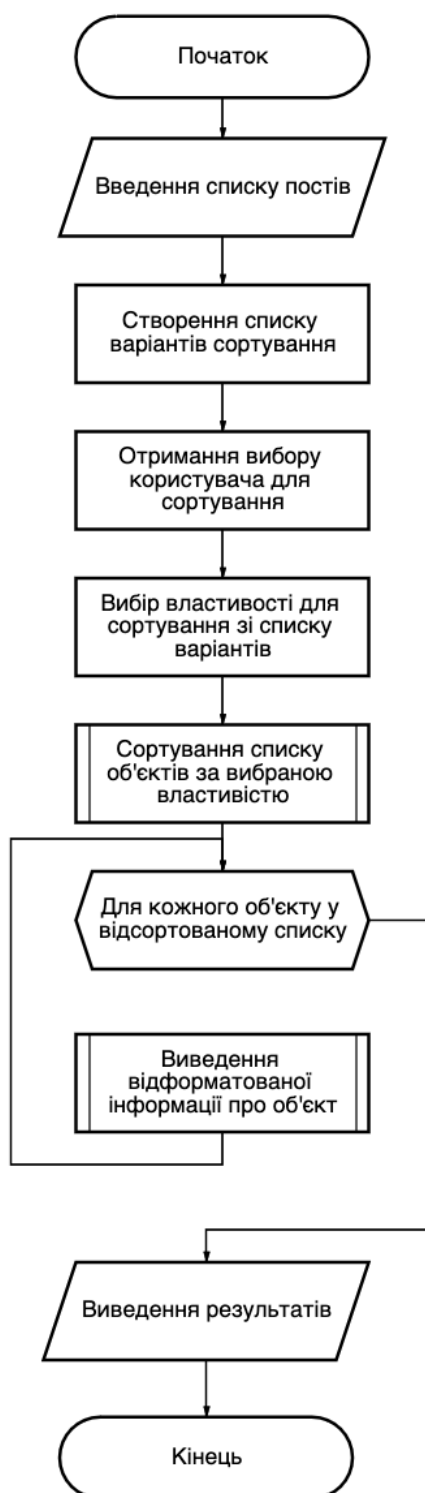


Рис. 2.3 – Схема методу сортування об'єктів Post.

Змн.	Арк.	№ докум.	Підпис	Дата

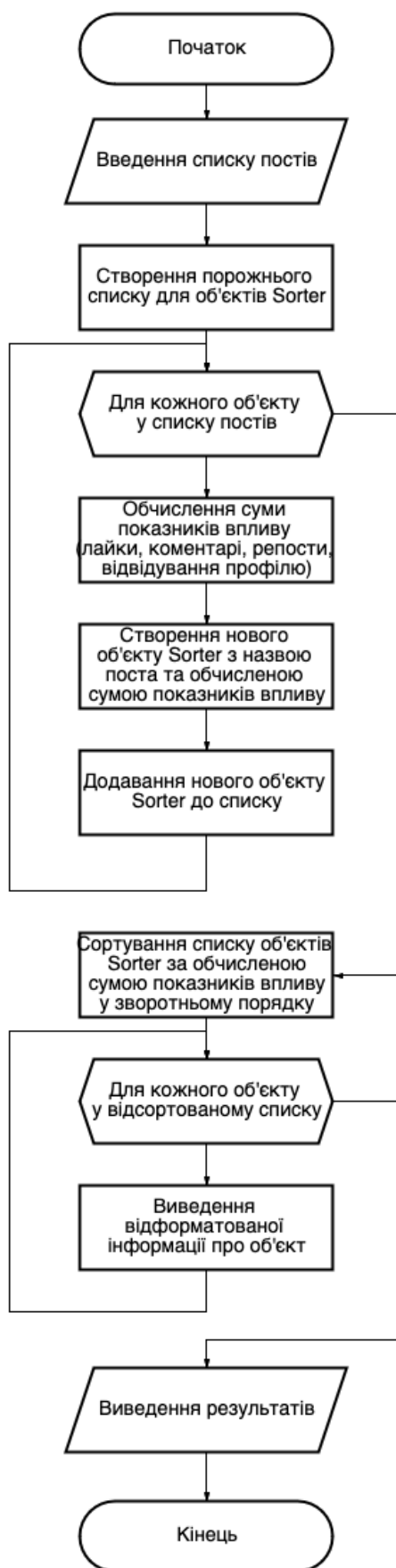


Рис. 2.4 – Схема методу узагальнення взаємодій з постом – show\_popular.

### 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Розробка головного класу Post

Для створення програмного забезпечення з метою аналізу даних з соціальних медіа з CSV-файлу, який містить інформацію про публікації, лайки, коментарі та репости, потрібно виконати такі кроки:

##### 1. Визначення вимог до програми:

- Розробка функціоналу для аналізу популярності та взаємодії з контентом.
- Опрацювання даних з CSV-файлу: визначення структури файлу, вибір необхідних полів для аналізу.

##### 2. Розробка архітектури програми:

- Визначення компонентів програми: модулі для обробки даних, аналізу популярності, взаємодії з контентом тощо.
- Встановлення взаємозв'язків між компонентами, їх взаємодія для забезпечення потрібного функціоналу.
- Вибір оптимальних методів обробки та аналізу даних для забезпечення швидкості та ефективності програми.

