

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра інформатики та програмної інженерії



Звіт до лабораторної роботи №1

з курсу

«Мультипарадигмене програмування»

студентки 2 курсу

групи ІТ-02

Наконечної Ірини Олександрівни

Викладач:

Очеретяний О.К.

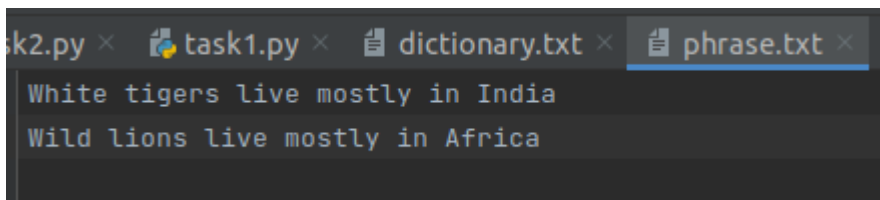
Київ – 2022

В ході реалізації лабораторної роботи я використовувала python, попередньо імпортувавши модуль для використання оператора **goto**.

Завдання 1

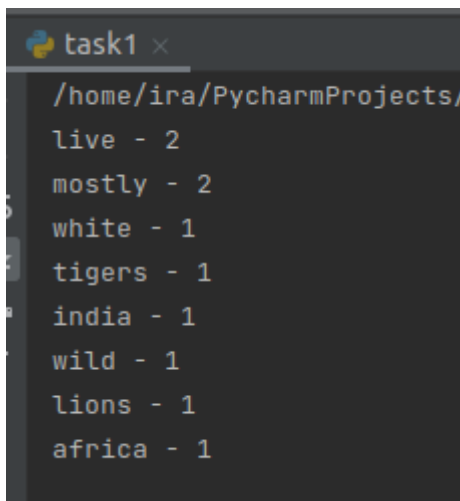
Обчислювальна задача тут тривіальна: для текстового файлу ми хочемо відобразити N (наприклад, 25) найчастіших слів і відповідну частоту їх повторення, упорядковано за зменшенням. Слід обов'язково нормалізувати використання великих літер і ігнорувати стоп-слова, як «the», «for» тощо. Щоб все було просто, ми не піклуємося про порядок слів з однаковою частотою повторень. Ця обчислювальна задача відома як **term frequency**.

Для виконання даного завдання я використовую файл phrase.txt. Ось наповнення цього файлу:



```
task2.py × task1.py × dictionary.txt × phrase.txt ×
White tigers live mostly in India
Wild lions live mostly in Africa
```

Результат роботи програми



```
task1 ×
/home/ira/PycharmProjects/
live - 2
mostly - 2
white - 1
tigers - 1
india - 1
wild - 1
lions - 1
africa - 1
```

Опис алгоритму

Для зберігання слів я використовую таку структуру даних як словник (words_set).

1. Зчитуємо файл

2. Йдемо по кожному рядку файлу
3. Зчитуємо кожен символ рядку
4. Формуємо з символів слово, розділяючи по пробілу або новому рядку
5. Переводимо слово в нижній регістр
6. Перевіряємо чи дане слово вже наявне в словнику
 - якщо не наявне - перевіряємо чи дане слово не є стоп-словом (чи не міститься в масиві `words_to_avoid`). Якщо дане слово не є стоп-словом, то додаємо його в словник, якщо це все-таки стоп-слово, то ігноруємо
 - якщо наявне - інкрементимо лічильник повторень біля цього слова
7. Сортуюмо слова за значенням лічильника
8. Виводимо на екран, відображаючи перші 25 слів, що відповідають значенню змінною `limited_output`

Завдання 2

Тепер, нам потрібно виконати задачу, що називається словниковим індексуванням. Для текстового файлу виведіть усі слова в алфавітному порядку разом із номерами сторінок, на яких ці слова знаходяться. Ігноруйте всі слова, які зустрічаються більше 100 разів. Припустимо, що сторінка являє собою послідовність із 45 рядків.

Для виконання даного завдання я використовую файл `dictionary.txt`. Наповнення файлу - уривок з “Pride and Prejudice”:

copy 1. task.txt 2. dictionary.txt 3. phrase.txt

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife. However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is universal, and will be so in all ages, to all minds, and in all climates. My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?"

Mr. Bennet replied that he had not.

"But it is," returned she; "for Mrs. Long has just been here, and she told me all about it. Mr. Bennet made no answer.

"Do not you want to know who has taken it?" cried his wife impatiently.

"You want to tell me, and I have no objection to hearing it." This was invitation enough.

"Why, my dear, you must know, Mrs. Long says that Netherfield is taken by a young man of large fortune; the son of the late Mr.彬利爵士. What is his name?"

"Bingley."

"Is he married or single?"

"Oh! single, my dear, to be sure! A single man of large fortune; four or five thousand a year. What business has he in London? What can he be doing there?"

"My dear Mr. Bennet," replied his wife, "how can you be so tiresome! You must know that I am sure he is only waiting for an opportunity of being introduced to you. He is very much attached to me. And so he is. Is that his design in settling here?"

"Design! nonsense, how can you talk so! But it is very likely that he may fall in love with one of the girls. I see no occasion for that. You and the girls may go, or you may send them by themselves, which perhaps will be best. My dear, you flatter me. I certainly have had my share of beauty, but I do not pre-

Результат роботи програми

```
a - 1, 2, 3
about - 1, 3
above - 3
abuse - 1
accept - 3
accomplished - 3
account - 1
acknowledged - 1, 2
acquaintance - 2, 3
acquainted - 2, 3
act - 2
actually - 2, 3
added - 3
addressed - 2
adjusting - 2
admiration - 3
```

Опис алгоритму

Для зберігання слів я знову ж таки використовую словник (`words_occurrence`).

1. Зчитуємо файл
2. Йдемо по кожному рядку файлу
3. Зчитуємо кожен символ рядку
4. Формуємо з символів слово, розділяючи по пробілу, новому рядку або розділових знаках
5. Переводимо слово в нижній регістр
6. Додаємо слово в словник, з вказівкою на сторінку, де це слово зустрічається. При цьому фіксуємо скільки разів слово зустрічається
7. Видаляємо зі словника слова, які зустрічаються більше 100 разів (`words_frequency_limit`)
8. Сортуюмо слова за алфавітним порядком
9. Виводимо слова на екран

Висновок

В рамках даної лабораторної роботи я реалізувала поставлені завдання в стилі імперативного програмування. Я зрозуміла наскільки використання функцій та циклів дозволяє спростити та скоротити час написання коду. Хоча завдання не є складними з точки зору реалізації, обмеження, наявні в даній лабораторній, значно ускладнюють процес виконання. Головною метою було використовувати якомога меншу кількість функцій, хоча використання деяких з них я все ж не змогла уникнути. Для роботи зі словником, я використала методи `keys()`, `items()`. Також, в першому завданні довелося застосувати метод `strip()`, для уникнення багів з пробілами та символами переносу, які виникали при зчитуванні вмісту файлу.