

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Інститут комп’ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту

Лабораторна робота №1

з дисципліни

«Об’єктно-орієнтоване програмування»

Виконала:

студентка групи КН-108

Семич Тамара

Прийняла:

Грабовська Н.Р.

Львів – 2019 р.

Зміст

1. Тема лабораторної роботи.
2. Мета роботи.
3. Вимоги.
4. Задача.
5. Висновок.

Тема : Утилітарні класи Java SE. Обробка масивів і рядків. Інтерактивні консольні програми для платформи .

Мета:

- Розробка власних утилітарних класів.
- Набуття навичок вирішення прикладних задач з використанням масивів і рядків.
- Реалізація діалогового режиму роботи з користувачем в консольних програмах мовою Java.

Вимоги:

1. Розробити та продемонструвати консольну програму мовою Java в середовищі Eclipse для вирішення прикладної задачі за номером, що відповідає збільшеному на одиницю залишку від ділення на 15 зменшеного на одиницю номера студента в журналі групи.

2. Використовуючи програму рішення завдання відповідно до прикладної задачі забезпечити обробку команд користувача у вигляді текстового меню : а. введення даних; б. перегляд даних; с. виконання обчислень; d. відображення результату; е. завершення програми і т.д.

3. Забезпечити обробку параметрів командного рядка для визначення режиму роботи програми:

а. параметр “-h” чи “-help”: відображається інформація про автора програми, призначення (індивідуальне завдання), детальний опис режимів роботи (пунктів меню та параметрів командного рядка);

б. параметр “-d” чи “-debug”: в процесі роботи програми відображаються додаткові дані, що полегшують налагодження та перевірку працездатності програми: діагностичні повідомлення, проміжні значення змінних, значення тимчасових змінних та ін.

4. При вирішенні прикладних задач використовувати латинку.

5.Продемонструвати використання об’єктів класу StringBuilder або StringBuffer .

6. Забороняється використовувати засоби обробки регулярних виразів: класи пакету java.util.regex (Pattern , Matcher та ін.), а також відповідні методи класу .

Розробник: Семич Тамара, КН-108, номер варіанту індивідуального завдання- 9.

Задача: Ввести текст. Знайти та вивести, скільки разів повторюється в тексті кожне слово. Результат вивести у вигляді таблиці.

Опис програми:

Ця програма дозволяє обробити введений текст та дізнатися скільки разів в тексті повторюється кожне слово. Реалізовані 2 можливі режими роботи:

-h – режим при якому на початку програми з'являється інформація про розробника, суть індивідуального завдання, можливі функції консольного меню.

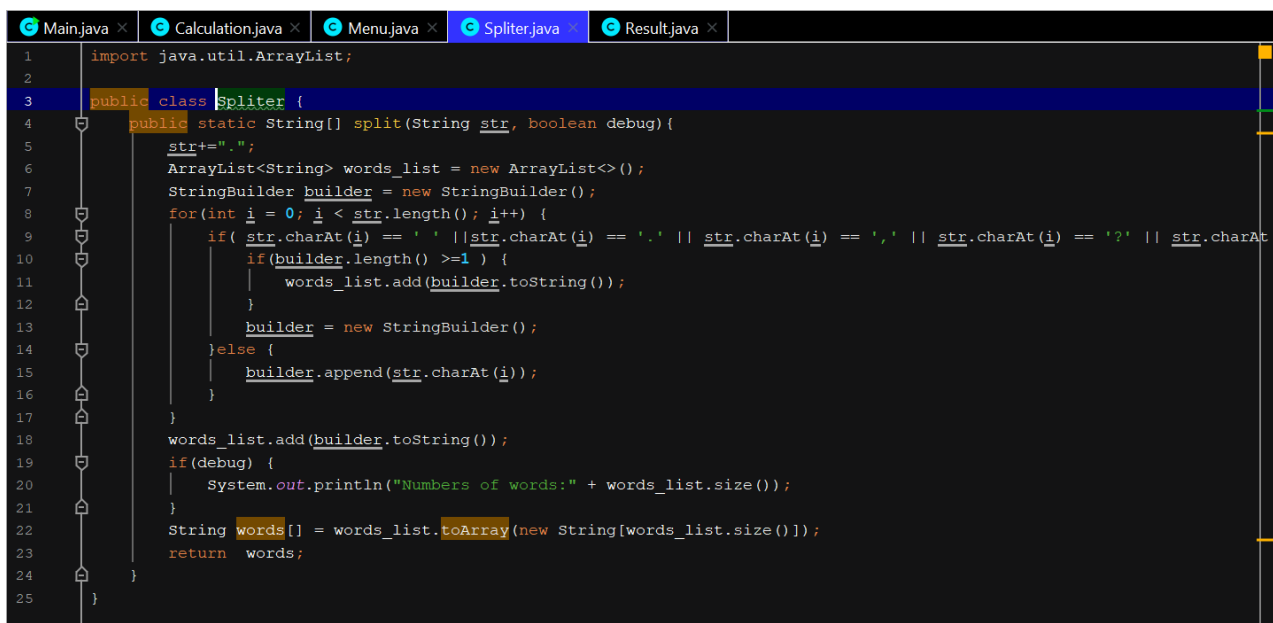
-d – спеціальний режим роботи, при якому користувачу виводить на екран хід виконання програми.