##### 3. Реалізація компонентів програми:

- Розробка коду компонентів відповідно до принципів ООП.
- Забезпечення коректної роботи кожного компонента та їх взаємодії з даними з CSV-файлу.

##### 4. Тестування програми:

- Проведення тестів для перевірки коректності роботи програми, відповідно до заданих вимог.
- Виявлення та усунення будь-яких помилок або недоліків.

##### 5. Виправлення та перевірка правильності роботи:

- Внесення виправлень у програму, якщо необхідно після тестування.

					5.151.1.ВКЗ-КП	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

- Повторна перевірка програми для забезпечення її коректної роботи після внесених змін.

З вже визначеними вимогами та розробленою архітектурою можна приступити до розробки програмного коду:

Для легкого маніпулювання даними що знаходяться в CSV файлі їх рекомендується перевести в об'єкти, для цього створюємо клас Post, який має всі необхідні атрибути, такі як:

```
self.post_title
self.likes
self.comments
self.reposts
self.profile_visits
```

```
11 class Post:
12
13     def __init__(self, post_title=None, likes=None, comments=None, reposts=None, profile_visits=None):
14         self.post_title = post_title or post_t.get_random()
15         self.likes = likes or random.randint(50, 3000)
16         self.comments = comments or random.randint(1, self.likes // 2)
17         self.reposts = reposts or random.randint(0, self.comments)
18         self.profile_visits = profile_visits or random.randint(self.likes, 10000)
19
20     def __str__(self):
21         return f"{self.post_title}, {self.likes}, {self.comments}, {self.reposts}, {self.profile_visits}"
22
23     def print_out(self):
24         return f"{colors.OKCYAN}Post Title: {self.post_title}, {colors.OKGREEN}Likes: {self.likes}, {colors.OKBLUE}Comments: {self.comments},
```

Рис. 3.1.1 – Клас об'єкту Post.

## Методи

**\_\_str\_\_:** Цей метод визначає рядкове представлення об'єкта типу "пост". Він викликається при використанні функції str() або при виведенні об'єкта на екран. У цьому випадку він повертає рядок, який містить інформацію про заголовок поста, кількість лайків, коментарів, репостів і відвідувань профілю.

**print\_out:** Цей метод створений для форматowanego виведення інформації про об'єкт типу "пост" у терміналі з використанням кольорів, які імпортовані з класу Bcolors. Він повертає рядок, який містить інформацію про за-

головок поста, кількість лайків, коментарів, репостів і відвідувань профілю, причому кожен параметр окремо забарвлюється відповідним кольором.

В даному прикладі у кожного атрибуту є параметри за замовчуванням, які використовують клас PostTitle та Generator щоб створити початковий CSV файл для тестування програми, давайте розглянемо в пункті 3.2:

					5.151.1.ВКЗ-КП	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.2 Класи для тестування

Клас `Generator` має в собі атрибут `rows`(поля) та метод `generate()` який за кількістю полів створює випадкові об'єкти класу `Post` та записує їх в файл.

```
6 class Generator:
7     def __init__(self, rows):
8         self.rows = rows
9
10    def generate(self, filename):
11        with open(f"{filename}.csv", "a") as f:
12            f.write("Post title, Likes, Comments, Reposts, Profile visits" + "\n")
13            for _ in range(self.rows):
14                post = Post()
15                f.write(str(post) + "\n")
```

Рис. 3.2.1 – Клас `Generator`.

Клас `PostTitle` має список вже заданих назв постів та метод який повертає один з цих назв – `get_random()`.

```
2 import random
3
4 class PostTitle:
5     def __init__(self):
6         self.post_titles = [ ... ]
7
8     def get_random(self):
9         random_title = random.choice(self.post_titles)
10        return random_title
```

Рис. 3.2.2 – Клас `PostTitle`.

Всі ці класи були створені для спрощення тестування програми і не мають прямого впливу на її роботу, хоча й дуже допоможуть в майбутньому, коли програма буде повністю готова.

### 3.3 Клас CSV\_Reader

Перейдемо до створення класу який буде відповідати за імпортування самого CSV-файлу та переведенню його в об'єкти Post та буде зручно зберігати їх в атрибуті `opened_file` який є масивом.

```
class CSV_Reader:

    def __init__(self) -> None:
        self.file = None
        self.opened_file = []

    def import_file(self):
        file_input = input(f"{Bcolors.WHITE}Enter your CSV file name: ")

        try:
            with open(f"{file_input}.csv") as opened_file:
                self.file = opened_file
                print(f"{Bcolors.OKGREEN}Imported successfully.")
                self.opened_file = self._csv_to_post()
        except FileNotFoundError:
            print(f"{Bcolors.WARNING}File \"{file_input}\" does not exist")

    def _csv_to_post(self):
        csv_file = self.file

        decoded_list = []

        for row in csv_file:
            decoded_list.append(row.split(","))

        for number in range(len(decoded_list)):
            decoded_list[number][-1] = decoded_list[number][-1].strip()

        decoded_list.pop(0)

        object_list = []

        for item in decoded_list:
            new_post = Post(item[0], int(item[1]), int(item[2]), int(item[3]), int(item[4]))
            object_list.append(new_post)

        return object_list
```

Рис. 3.3.1 – Клас CSV\_Reader.

