

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ  
КАФЕДРА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

ЛАБОРОТОРНА РОБОТА №3  
з дисципліни «ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»  
на тему “Робота зі строками”

Студента 2 курсу групи АД-181

Гежа Н. І.

Перевірив

доцент Рудніченко Н. Д.

Одеса - 2019

3MCT

## Введення

Ціллю цієї роботи є ознайомлення з класом `String` та використання його для збереження і роботи над текстом.

## Теоретична частина

Символьний рядок в Java є послідовність символів в юнікодi. В Java відсутній вбудований тип для символьних рядків. Замість цього, в стандартній бібліотеці Java міститься клас String. Кожний символьний рядок, укладений в лапки, являє собою екземпляр класу String.

За допомогою методу `substring()` класу String можна виділити підрядок з окремого рядка. Другий параметр методу `substring()` позначає позицію символу, який не слід включати до складу підрядка. Довжина нового підрядка буде дорівнює  $b - a$ , де  $a$  - перший параметр `substring()`,  $b$  - другий параметр. Щоб перевірити дві символьні рядки на рівність, досить викликати метод `equals()`.

В Java надається можливість об'єднати два символьні рядки, використовуючи знак `+`. Щоб отримати довжину рядка, використовується метод `length()`. Щоб отримати певний символ в рядку, використовується метод `charAt(index)`, де `index` - індекс символу в рядку.

В Java не можна змінювати окремі символи в рядку, тому об'єкти типу String називаються незмінними (`immutable`). Так що, якщо потрібно замінити символьний рядок у змінній з «Hello» на «Help», цього не можна домогтися простою заміною символів. Для цього потрібно створити новий рядок, яка буде містити незмінну частину старої + змінна частина.

### Завдання

1. Реалізуйте метод, який приймає на вхід рядок і повертає true, якщо рядок закінчується на «ed» і false в іншому випадку.
2. Реалізуйте метод, який приймає на вхід рядок і повертає суму чисел, які були знайдені в цьому рядку (якщо цифр немає - повертається 0).
3. Реалізуйте метод, який приймає на вхід рядок і повертає довжину найдовшого «блоку» символів в цьому рядку («блок» - декілька однакових символів підряд, наприклад, в рядку «aaBCS» - найдовший блок «aa» довжиною 2 символи)
4. Реалізуйте метод, який приймає на вхід рядок і виводить в консоль знайдені слова в цьому рядку (слова відокремлюються пробілами, крім випадків коли слово стоїть на початку або в кінці)
5. Реалізуйте метод, який приймає на вхід два рядки (А і Б) і повертає рядок, яка складається по черзі з і-их символів кожного з рядків. Наприклад, спочатку перший символ першого рядка, потім перший символ другого рядка, потім другий символ першого рядка, потім другий символ другого рядка і т.д. Якщо символи в одному з рядків закінчаться, що залишилися символи іншого рядка дописують в кінець

## Кід програми

```

package AD181.Gezha;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        String input = Main.getString("Input a character string");
        if (Main.stringEndsWith(input, "ed")) {
            System.out.println("The string ends with 'ed'");
        } else {
            System.out.println("The string does not end with 'ed'");
        }

        input = Main.getString("Input a string of digits");
        System.out.println("Sum of digits = " + Main.sumDigitsInString(input));

        input = Main.getString("Input a string of characters");
        System.out.println("Length of the longest block: " +
Main.getMaxBlockLength(input));

        input = Main.getString("Input a string with words");
        printWordsInString(input);

        String input1 = Main.getString("Input the first string");
        String input2 = Main.getString("Input the second string");
        String mergedString = Main.getMergedString(input1, input2);
        System.out.println("Merged string:\n" + mergedString);
    }

    private static boolean stringEndsWith(String string, String ending) {
        return string.endsWith(ending);
    }

    private static int sumDigitsInString(String string) {
        int sum = 0;
        for (int charIndex = 0; charIndex < string.length(); ++charIndex) {
            int charValue = Character.getNumericValue(string.charAt(charIndex));
            sum += charValue;
        }
        return sum;
    }

    private static int getMaxBlockLength(String string) {
        int currentBlockLength = 0;
        int maxBlockLength = 0;
        char lastBlockCharacter = '\0';
        for (int charIndex = 0; charIndex < string.length(); ++charIndex) {
            char currentCharacter = string.charAt(charIndex);
            if (currentCharacter != lastBlockCharacter) {
                currentBlockLength = 0;
            }

            ++currentBlockLength;
            lastBlockCharacter = currentCharacter;

            if (currentBlockLength > maxBlockLength) {
                maxBlockLength = currentBlockLength;
            }
        }
    }
}

```

```

    }
    return maxBlockLength;
}

private static void printWordsInString(String string) {
    System.out.println("Separate words in the entered string:");
    int wordStartIndex = 0;
    int wordEndIndex = 0;
    for (int charIndex = 0; charIndex < string.length(); ++charIndex) {
        if (string.charAt(charIndex) == ' ') {
            // this "if" prevents errors with multiple whitespaces
            if (wordStartIndex != wordEndIndex) {
                System.out.println(string.substring(wordStartIndex, wordEndIndex));
            }
            wordStartIndex = charIndex + 1;
        }

        wordEndIndex = charIndex + 1;
    }
    System.out.println(string.substring(wordStartIndex, wordEndIndex));
}

private static String getMergedString(String string1, String string2) {
    int maxLength = Math.max(string1.length(), string2.length());

    // builds the merged string from both strings
    StringBuilder resultingString = new StringBuilder();
    for (int charIndex = 0; charIndex < maxLength; ++charIndex) {
        if (charIndex < string1.length()) {
            resultingString.append(string1.charAt(charIndex));
        }

        if (charIndex < string2.length()) {
            resultingString.append(string2.charAt(charIndex));
        }
    }

    return resultingString.toString();
}

private static String getString(String prompt) {
    Scanner console = new Scanner(System.in);
    System.out.println(prompt);
    String line = "";
    do {
        line = console.nextLine();
    } while (line.length() == 0);

    return line;
}
}

```

## Вивід програми

Програма була запущена 2 рази з різними вхідними параметрами (Рис. 1, Рис. 2). У обох випадках вивід програми був правильний.

```
Input a character string
hello, underworld
The string does not end with 'ed'
Input a string of digits
12002101111
Sum of digits = 10
Input a string of characters
qqwwWwertYYYY
Length of the longest block: 4
Input a string with words
a string with words
Separate words in the entered string:
a
string
with
words
Input the first string
good morning
Input the second string
good evening
Merged string:
ggooooodd meovrenniinnngg
```

Рис. 1: Вивід 1

```
Input a character string
decimated
The string ends with 'ed'
Input a string of digits
123456789
Sum of digits = 45
Input a string of characters
oOoowo
Length of the longest block: 2
Input a string with words
Single
Separate words in the entered string:
Single
Input the first string
the first string
Input the second string
not the first string
Merged string:
tnhoet ftihres tf isrtsrti nsstring
```

Рис. 2: Вивід 2



### Висновок

Було проведено ознайомлення з класом String на мові Java, та його практичне застосування для збереження та обробки тексту.

Посилання на github репозиторій: <https://github.com/onpu-ad181ng/gezha-oop-lab3>

### Список літератури

1. Приклади та інструкції з опису лабораторної роботи.
2. Форум Stackoverflow