Ієрархія та структура об'єктів:

1. Клас Main, який забезпечує обробку параметрів, та викликає інші класи.
2. Клас Menu, який містить меню.
3. Клас Spliter, який забезпечує поділ тесту на слова (альтернатива методу split).
4. Клас Calculation, який рахує скільки раз повторюється кожне слово.
5. Клас Result, який виводить результат.

Важливі фрагменти коду:

Клас Spliter:



```
1 import java.util.ArrayList;
2
3 public class Spliter {
4     public static String[] split(String str, boolean debug) {
5         str+=".";
6         ArrayList<String> words_list = new ArrayList<>();
7         StringBuilder builder = new StringBuilder();
8         for(int i = 0; i < str.length(); i++) {
9             if( str.charAt(i) == ' ' || str.charAt(i) == '.' || str.charAt(i) == ',' || str.charAt(i) == '?' || str.charAt(i) == '\n' ) {
10                 if(builder.length() >=1 ) {
11                     words_list.add(builder.toString());
12                 }
13                 builder = new StringBuilder();
14             } else {
15                 builder.append(str.charAt(i));
16             }
17         }
18         words_list.add(builder.toString());
19         if(debug) {
20             System.out.println("Numbers of words:" + words_list.size());
21         }
22         String words[] = words_list.toArray(new String[words_list.size()]);
23         return words;
24     }
25 }
```

Клас Calculation:

```
Main.java × Calculation.java × Menu.java × Spliter.java × Result.java ×
1  import java.util.*;
2
3  public class Calculation {
4      public static <T> ArrayList<T> removeDuplicates(ArrayList<T> list)
5      {
6          Set<T> set = new LinkedHashSet<>();
7          set.addAll(list);
8          list.clear();
9          list.addAll(set);
10         return list;
11     }
12     public static HashMap<String,Integer> result(String str[], boolean debug){
13         HashMap<String, Integer> map = new HashMap<>();
14         ArrayList<String> words_list = new ArrayList<>();
15         for(int i =0; i < str.length-1; i++) {
16             words_list.add(str[i]);
17         }
18         removeDuplicates(words_list);
19         if(debug) {
20             System.out.println("Words without duplicates:" + words_list.size());
21         }
22         int count;
23         for(int i = 0; i < words_list.size(); i++) {
24             count = 0;
25             for(int j = 0; j < str.length-1; j++) {
26                 if( str[j].equals(words_list.get(i))) {
27                     count++;
28                 }
29             }
30
31             map.put(words_list.get(i), count);
32         }
33     }
34 }
```

Висновок: на цій лабораторній роботі я вивчила основні принципи об'єктно-орієнтованого програмування. Покращила навички розробки утилітарних класів (Клас Spliter), та вирішила з їхньою допомогою своє індивідуальне завдання.