#### Принцип дії:

1. **Ініціалізація:** У конструкторі `__init__`, він ініціалізує дві змінні: `file` (яка представлятиме відкритий файл) та `opened_file` (яка буде зберігати об'єкти, що відображають дані з CSV файлу).
2. **Імпорт файлу:** Метод `import_file` отримує назву файлу від користувача, як введення. Він відкриває цей файл (якщо він існує), читає його



вміст і перетворює його на список об'єктів Post. Якщо файл не знайдено, виводиться відповідне повідомлення.

3. **Перетворення CSV у пости:** Метод `_csv_to_post` відповідає за перетворення вмісту CSV файлу на об'єкти Post. Він читає CSV файл, розділяє кожен рядок за комами, видаляє зайві пробіли, видаляє перший рядок (який зазвичай містить заголовки стовпців), і створює об'єкти Post з кожного рядка, використовуючи дані з цього рядка.

					5.151.1.BK3-КП	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.4 Клас Bcolors

Ще один клас який можна віднести до допоміжних – Bcolors, ось що про нього потрібно знати:

1. **Статичні атрибути:** Клас Bcolors має ряд статичних атрибутів, які представляють різні кольори. Кожен атрибут є рядком, що містить спеціальний код кольору ANSI для відображення тексту в відповідному кольорі у терміналі.
2. **Використання кольорів:** Ці атрибути можна використовувати, додавши їх до виведеного тексту за допомогою спеціальних кодів у форматі Escape-послідовностей. Наприклад, Bcolors.OKGREEN додасть код кольору, щоб текст виводився зеленим кольором. Це може бути корисно для покращення зручності виведення інформації в консольному інтерфейсі вашої програми.
3. **Нестатичні атрибути та конструктор:** Оскільки всі атрибути цього класу є статичними, йому не потрібен конструктор. Статичні атрибути доступні через ім'я класу, а не через екземпляр класу. Тому немає потреби в ініціалізації цих атрибутів для кожного об'єкта класу. Вони доступні всюди в програмі, де імпортований цей клас.

```
colors.py > ...
1  class Bcolors:
2      HEADER = '\033[95m'
3      OKBLUE = '\033[94m'
4      OKCYAN = '\033[96m'
5      OKGREEN = '\033[92m'
6      WARNING = '\033[93m'
7      FAIL = '\033[91m'
8      ENDC = '\033[0m'
9      BOLD = '\033[1m'
10     UNDERLINE = '\033[4m'
11     WHITE = '\033[97m'
12
```

Рис. 3.4.1 – Клас Bcolors.

### 3.5 Сортуння та клас Sorter

Одним із найважливіших завдань у рамках цього проекту є організація та обчислення загальних показників для кожного поста. Для цього був створений окремий клас під назвою Sorter, який допомагає в сортуванні та аналізі інформації.

Клас Sorter має ряд методів, серед яких конструктор (**init**), який ініціалізує атрибути об'єктів, такі як назва поста (**post\_title**) та кількість переглядів (**impressions**). Метод **print\_out** виводить інформацію про пост, включаючи назву та кількість переглядів. Метод **print\_list** приймає список постів і виводить інформацію про кожен з них. Метод **sort** сортує список постів за вибраним параметром (наприклад, назвою, кількістю лайків тощо) та виводить відсортований список. Метод **show\_popular** аналізує популярність кожного поста, розраховуючи загальну кількість взаємодій (лайки, коментарі, репости, відвідування профілю) і виводить пости в порядку спадання популярності.

```
4 class Sorter:
5
6     def __init__(self, post_title=None, impressions=None):
7         self.post_title = post_title or "Example"
8         self.impressions = impressions or "0"
9
10    def print_out(self):
11        return f"{Bcolors.OKCYAN}Post Title: {self.post_title}, {Bcolors.OKGREEN}Impressions: {self.impressions}"
12
13    def print_list(self, post_list):
14        for item in post_list:
15            print(item.print_out())
16
17    def sort(self, post_list):
18        option_list = ["post_title", "likes", "comments", "repost", "profile_visits", "topic"]
19        sort_by = int(input("Do you want to sort by Title(1), Likes(2), Comment(3), Reposts(4), Profile visits(5): "))
20        attr_sort = option_list[sort_by - 1]
21        sorted_test = sorted(post_list, key=attrgetter(attr_sort), reverse=True)
22        for test in sorted_test:
23            print(test.print_out())
24
25    def show_popular(self, post_list):
26        imr_list = []
27        for post in post_list:
28            impressions = post.likes + post.comments + post.reposts + post.profile_visits
29            imr_list.append(Sorter(post.post_title, impressions))
30        sorted_imp = sorted(imr_list, key=attrgetter("impressions"), reverse=True)
31        for srd in sorted_imp:
32            print(srd.print_out())
```

Рис. 3.5.1 – Клас Sorter.

Цей клас був створений таким, як він є, для того, щоб забезпечити зручний та ефективний інструмент для обробки даних про пости у соціальних мережах. Основна мета цього класу полягає в тому, щоб упорядковувати та аналізувати інформацію про пости з різних кутів зору, таких як загальна популярність, кількість лайків, коментарів, репостів тощо.

Створення окремого класу дозволяє легко маніпулювати та аналізувати дані, використовуючи спеціалізовані методи та функції. Наприклад, метод `sort` дозволяє швидко сортувати список постів за різними критеріями, тоді як метод `show_popular` робить можливим виділення найпопулярніших постів.

