

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота №1

з дисципліни

“Об’єктно-орієнтоване програмування”

Виконав:

студент групи КН-109

Коржов Володимир

Викладач:

Мочурад Л.І.

Лабораторна робота №1

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;
- д. 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);

b. 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. Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. `Helper Class`) та для обробки даних використовувати відповідні статичні методи.

7. Забороняється використовувати засоби обробки регулярних виразів: класи пакету `java.util.regex` (`Pattern` , `Matcher` та ін.), а також відповідні методи класу `String` (`matches` , `replace` , `replaceFirst` , `replaceAll` , `split`).

Інформація про розробника

– *Коржов Володимир Вікторович;*

– *студент групи КН-109;*

– *варіант №8*

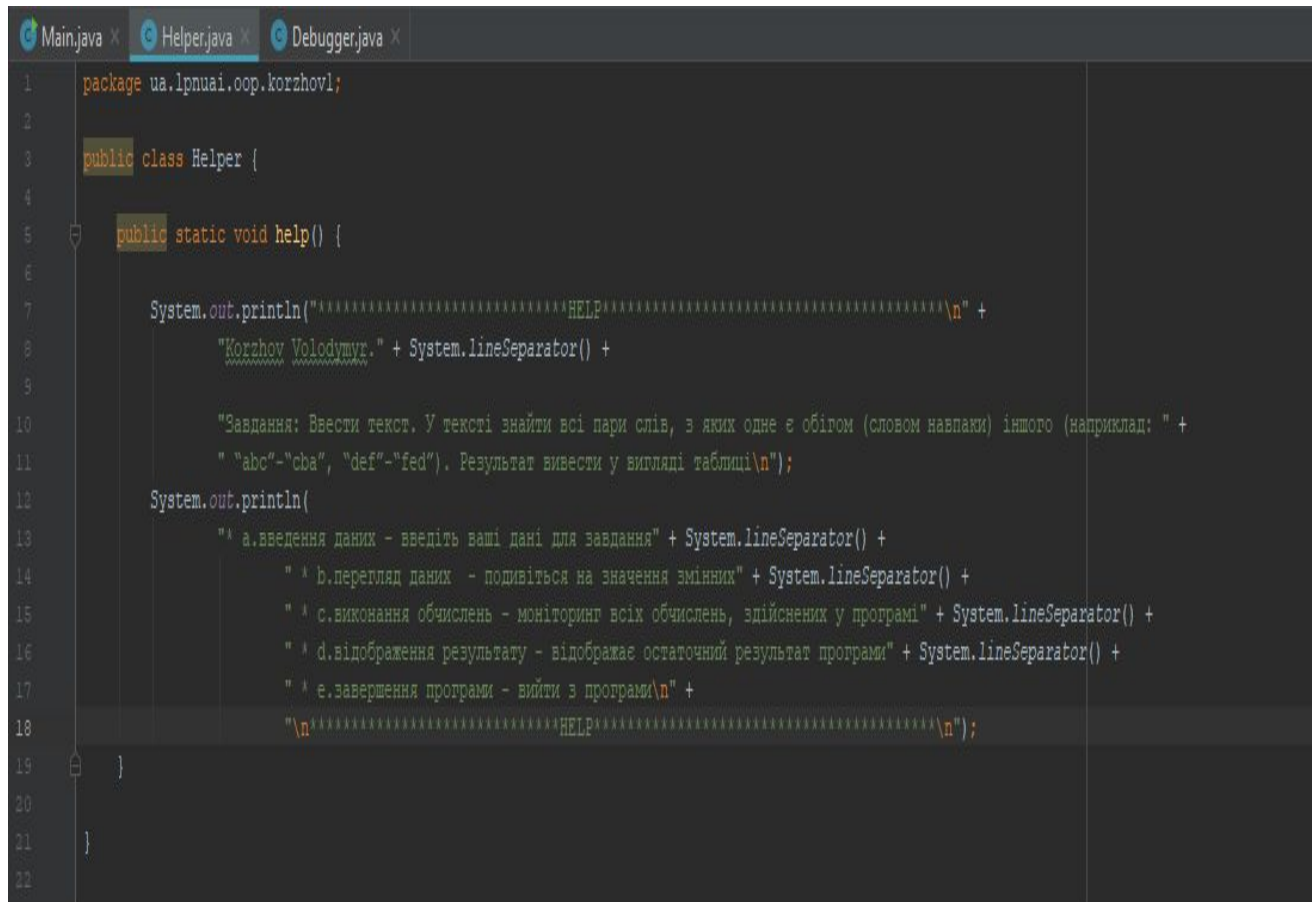
Задача

Ввести текст. У тексті знайти всі пари слів, з яких одне є обігом (словом наивпаки) іншого (наприклад: “abc”-“cba”, “def”-“fed”). Результат вивести у вигляді таблиці.

