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

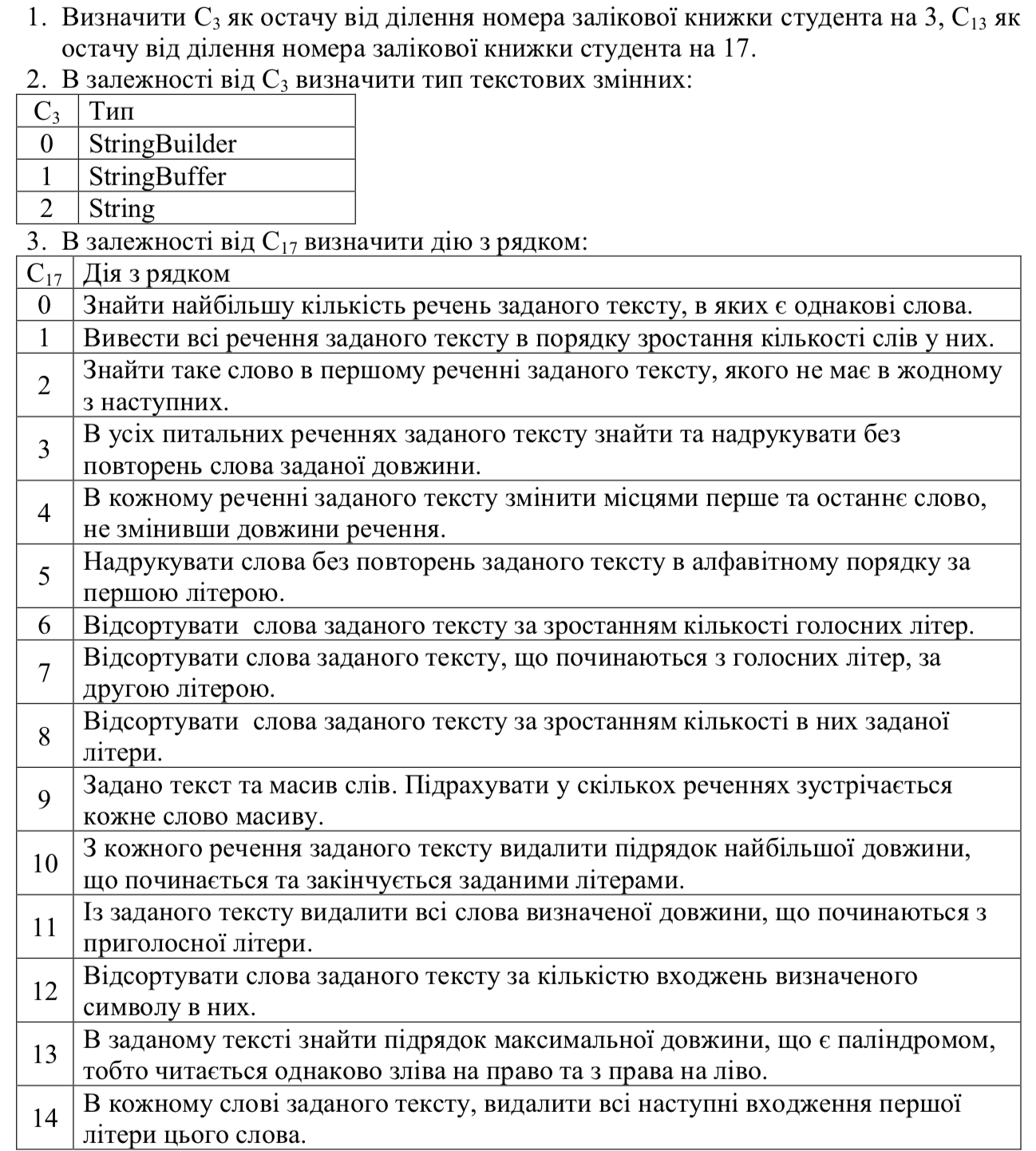
ЗМІСТ

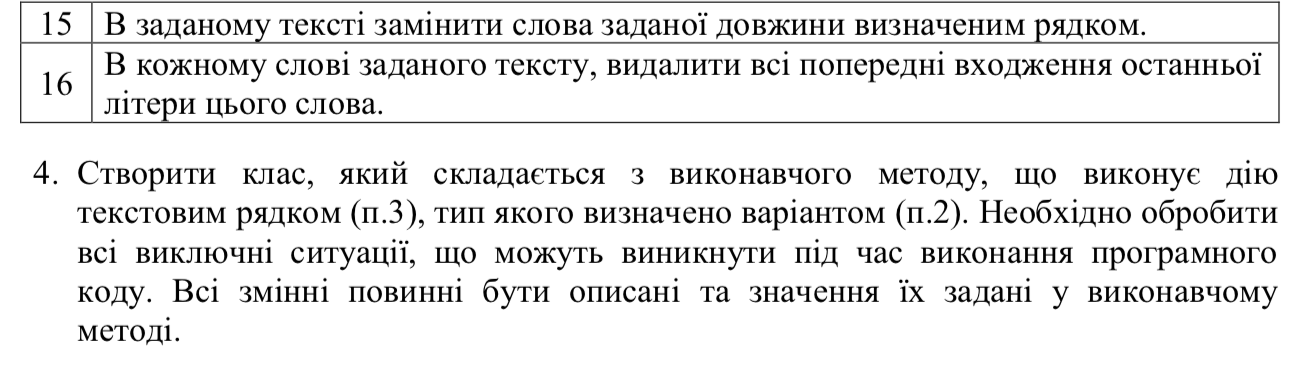
[1. постановка задачі 3](#_Toc506126329)

[2. Висновок 4](#_Toc506126330)

[3. Код програми 5](#_Toc506126331)

# постановка задачі





6116 % 3 == 2 ()

6116 % 17 == 13 ()

# Висновок

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

# Код програми

/\*\*

\* 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);

}

}

}