Крім того, використання класу дозволяє зберігати інформацію про кожен пост у відокремлених об'єктах, що спрощує доступ до цієї інформації та робить код більш структурованим та зрозумілим для подальшого розширення та модифікації. Такий підхід сприяє підтримці чистого та організованого коду, що є важливим для подальшого розвитку проекту та підтримання принципів ООП.

### 3.6 Основний файл виконання програми

Файл `main.py` - це головний файл програми, де відбувається взаємодія з іншими частинами програми. У парадигмі об'єктно-орієнтованого програмування (ООП) `main.py` виконує роль координатора роботи об'єктів та визначає послідовність їх викликів та дій.

У моєму коді, `main.py` відповідає за створення об'єктів `Post`, `CSV_Reader` і `Sorter`, тобто за їх створення та управління.

Структура файлу `main.py` досить проста і пряма:

1. Спочатку створюються об'єкти для постів (`Post`), `CSV` читача (`CSV_Reader`) та сортувальника (`Sorter`).
2. Потім `main.py` імпортує файл (з використанням методу `import_file()` з `CSV` читача) та отримує відкритий файл.

					5.151.1.BK3-КП	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Далі створюється цикл, який перевіряє, чи існує файл. Якщо файл не існує, програма намагається знову імпортувати його.
4. Після цього відбувається безкінечний цикл, де користувач може обирати, що саме він хоче зробити з файлом: переглянути, відсортувати, отримати статистику або вибрати інший файл. Залежно від вибору користувача, викликаються відповідні методи об'єкта `Sorter`, який здійснює відповідні операції над відкритим файлом.

```

post = Post()
csv_reader = CSV_Reader()
sorter = Sorter()

while not csv_reader.file:
    csv_reader.import_file()
    opened_file = csv_reader.opened_file
else:
    while True:
        show_sort = int(input(f"{Bcolors.WHITE}Do you want to view(1), sort(2), get statistics for your file(3), or pick other file(4)?"))

        if show_sort == 1:
            sorter.print_list(opened_file)
        elif show_sort == 2:
            sorter.sort(opened_file)
        elif show_sort == 3:
            sorter.show_popular(opened_file)
        elif show_sort == 4:
            csv_reader.file = None
            csv_reader.import_file()
            while not csv_reader.file:
                csv_reader.import_file()
            else:
                opened_file = csv_reader.opened_file

```

Рис. 3.6.1 – Файл `main.py`.

На цьому етапі розробку коду ПО можна вважати закінченою, всі потрібні класи, методи та модулі були створені та всі вимоги до функціоналу програми задовільнені.

## 4. КЕРІВНИЦТВО КОРИСТУВАЧА


### 4.1 Основний функціонал

Програма має наступний функціонал:

- Перегляд файлу в форматованому вигляді (view)
- Сортування за назвою (Title)
- Сортування за кількістю лайків (Likes)
- Сортування за кількістю коментарів (Comment)
- Сортування за кількістю репостів (Reposts)
- Сортування за кількістю візитів профілю з посту (Profile visits)
- Отримання загальної статистики для кожного посту (get statistics for your file)
- Вибір іншого файлу (pick other file)

### 4.2 Запуск програми

Для запуску програми треба виконати наступні дії:

1. Перейти в внутрішній каталог програми.
2. Відкрити файл main.py.
3. В Visual Studio Code, в правому верхньому кутку натиснути на кнопку  (Run Python File).

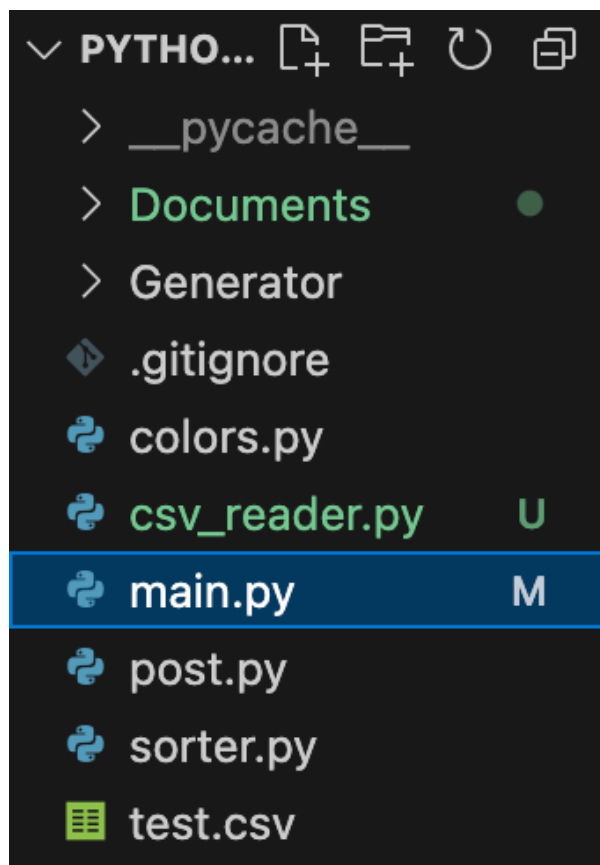


Рис. 4.2.1 – Файл main.py в кореневому каталозі.