Опис класів

Я написав 3 класи: **Helper**, **Debugger** і **Main**.

У **Helper** написана загальна інформація про завдання, про його розробника та поради щодо навігації у інтерфейсі програми:



```
1 package ua.lpnuai.oop.korzhovl;
2
3 public class Helper {
4
5     public static void help() {
6
7         System.out.println("*****HELP*****\n" +
8             "Korzhov Volodymyr." + System.lineSeparator() +
9
10            "Завдання: Ввести текст. У тексті знайти всі пари слів, з яких одне є обігом (словом наизпак) іншого (наприклад: " +
11            " "abc"->"cba", "def"->"fed"). Результат вивести у вигляді таблиці\n");
12        System.out.println(
13            "  ^ а.введення даних - введіть ваші дані для завдання" + System.lineSeparator() +
14            "  ^ б.перегляд даних - подивіться на значення змінних" + System.lineSeparator() +
15            "  ^ в.виконання обчислень - моніторинг всіх обчислень, здійснених у програмі" + System.lineSeparator() +
16            "  ^ г.відображення результату - відображає остаточний результат програми" + System.lineSeparator() +
17            "  ^ ґ.завершення програми - вийти з програми\n" +
18            "\n*****HELP*****\n");
19    }
20
21 }
22
```

В **Debugger** відбувається перевірка вводу користувача і також можливий режим налаштування програми:

```
Main.java x Helper.java x Debugger.java x
1 package ua.lpnuai.oop.korzhovl;
2
3 public class Debugger {
4
5     public static void debug(String string, boolean debug, boolean var_show){
6
7         StringBuffer sb = new StringBuffer();
8         int i, j, n, counter = 0, wordLen = 0, count = 0, index = 0;
9
10        for(i = 0; i < string.length(); i++) {
11
12            if (debug || var_show) {
13                System.out.println("i = " + i);
14            }
15
16            if (string.charAt(i) != ' ') {
17                wordLen++;
18            }
19            else if(string.charAt(i) == ' '){
20
21                if (debug){
22                    System.out.println("wordLen = " + wordLen);
23                }
24                for (j = i + 1; j < string.length(); j++){
25                    if(var_show){
26                        System.out.println("wordLen = " + wordLen);
27                        System.out.println("j = " + j);
28                    }
29                }
30                if(string.charAt(j) != ' ') {
31
32                    for (n = j; n < string.length(); n++) {
33                        if(var_show){
34                            System.out.println("n = " + n);
35                        }
36                    }
37                    if (string.charAt(n) == ' ') {
38                        break;
39                    }
40                    counter++;
41                }
42                if (debug){
43                    System.out.println("word2Len = " + counter);
44                }
45                if(var_show){
46                    System.out.println("counter = " + counter);
47                }
48                if (counter == wordLen && counter != 0) {
49                    n--;
50                    for (int k = i - wordLen; k < i && n >= 0; k++, n--) {
51                        if(var_show){
52                            System.out.println("k = " + k + "\nn = " + n);
53                        }
54                    }
55                    if (string.charAt(k) == string.charAt(n)) {
56                        ...
57                    }
58                }
59            }
60        }
61    }
62}
```

Debugger > debug()

```
Main.java x Helper.java x Debugger.java x
52         count++;
53         if (debug){
54             System.out.println("Amount of same letters:\n"+count);
55         }
56         else if(var_show){
57             System.out.println("count = "+count);
58         }
59     } else {
60         j += count;
61         if (debug){
62             System.out.println("Wrong letter!");
63         }
64         break;
65     }
66 }
67 if (count == wordLen) {
68     if (debug){
69         System.out.println("Success!");
70     }
71     for (int l = i - wordLen; l < i; l++) {
72         index = l;
73         sb.insert( offset: index-1, string.charAt(l));
74         index++;
75     }
76     String res = new String(sb);
77     for(int m = 0; m < 4*index - 1; m++) {
78         System.out.print("-");
79     }
80     System.out.print("\n|" + res + "|" + sb.reverse()+"|\n");
81     sb.setLength(0);
82 }
83 wordLen = 0;
84 }
85 else if(counter != wordLen){
86     if(debug){
87         System.out.println("Wrong words!");
88     }
89 }
90 j+=counter-1;
91 counter = 0;
92 count = 0;
93 }
94 }
95 wordLen = 0;
96 }
97 }
98 for(int m = 0; m < 4*index - 1; m++) {
99     System.out.print("-");
100 }
101 System.out.println("\n");
102 }
```

