

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота №8
з дисципліни
«Об’єктно-орієнтовне програмування»

Виконала:
студента групи КН-110
Шваєнко Олександр
Викладач: Гасько
Р. Т.

Львів – 2018 р.

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 номеру студента в журналі групи з поверненням до початку. Наприклад 1 ->1, 2->2, ..., 15->15, 16->1, 17->2, ..., 30->15, 31-1 і т.д.
2. Використовуючи програму рішення завдання відповідно до прикладної задачі забезпечити обробку команд користувача у вигляді текстового меню :
 - a. введення даних;
 - b. перегляд даних;
 - c. виконання обчислень;
 - d. відображення результату;
 - e. завершення програми і т.д.
3. Забезпечити обробку параметрів командного рядка для визначення режиму роботи програми:
 - a. 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. Пропри продемонструвати використання об’єктів класу StringBuilder або StringBuffer (<http://docs.oracle.com/javase/8/docs/api/java/lang/...>) 6. Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. https://en.wikipedia.org/wiki/Helper_class) та для обробки даних використовувати відповідні статичні методи. <https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html>
7. Забороняється використовувати засоби обробки регулярних виразів: класи пакету java.util.regex (Pattern , Matcher та ін.), docs.oracle.com/javase/7/docs/api/java/util/regex/package-summary.html <https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>

<https://docs.oracle.com/javase/7/docs/api/java/util/regex/Matcher.html> а також відповідні методи класу String (matches , replace , replaceFirst , replaceAll , split) - docs.oracle.com/javase/7/docs/api/java/lang/String.html

Завдання:

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

Код:

```
package lab8;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        System.out.println("Hi! I can count the same words. For more information enter H or help.");

        String msg = null;

        System.out.print("Enter your command: ");
        Scanner scan = new Scanner(System.in);
        String comm = scan.nextLine();

        while (!("E".equals(comm))) {
            if ( "H".equals(comm) || "help".equals(comm)) {
                Helper help = new Helper();
                System.out.println(help.execute());
            }

            else if( "N".equals(comm)) {
                New text = new New();
                msg = text.execute(scan);
            }

            else if( "V".equals(comm)) {
                if (msg.equals(null))
                    System.out.println("No data.");
                else
                    System.out.println(msg);
            }

            else if ( "C".equals(comm)) {
                Count.execute(msg);
                System.out.println("Counted it.");
            }

            else if("R".equals(comm)) {
                System.out.println(Count.execute(msg));
            }
        }
    }
}
```

```

        else if (!("N".equals(comm)) || !("V".equals(comm)) ||
!("C".equals(comm))) {
            System.out.println("Sorry! the command wasn't found.");
        }
        System.out.print("Enter your command: ");
        comm = scan.nextLine();
    }

    scan.close();
    System.out.println("The program was succesfully closed.");

}
}

```

```
package lab8;
```

```
import java.util.Scanner;
```

```

public class New {
    public String execute(Scanner scan) {    System.out.print("Enter
your text: ");

        String text = scan.nextLine();

        return text;
    }
}

```

```
package lab8;
```

```

public class Helper implements Command {
    public String execute(){

        return ("Created by Tania Shevchuk. \n\n" +
"The program is designed to counts the same words in the
text.\n" +

        "Works with StringBuilder. \n\n"
+ "Choose your command:\n"
+ "N -  enter your data; \n"
+ "V -  view your text; \n"
+ "C -  count the same words; \n"
+ "R -  output result; \n"
+ "E -  close the program.");

    }

}

```

```
package lab8;
```

```

public class Count {
    public static String execute(String input) {

        String[] text = input.split(" ");

        StringBuilder build = new StringBuilder();

        int counter = 0;        boolean
was = false;        for (int i=0;
i<text.length; i++) {
            counter = 1;
            for (int k=i-1; k>=0; k--) {
                if (text[i].equals(text[k]))
                    was = true;
            }

            if (was == false) {
                for(int j=i+1; j<text.length; j++) {
                    if (text[i].equals(text[j]))
counter++;
                }
            }
            build.append(text[i] + " - ").append(counter + "\n");
        }

        return build.toString();
    }
}

package lab8;
public interface Command
{
    public String execute();
}

```

Результати роботи: