

Міністерство освіти і науки України  
Національний технічний університет України „КПІ”  
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки  
інформації та управління

## **ЗВІТ**

до лабораторної роботи № 3  
з предмету:

„ОСНОВИ ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ”

**Виконав  
студент**

*ІП-61 Кушка Михайло  
Олександрович, 2-й курс, ІП-6116*

---

(№ групи, прізвище, ім'я, по батькові, курс, номер  
залікової книжки)

**Прийняв**

*Подрубайло О.О.*

---

(посада, прізвище, ім'я, по батькові )

Київ 2018

## **ЗМІСТ**

<b>1. ПОСТАНОВКА ЗАДАЧІ .....</b>	<b>3</b>
<b>2. ВИСНОВОК.....</b>	<b>4</b>
<b>3. КОД ПРОГРАМИ .....</b>	<b>5</b>

# 1. ПОСТАНОВКА ЗАДАЧІ

1. Визначити  $C_3$  як остачу від ділення номера залікової книжки студента на 3,  $C_{13}$  як остачу від ділення номера залікової книжки студента на 17.
2. В залежності від  $C_3$  визначити тип текстових змінних:

$C_3$	Тип
0	StringBuilder
1	StringBuffer
2	String

3. В залежності від  $C_{17}$  визначити дію з рядком:

$C_{17}$	Дія з рядком
0	Знайти найбільшу кількість речень заданого тексту, в яких є однакові слова.
1	Вивести всі речення заданого тексту в порядку зростання кількості слів у них.
2	Знайти таке слово в першому реченні заданого тексту, якого не має в жодному з наступних.
3	В усіх питальних реченнях заданого тексту знайти та надрукувати без повторень слова заданої довжини.
4	В кожному реченні заданого тексту змінити місцями перше та останнє слово, не змінивши довжини речення.
5	Надрукувати слова без повторень заданого тексту в алфавітному порядку за першою літерою.
6	Відсортувати слова заданого тексту за зростанням кількості голосних літер.
7	Відсортувати слова заданого тексту, що починаються з голосних літер, за другою літерою.
8	Відсортувати слова заданого тексту за зростанням кількості в них заданої літери.
9	Задано текст та масив слів. Підрахувати у скількох реченнях зустрічається кожне слово масиву.
10	З кожного речення заданого тексту видалити підрядок найбільшої довжини, що починається та закінчується заданими літерами.
11	Із заданого тексту видалити всі слова визначеної довжини, що починаються з приголосної літери.
12	Відсортувати слова заданого тексту за кількістю входжень визначеного символу в них.
13	В заданому тексті знайти підрядок максимальної довжини, що є паліндромом, тобто читається однаково зліва на право та з права на ліво.
14	В кожному слові заданого тексту, видалити всі наступні входження першої літери цього слова.
15	В заданому тексті замінити слова заданої довжини визначеним рядком.
16	В кожному слові заданого тексту, видалити всі попередні входження останньої літери цього слова.

4. Створити клас, який складається з виконавчого методу, що виконує дію текстовим рядком (п.3), тип якого визначено варіантом (п.2). Необхідно обробити всі виключні ситуації, що можуть виникнути під час виконання програмного коду. Всі змінні повинні бути описані та значення їх задані у виконавчому методі.

$$6116 \% 3 == 2 \ (C_3 == 2)$$

$$6116 \% 17 == 13 \ (C_{17} == 13)$$

## **2. ВИСНОВОК**

Найважчим в даній лабораторній роботі було розробити алгоритм пошуку поліндруму в рядку, оскільки алгоритм виявився не зовсім очевидним. Складнощі з програмуванням на Java в даній роботі не виникли.

### 3. КОД ПРОГРАММЫ

```
/**
 * Java labs - Lab3
 * @version 1.1 2018-03-04
 * @author Misha Kushka
 */

class MyString {
    private final String str;

    MyString(String newStr) {
        str = newStr;
    }

    void print() {
        if (this.str.length() == 0) {
            System.err.println("The string is empty");
        } else {
            System.out.println("String:\n" + this.str);
        }
    }

    boolean checkPalindrome(String s) {
        int n = s.length();
        for (int i = 0; i < n / 2; ++i) {
            if (s.charAt(i) != s.charAt(n-i-1))
                return false;
        }

        return true;
    }

    String longestPalindrome() {
        int left = 0;
        final int right = this.str.length();
        int j = right;
        String temp = "";
        String theLongest = "";

        while (left < right-1) {
            temp = this.str.substring(left, j);
            j -= 1;

            if (this.checkPalindrome(temp) && theLongest.length() < temp.length())
            {
                theLongest = temp;
            }

            if (j < left+2) {
                left += 1;
            }
        }
    }
}
```

```

        j = right;
    }
}

    return theLongest;
}

}

public class Main {

    public static void main(String[] args) {
        String str = "lsfjlsdabbasrmrsljdljjddd";
        MyString s = new MyString(str);
        s.print();

        // Find the longest palindrome
        String longest = s.longestPalindrome();

        if (longest.length() == 0) {
            System.err.println("There is no polindrom in the line (lenght of
polindrome must be > 1)");
        } else {
            System.out.println("\nThe longest palindrome: " + longest);
        }
    }

}

```