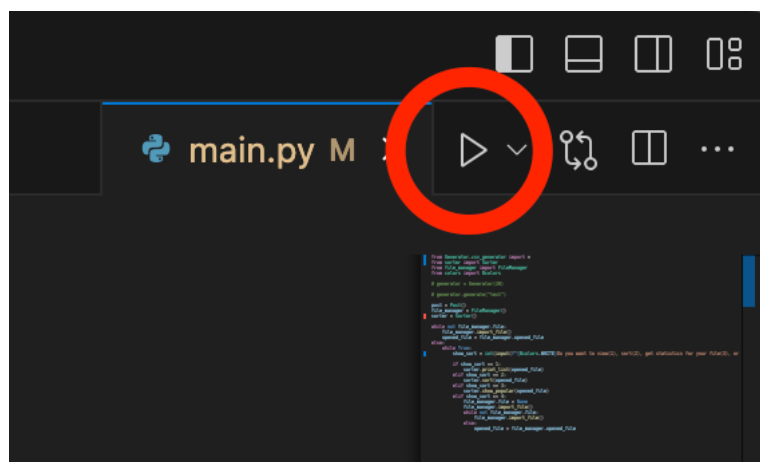


Рис. 4.2.2 – Кнопка запуску файлу (відмічена червоним кольором).

### 4.3 Імпортування файлу

Перед використанням програми треба переконатися в наявності сумісного CSV файлу в корневому каталозі, сумісний файл повинен мати наступні колонки в зазначеному порядку:

#### **Post title, Likes , Comments, Reposts, Profile visits**

1. Після переконання в тому, що файл міститься в каталозі програми, запусіть програму:

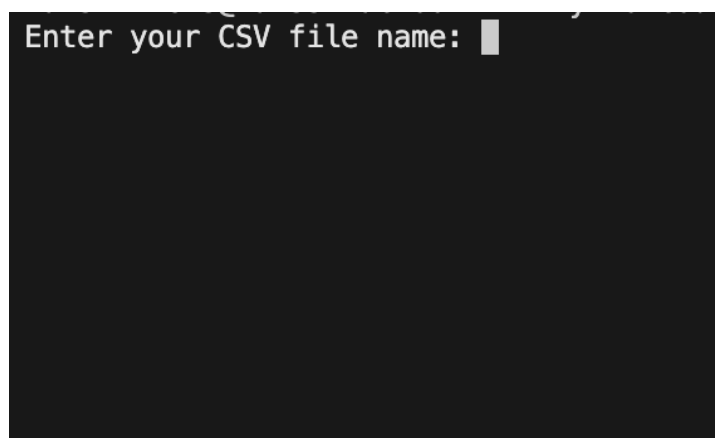


Рис. 4.3.1 – Початковий екран вибору файлу.

2. Після запуску, на початковому екрані (Рис. 4.3.1), введіть ім'я файлу у форматі лише назви (без формату файлу).

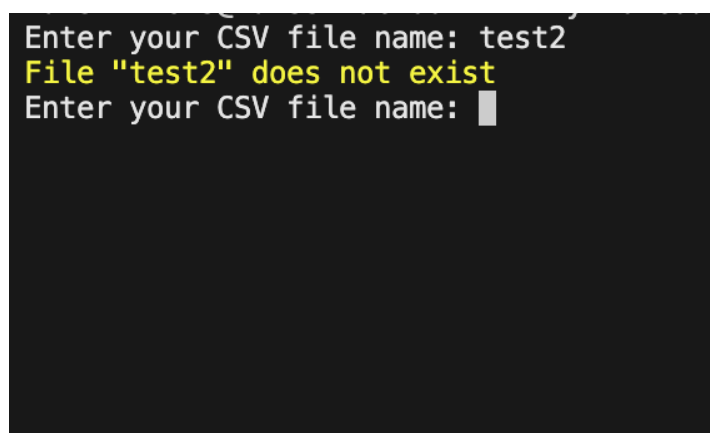


Рис. 4.3.2 – Помилка при знаходженні файлу.



3. Помилка “File “ім’я файлу” does not exist” (Рис 4.3.2) наголошує користувачеві на тому, що файлу з введеною назвою не існує, в такому випадку рекомендується перевірити наявність файлу в корневій папці програми або впевнитися що назва була вказана правильно.

