Науменко Д.

КІТ 101.8А

Лабораторна робота №3. Утилітарні класи. Обробка масивів і рядків

**Мета :**

1. Розробка власних утилітарних класів.

2.Набуття навичок вирішення прикладних задач з використанням масивів і рядків.

**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. При вирішенні прикладних задач використовувати [латинку](https://uk.wikipedia.org/wiki/%D0%9B%D0%B0%D1%82%D0%B8%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%B0%D0%B1%D0%B5%D1%82%D0%BA%D0%B0).
3. Продемонструвати використання об’єктів класу [StringBuilder](http://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html) або [StringBuffer](http://docs.oracle.com/javase/8/docs/api/java/lang/StringBuffer.html).
4. Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. [Helper Class](https://en.wikipedia.org/wiki/Helper_class)) та для обробки даних використовувати відповідні [статичні](https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html) методи.
5. Забороняється використовувати засоби обробки регулярних виразів: класи пакету [java.util.regex](https://docs.oracle.com/javase/7/docs/api/java/util/regex/package-summary.html) ([Pattern](https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html), [Matcher](https://docs.oracle.com/javase/7/docs/api/java/util/regex/Matcher.html) та ін.), а також відповідні методи класу [String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html) ([matches](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#matches(java.lang.String)), [replace](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#replace(java.lang.CharSequence,%20java.lang.CharSequence)), [replaceFirst](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#replaceFirst(java.lang.String,%20java.lang.String)), [replaceAll](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#replaceAll(java.lang.String,%20java.lang.String)), [split](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#split(java.lang.String,%20int))).

**Прикладна задача** : Ввести текст. У тексті знайти всі пари слів, з яких одне є обігом (словом навпаки) іншого (наприклад: “abc”-“cba”, “def”-“fed”).

**Опис програми** : текст вводиться з клавіатури. Застосовано stringbuilder для порівняння рядків, які може являтися паліндромом, а може і ні.

**Код програми** :

public class SecondLab {

public static void main(String[] args) {

Scanner in = new Scanner (System.in);

System.out.println("Input string: ");

String name = in.nextLine();

String test = "dimon";

System.out.printf("Your String: %s\n", name);

in.close();

func (name,test);

}

public static void func (String name,String test)

{

StringBuilder res = new StringBuilder(test);

if(name.contains(res.reverse()))

{

System.out.println("Your reverse word: ");

System.out.println(res);

}

}

}

**Робота програми** :

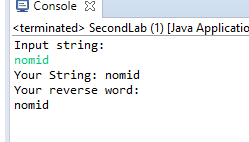


Рис. 1. – Робота програми.

**Висновок** : на лабораторній роботі ми набули навичок вирішення прикладних задач з використанням масивів і рядків