Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Лабораторна робота №1

з дисципліни

“Мультипарадигменне програмування”

Виконав
студент групи ІТ-03:

Дудченко О. О.

Перевірили:
Очеретяний О. К.
Глушко Б. С.

Київ 2021

Тема: імперативне програмування

Склад репозитрію: task1.cs, task2.cs, report.pdf

Платформа

Лабораторна робота була виконана на платформі .NET мовою C#. Мова була обрана через наявність оператору goto, що є ключовою вимогою до обраної мови програмування.

Нижче буде наведено опис алгоритму роботи програми.

Task 1

Вирішення задачі term frequency.

Передумова: надано файл і **правильно задано шлях до нього**
Алгоритм роботи наступний: з файлу вичитується його зміст. Далі ми повинні обробити текст, оскільки не доступні засоби методи String, то просто приведемо кожну літеру до нижнього регістру.

Далі сформуємо наступну структуру: двовимірний масив, що буде містити “кортежі” слова та його частоти

Далі ми наповнюємо наш масив словами, та при повторній зустрічі збільшуємо другий член кортежу.

Після цього виймаємо частоти, сортуємо їх. Та фінальним етапом є друкування рельтату: виводимо перші 25 частот, та відповідні їм слова.

Task 2

Словникове індексування.

Перший етап: вигрузка тексту в пам’ять. Далі пробіжимося по кожному рядку та зведемо літери верхнього регістру до нижнього, як і в попередньому завданні.

Далі формуємо двовимірний масив, де кортеж буде складатися з чотирьох елементів: слово, частота, сторінки, остання додана сторінка відповідно.

Це допоможе використати подібний підхід першого завдання у вирішенні другого.

Далі ми послідовно наповнюємо масив значеннями, відповідно обробляючи їх. Далі виймаємо слова, сортуємо та послідовно друкуємо вивід паралельно звертаючи увагу на ігнорування слів кількістю більше 100.

Деталі

В рамках лабораторної роботи потрібно було не використовувати засоби мови програмування для розбиття свого коду на класи та їх методи (ООП) чи процедури/функції.
Додатково можна було реалізувати процес обробки інформації самостійно, для глибокого занурення у імперативний стиль.

В рамках додаткової умови:

- Не використовувались вбудовані функції для сортування (замість цього було реалізовано бульбашкове сортування)
- Не використовувались динамічні структури даних
- Не використовувались set, map
- Не використовувались struct

Використовувалися вбудовані методи: зчитування тексту з файлу, порівняння рядків (equals, == не задовольняє) та compareTo у рядка.

Висновок

У ході лабораторної роботи було зроблено два завдання в суто імперативній парадигмі з виконанням додаткової умови на реалізацію алгоритму сортування та не використання структур даних.