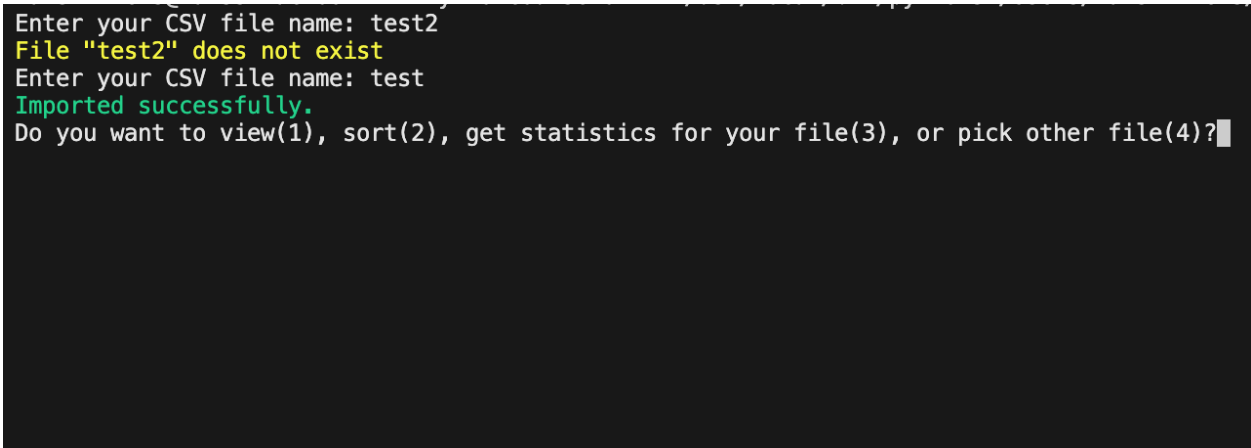


Рис. 4.3.3 – Приклад коректно введеного файлу.

4. Якщо файл було введено коректно програма засвідчить про це написом “Imported successfully” (Рис 4.3.3).

4.4 Робота з файлом

Функція view:

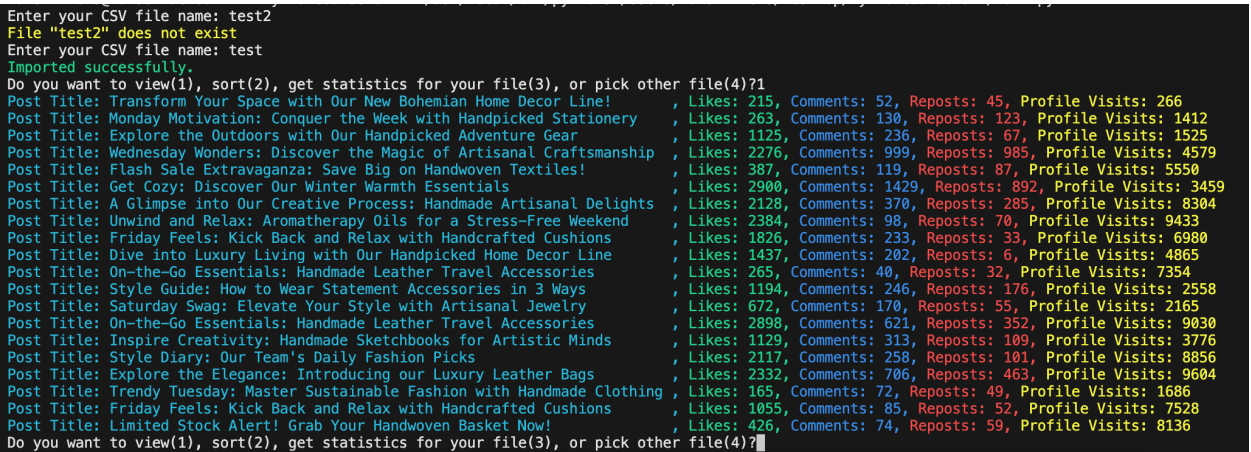


Рис. 4.4.1 – Перегляд файлу за допомогою опції “view”.

Фнкція “view” (Рис 4.3.4) виводить всі пости в файлі CSV в оригінальному їх порядку, виділяючи дані різними кольорами для більш комфортного перегляду.

**Функція sort:**

Рис. 4.4.2 – Вибір стовпчика сортування.

```
Enter your CSV file name: test
Imported successfully.
Do you want to view(1), sort(2), get statistics for your file(3), or pick other file(4)?2
Do you want to sort by Title(1), Likes(2), Comment(3), Reposts(4), Profile visits(5): █
```

При виборі функції sort користувачу пропонується обрати стовпчик по якому буде проходити сортування (від більшого числа до меншого).

```
Do you want to sort by Title(1), Likes(2), Comment(3), Reposts(4), Profile visits(5): 2
Post Title: Get Cozy: Discover Our Winter Warmth Essentials , Likes: 2900, Comments: 1429, Reposts: 892, Profile Visits: 3459
Post Title: On-the-Go Essentials: Handmade Leather Travel Accessories , Likes: 2898, Comments: 621, Reposts: 352, Profile Visits: 9030
Post Title: Unwind and Relax: Aromatherapy Oils for a Stress-Free Weekend , Likes: 2384, Comments: 98, Reposts: 70, Profile Visits: 9433
Post Title: Explore the Elegance: Introducing our Luxury Leather Bags , Likes: 2332, Comments: 706, Reposts: 463, Profile Visits: 9604
Post Title: Wednesday Wonders: Discover the Magic of Artisanal Craftsmanship , Likes: 2276, Comments: 999, Reposts: 985, Profile Visits: 4579
Post Title: A Glimpse into Our Creative Process: Handmade Artisanal Delights , Likes: 2128, Comments: 370, Reposts: 285, Profile Visits: 8304
Post Title: Style Diary: Our Team's Daily Fashion Picks , Likes: 2117, Comments: 258, Reposts: 101, Profile Visits: 8856
Post Title: Friday Feels: Kick Back and Relax with Handcrafted Cushions , Likes: 1826, Comments: 233, Reposts: 33, Profile Visits: 6980
Post Title: Dive into Luxury Living with Our Handpicked Home Decor Line , Likes: 1437, Comments: 202, Reposts: 6, Profile Visits: 4865
Post Title: Style Guide: How to Wear Statement Accessories in 3 Ways , Likes: 1194, Comments: 246, Reposts: 176, Profile Visits: 2558
Post Title: Inspire Creativity: Handmade Sketchbooks for Artistic Minds , Likes: 1129, Comments: 313, Reposts: 109, Profile Visits: 3776
Post Title: Explore the Outdoors with Our Handpicked Adventure Gear , Likes: 1125, Comments: 236, Reposts: 67, Profile Visits: 1525
Post Title: Friday Feels: Kick Back and Relax with Handcrafted Cushions , Likes: 1055, Comments: 85, Reposts: 52, Profile Visits: 7528
Post Title: Saturday Swag: Elevate Your Style with Artisanal Jewelry , Likes: 672, Comments: 170, Reposts: 55, Profile Visits: 2165
Post Title: Limited Stock Alert! Grab Your Handwoven Basket Now! , Likes: 426, Comments: 74, Reposts: 59, Profile Visits: 8136
Post Title: Flash Sale Extravaganza: Save Big on Handwoven Textiles! , Likes: 387, Comments: 119, Reposts: 87, Profile Visits: 5550
Post Title: On-the-Go Essentials: Handmade Leather Travel Accessories , Likes: 265, Comments: 40, Reposts: 32, Profile Visits: 7354
Post Title: Monday Motivation: Conquer the Week with Handpicked Stationery , Likes: 263, Comments: 130, Reposts: 123, Profile Visits: 1412
```

