МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВПО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №1

«Знакомство с языком программирования Java и средой разработки Eclipse»

по дисциплине

«Разработка программных систем»

Выполнила студентка группы ИВТ-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Седов М.Д./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Долженкова М.Л./

Киров 2020

1. Задание

Разработать библиотеку статических методов для работы со строками. Должны быть реализованы следующие публичные методы:

* public static isPrefix(String str, String sub) – определяет, является ли строка sub префиксом строки str;
* public static isSuffix(String str, String sub) – определяет, является ли строка sub суффиксом строки str;
* public static isSubstring(String str, String sub) – определяет, является ли строка sub подстрокой строки str;
* public static isSubsequence(String str, String sub) – определяет, является ли строка sub подпоследовательностью строки str.

1. Листинг программы

Листинг программы приведен в приложении А.

1. Экранные формы программы

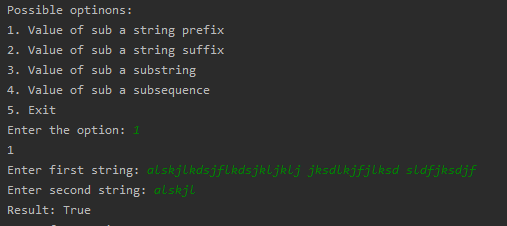


Рисунок 1 – Пример выполнения метода isPrefix

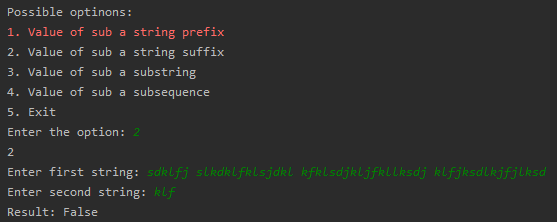


Рисунок 2 – Пример выполнения метода isSuffix

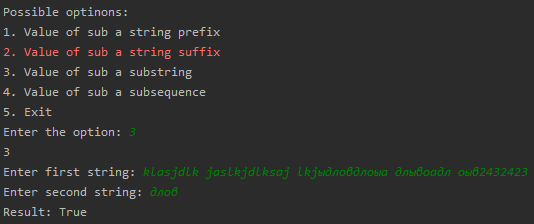


Рисунок 3 – Пример выполнения метода isSubstring

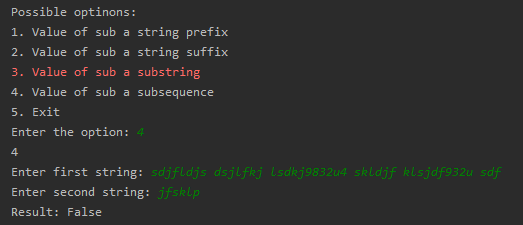


Рисунок 3 – Пример выполнения метода isSubsequence

1. Вывод

В ходе выполнения лабораторной работы были изучены основные конструкции языка программирования Java, структура программы, стандартные средства ввода/вывода, работа с классами, а также создание библиотеки в виде переносимого jar архива.; изучен основной функционал интегрированный среды разработки IntelliJ IDEA; написана библиотека для работы со строками

Приложение А

(обязательное)

Листинг программы

Main.java

package com.max;

import max.tools.MyString;

import java.util.InputMismatchException;

import java.util.Scanner;

public class Main {

static final String RED = "\033[0;31m";

static final String RESET = "\033[0m";

static void mainMenu(int index) {

System.out.println("Possible optinons:");

String options[] = {"1. Value of sub a string prefix\n",

"2. Value of sub a string suffix\n",

"3. Value of sub a substring\n",

"4. Value of sub a subsequence\n",

"5. Exit\n"};

if (index != -1) {

for (int i = 0; i < index; i++)

System.out.print(RESET + options[i]);

System.out.print(RED + options[index]);

for (int i = index + 1; i < options.length; i++)

System.out.print(RESET + options[i]);

} else {

for (int i = 0; i < options.length; i++)

System.out.print(RESET + options[i]);

}

System.out.print("Enter the option: ");

}

public static void main(String[] args) {

int currentPos = -1;

while(true) {

System.out.println("=======================================");

mainMenu(currentPos);

Scanner scanner = new Scanner(System.in);

try {

int code = scanner.nextInt();

System.out.println(code);

currentPos = code - 1;

if (code == 5) break;

System.out.print("Enter first string: ");

scanner.nextLine();

String str = scanner.nextLine();

System.out.print("Enter second string: ");

String sub = scanner.nextLine();

boolean result = true;

switch (code) {

case 1:

result = MyString.isPrefix(str, sub);

break;

case 2:

result = MyString.isSuffix(str, sub);

break;

case 3:

result = MyString.isSubstring(str, sub);

break;

case 4:

result = MyString.isSubsequence(str, sub);

break;

}

System.out.printf("Result: %s\n", result ? "True" : "False");

System.out.print("Enter for continue: ");

scanner.nextLine();

}

catch (InputMismatchException e) {

continue;

}

}

}

}

MyString.java

package max.tools;

public class MyString {

public static boolean isPrefix(String str, String sub) {

int n = str.length(), m = sub.length();

if (m > n) return false;

for (int i = 0; i < m; i++) {

if (sub.charAt(i) != str.charAt(i)) return false;

}

return true;

}

public static boolean isSuffix(String str, String sub) {

int n = str.length(), m = sub.length();

if (m > n) return false;

for (int i = m - 1; i >= 0; i--) {

if (sub.charAt(i) != str.charAt(i + n - m)) return false;

}

return true;

}

public static boolean isSubstring(String str, String sub) {

int m = sub.length(), n = str.length();

if (m > n) return false;

for (int i = 0; i <= n - m; i++) {

if (str.substring(i, i + m).equals(sub)) return true;

}

return false;

}

public static boolean isSubsequence(String str, String sub) {

int n = str.length(), m = sub.length();

int p = 0, t = 0, cnt = 0;

while (p < m && m - p <= n - t) {

cnt++;

if (sub.charAt(p) == str.charAt(t))

p++;

t++;

}

return p == sub.length();

}

}