Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

ЛАБОРАТОРНА РОБОТА № 5

з дисципліни «Основи програмування»

Тема: Потоки вводу-виводу

Виконали:

Студентки групи ІА-31

Горлач Д., Макасєєва М.,

Соколова П.

Перевірив:

Степанов А.

Київ-2023

Тема: Потоки вводу-виводу

Мета: Ознайомитись з наданими нам матеріалами, пригадати інформацію надану на лекціях з даної теми, також навчившись з минулих лабораторних робіт правильно використовувати знання та реалізовувати за допомогою них завдання, виконати поставлену нам задачу. А саме, скористатись новими знаннями та реалізувати за допомогою них подане завдання.

Хід роботи:

Шукаємо файл по його назві. Якщо назва - null, викидаємо помилку (IllegalArgumentException), а також якщо файл не був знайдений (FileNotFoundException). Читаємо файл рядок за рядком, кожен з яких перетворюємо на масив слів, за допомогою методу String.split, і добавляємо в список (ArrayList). Використовуючи цикл - for біжемо по списку слів і інкрементуємо кількість знайдено конкретного слова і зберігаємо в Map як ключ-значення (слово-кількість). А також є додаткові 2 змінні - слово (oftenWord) і кількість (oftenWordCount), в яких зберігається найчастіше слово і кількість цього слова відповідно, перевіряючи oftenWordCount на кожній ітерації.

2. Виконати завдання з таблиці 2 відповідно до свого варіанту у таблиці 1.
- Завдання має бути реалізовано як окремий клас.
- Клас має складатись щонайменше з таких методів:
  - public static void main(String[] agrs) - точка входу.
  - Метод, що реалізує задане завдання. Метод має перевіряти аргументи та у разі їх помилковості аварійно закінчувати свою роботу шляхом викидання стандартного виключення IllegalArgumentException або NullPointerException. В разі неможливості виконання операції, метод повинен викидати IOException або FileNotFoundException. В жодному разі цей метод не повинен напряму взаємодіяти з користувачем через консоль або інший UI (ніколи не змішуйте бізнес-логіку та користувацький інтерфейс).
- Клас може містити інші допоміжні методи.
- При виконанні завдань слід звернути увагу на ефективність з точки зору швидкодії:
  - операції вводу/виводу слід здійснювати через буфер;
  - програму потрібно написати так, щоб зчитування кожного фрагмента файлу здійснювалось лише один раз (тобто від початку файлу до кінця за один прохід без повторного зчитування).

Рис 1. Задача

6	2	Англійська	Так

Рис 2. Наш варіант

2	String commonestWord(String filename) Знайти у текстовому файлі слово, яке зустрічається найчастіше. Якщо таких слів декілька – повернути будь-яке з них.

Рис. 3 завдання

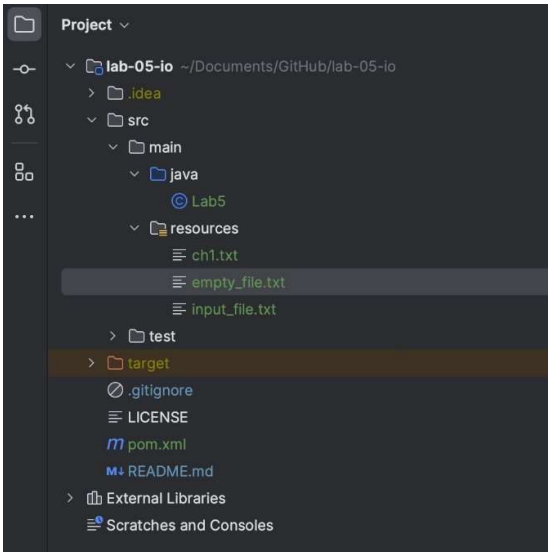

Рис. 4 структура проекту

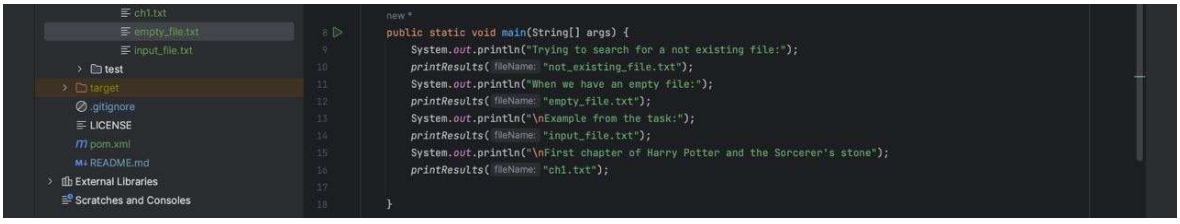

Рис. 5 метод main

Рис. 6 результат виконання

Висновок:

За допомогою блоку - try-with-resources після роботи з файлом автоматично звільняються ресурси. Використовуючи Buffer можна отримувати дані не по одному символу, а одразу рядком, після чого кожен цей рядок можна перетворювати на масив слів, який ми додаємо в загальний список. І використовуючи Map можна зберігати кількість знаходжень кожного конкретного слова. Так як ключ в Map регістро-залежний, нам не треба перевіряти регістр щоб дізнатися чи є слова різними, тому що для Map навіть регістр 1 символа в слові вважається новим словом.