Рис. 4.4.3 – Приклад сортування за кількістю лайків (2).

**Функція get statistics for your file:**

Функція get statistics for your file показує загальну кількість взаємодій користувачів з постом щоб отримати об’єктивний список самих популярних постів.

Рис. 4.4.4 – Приклад роботи функції get statistics for your file.

```
Do you want to view(1), sort(2), get statistics for your file(3), or pick other file(4)?3
Post Title: Explore the Elegance: Introducing our Luxury Leather Bags , Impressions: 13105
Post Title: On-the-Go Essentials: Handmade Leather Travel Accessories , Impressions: 12901
Post Title: Unwind and Relax: Aromatherapy Oils for a Stress-Free Weekend , Impressions: 11985
Post Title: Style Diary: Our Team's Daily Fashion Picks , Impressions: 11332
Post Title: A Glimpse into Our Creative Process: Handmade Artisanal Delights , Impressions: 11087
Post Title: Friday Feels: Kick Back and Relax with Handcrafted Cushions , Impressions: 9072
Post Title: Wednesday Wonders: Discover the Magic of Artisanal Craftsmanship , Impressions: 8839
Post Title: Friday Feels: Kick Back and Relax with Handcrafted Cushions , Impressions: 8720
Post Title: Limited Stock Alert! Grab Your Handwoven Basket Now! , Impressions: 8695
Post Title: Get Cozy: Discover Our Winter Warmth Essentials , Impressions: 8680
Post Title: On-the-Go Essentials: Handmade Leather Travel Accessories , Impressions: 7691
Post Title: Dive into Luxury Living with Our Handpicked Home Decor Line , Impressions: 6510
Post Title: Flash Sale Extravaganza: Save Big on Handwoven Textiles! , Impressions: 6143
Post Title: Inspire Creativity: Handmade Sketchbooks for Artistic Minds , Impressions: 5327
Post Title: Style Guide: How to Wear Statement Accessories in 3 Ways , Impressions: 4174
Post Title: Saturday Swag: Elevate Your Style with Artisanal Jewelry , Impressions: 3062
Post Title: Explore the Outdoors with Our Handpicked Adventure Gear , Impressions: 2953
Post Title: Trendy Tuesday: Master Sustainable Fashion with Handmade Clothing , Impressions: 1972
```

**Функція pick another file:**

Функція pick another file повертає користувача в меню вибору файлу (див розділ “4.3 Імпортування файлу”)

## ВИСНОВКИ

Під час виконання цієї курсової роботи було успішно створено застосунок для роботи з CSV-файлами та їх вмістом. Проведено аналіз процесу розробки програмного забезпечення та визначено необхідність такого програмного забезпечення для ефективного сортування та аналізу постів соціальних мереж. Основним методом для вирішення цієї задачі стало створення програмного забезпечення за допомогою мови Python, яке включало в себе роботу з класами та парадигмою ООП.

Був розроблений загальний алгоритм роботи програми, в якому були описані класи, які було доцільно створити, та їх призначення. Також була створена блок-схема основних функцій. Додатково були розроблені функціональні алгоритми роботи програми, де детально описано вимоги до додатку, коротко описаний його функціонал.

Під час розробки програмного забезпечення роботу було розбито на кроки для спрощення та зручності виконання завдання, і написаний відповідний код. У керівництві користувача детально описано необхідні дії для запуску додатку та його використання.

В результаті виконання курсової роботи було успішно досягнуто мети - розроблено програмне забезпечення для аналізу даних з соціальних медіа на мові Python, яке дозволяє ефективно аналізувати дані та виводити статистику постів.