Debugger > debug()

І у **Main** я пов'язую всі класи і також реалізую інтерфейс програми:

```
Main.java x Helper.java x Debugger.java x
1 package ua.lpnuai.oop.korzhovl;
2
3 import java.util.Scanner;
4
5 import static ua.lpnuai.oop.korzhovl.Debugger.debug;
6 import static ua.lpnuai.oop.korzhovl.Helper.help;
7
8 public class Main {
9
10     public static void main(String[] args) {
11
12         String input;
13         String str = "yeet";
14
15         do{
16             System.out.println("Виберіть вашу дію: \n");
17             System.out.println("a. введення даних;\n" +
18                 "b. перегляд даних;\n" +
19                 "c. виконання обчислень;\n" +
20                 "d. відображення результату;\n" +
21                 "e. завершення програми;\n" +
22                 "-h або help для додаткові інформації;\n" +
23                 "-d або debug - режим налаштування ВКЛ або ВИКЛ;\n");
24
25             Scanner in = new Scanner(System.in);
26             input = in.nextLine();
27
28             switch (input) {
29                 case "a":
30                     case "введення даних":
31                         Scanner c = new Scanner(System.in);
32                         str = c.nextLine();
33                         break;
34                 case "b":
35                     case "перегляд даних":
36                         if (str.equals("yeet")) {
37                             System.out.println("Немає даних!");
38                         } else {
39                             debug(str, debug: false, var_show: true);
40                         }
41                         break;
42                 case "c":
43                     case "виконання обчислень":
44                         if (str.equals("yeet")) {
45                             System.out.println("Немає даних!");
46                         } else {
47                             System.out.println("Обчислення виконуються....");
48                             debug(str, debug: false, var_show: false);
49                         }
50                         break;
51                 case "d":
52                     case "відображення результату":
```

```
30 case "введення даних":
31     Scanner c = new Scanner(System.in);
32     str = c.nextLine();
33     break;
34 case "b":
35 case "перегляд даних":
36     if (str.equals("yeet")) {
37         System.out.println("Немає даних!");
38     } else {
39         debug(str, debug: false, var_show: true);
40     }
41     break;
42 case "c":
43 case "виконання обчислень":
44     if (str.equals("yeet")) {
45         System.out.println("Немає даних!");
46     } else {
47         System.out.println("Обчислення виконуються...");
48         debug(str, debug: false, var_show: false);
49     }
50     break;
51 case "d":
52 case "відображення результату":
53     if (str.equals("yeet")) {
54         System.out.println("Немає даних!");
55     } else {
56         debug(str, debug: false, var_show: false);
57     }
58     break;
59 case "-h":
60 case "-help":
61     help();
62     break;
63 case "-d":
64 case "-debug":
65     if (str.equals("yeet")) {
66         System.out.println("Немає даних!");
67     } else {
68         debug(str, debug: true, var_show: true);
69     }
70     break;
71 case "e":
72 case "завершення програми":
73     System.out.println("ending...");
74     break;
75 default:
76     System.out.println("Неправильна команда!");
77 }
78 }while( !(input.equals("e") || (input.equals("завершення програми"))));
79 }
80 }
81 }
```


Результат роботи програми:

```
Виберіть вашу дію:
a. введення даних;
b. перегляд даних;
c. виконання обчислень;
d. відображення результату;
e. завершення програми
-h або help для додаткові інформації
-d або debug - режим налаштування ВКЛ або ВИКЛ

a
avvb a bvva
Виберіть вашу дію:
a. введення даних;
b. перегляд даних;
c. виконання обчислень;
d. відображення результату;
e. завершення програми
-h або help для додаткові інформації
-d або debug - режим налаштування ВКЛ або ВИКЛ

d
-----
|bvva|avvb|
-----

Виберіть вашу дію:
a. введення даних;
b. перегляд даних;
c. виконання обчислень;
d. відображення результату;
e. завершення програми
-h або help для додаткові інформації
-d або debug - режим налаштування ВКЛ або ВИКЛ

e
ending....
|
Process finished with exit code 0
```

Висновок: я розробив власні утилітарні класи, набув навичок вирішення прикладних задач з використанням масивів і рядків та реалізував діалоговий режим роботи з користувачем в консольних програмах мовою Java.