					5.151.1.VK3-КП	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

## ЛІТЕРАТУРА

1. CSV – Вікіпедія: веб-сайт. URL: <https://uk.wikipedia.org/wiki/CSV>
2. How to read CSV files with Python #1 - Without using python libraries:  
веб-сайт. URL:  
<https://youtu.be/GBD4na8AQjY?si=x9-NbacAcLiKra48> (дата звернення 19.01.2024)
3. W3Schools Online Web Tutorials: веб-сайт. URL:  
<https://www.w3schools.com> (дата звернення 14.01.2024)
4. Пришвидшений курс Python. Практичний проєктно-орієнтований вступ до програмування – Ерік Маттерс, 2021 – 600 с

					5.151.1.ВКЗ-КП	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

## ДОДАТКИ

### Додаток А

```
post = Post()
csv_reader = CSV_Reader()
sorter = Sorter()

while not csv_reader.file:
    csv_reader.import_file()
    opened_file = csv_reader.opened_file
else:
    while True:
        show_sort = int(input(f'{Bcolors.WHITE}Do you want to view(1), sort(2),
get statistics for your file(3), or pick other file(4)?'))

        if show_sort == 1:
            sorter.print_list(opened_file)
        elif show_sort == 2:
            sorter.sort(opened_file)
        elif show_sort == 3:
            sorter.show_popular(opened_file)
        elif show_sort == 4:
            csv_reader.file = None
            csv_reader.import_file()
            while not csv_reader.file:
                csv_reader.import_file()
            else:
                opened_file = csv_reader.opened_file
```

					5.151.1.ВКЗ-КП	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

```

class CSV_Reader:

    def __init__(self) -> None:
        self.file = None
        self.opened_file = []

    def import_file(self):
        file_input = input(f'{Bcolors.WHITE}Enter your CSV file name: ')

        try:
            with open(f'{file_input}.csv') as opened_file:
                self.file = opened_file
                print(f'{Bcolors.OKGREEN}Imported successfully.')
                self.opened_file = self._csv_to_post()
        except FileNotFoundError:
            print(f'{Bcolors.WARNING}File \"{file_input}\" does not exist")

    def _csv_to_post(self):
        csv_file = self.file

        decoded_list = []

        for row in csv_file:
            decoded_list.append(row.split(","))

        for number in range(len(decoded_list)):
            decoded_list[number][-1] = decoded_list[number][-1].strip()

        decoded_list.pop(0)

        object_list = []

        for item in decoded_list:
            new_post = Post(item[0], int(item[1]), int(item[2]), int(item[3]),
int(item[4]))
            object_list.append(new_post)

        return object_list

```

```
fake = Faker()
```

```
class Post:
```

```
    def __init__(self, post_title=None, likes=None, comments=None, reposts=None,
profile_visits=None):
```

```
        self.post_title = post_title or PostTitle.get_random()
```

```
        self.likes = likes or random.randint(50, 3000)
```

```
        self.comments = comments or random.randint(1, self.likes // 2)
```

```
        self.reposts = reposts or random.randint(0, self.comments)
```

```
        self.profile_visits = profile_visits or random.randint(self.likes, 10000)
```

```
    def __str__(self):
```

```
        return f'{self.post_title}, {self.likes}, {self.comments}, {self.reposts},
{self.profile_visits}'
```

```
    def print_out(self):
```

```
        return f'{Bcolors.OKCYAN}Post Title: {self.post_title}, {Bcol-
ors.OKGREEN}Likes: {self.likes}, {Bcolors.OKBLUE}Comments:
{self.comments}, {Bcolors.FAIL}Reposts: {self.reposts}, {Bcol-
ors.WARNING}Profile Visits: {self.profile_visits}'
```

					5.151.1.BK3-KП	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

```
class Sorter:
```

```
    def __init__(self, post_title=None, impressions=None):
```

```
        self.post_title = post_title or "Example"
```

```
        self.impressions = impressions or "0"
```

```
    def print_out(self):
```

```
        return f'{Bcolors.OKCYAN}Post Title: {self.post_title}, {Bcol-  
ors.OKGREEN}Impressions: {self.impressions}'
```

```
    def print_list(self, post_list):
```

```
        for item in post_list:
```

```
            print(item.print_out())
```

```
    def sort(self, post_list):
```

```
        option_list = ["post_title", "likes", "comments", "repost", "profile_visits",  
"topic"]
```

```
        sort_by = int(input("Do you want to sort by Title(1), Likes(2), Comment(3),  
Reposts(4), Profile visits(5): "))
```

```
        atter_sort = option_list[sort_by - 1]
```

```
        sorted_test = sorted(post_list, key=attrgetter(atter_sort), reverse=True)
```

```
        for test in sorted_test:
```

```
            print(test.print_out())
```

```
    def show_popular(self, post_list):
```

```
        imr_list = []
```

```
        for post in post_list:
```

```
            impressions = post.likes + post.comments + post.reposts +  
post.profile_visits
```

```
            imr_list.append(Sorter(post.post_title, impressions))
```

```
        sorted_imp = sorted(imr_list, key=attrgetter("impressions"), reverse=True)
```

```
        for srd in sorted_imp:
```

```
            print(srd.print_out())
```



class Bcolors:

HEADER = '\033[95m'

OKBLUE = '\033[94m'

OKCYAN = '\033[96m'

OKGREEN = '\033[92m'

WARNING = '\033[93m'

FAIL = '\033[91m'

ENDC = '\033[0m'

BOLD = '\033[1m'

UNDERLINE = '\033[4m'

WHITE = '\033[97m'

					5.151.1.BK3-КП	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		