|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

**Отчет по лабораторной работе 7**

***по дисциплине «Языки программирования для работы с большими данными»***

Студент ИУ6-21М **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гришина А. М.**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Степанов П. В.**

(Подпись, дата) (И.О.Фамилия)

*2020 г.*

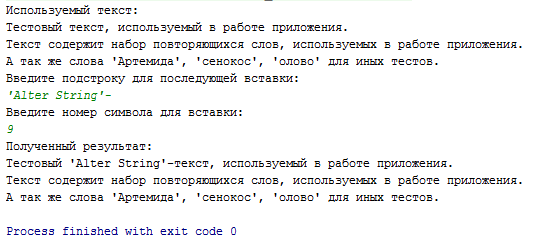
Вариант 1

1. В тексте после k-го символа вставить заданную подстроку.

**Код**:

**package** com.company;  
  
**import** java.util.Scanner;  
  
**public class** Main {  
  
 **public static void** main(String[] args) {  
  
 Scanner cmd = **new** Scanner(System.***in***);  
  
 String base =  
 **"Тестовый текст, используемый в работе приложения.\n"** +  
 **"Текст содержит набор повторяющихся слов, используемых в работе приложения.\n"** +  
 **"А так же слова 'Артемида', 'сенокос', 'олово' для иных тестов."**;  
  
 System.***out***.println(**"Используемый текст:\n"** + base);  
 System.***out***.println(**"Введите подстроку для последующей вставки:"**);  
 String alt = cmd.nextLine();  
 System.***out***.println(**"Введите номер символа для вставки:"**);  
 **int** k = cmd.nextInt();  
 StringBuffer sb = **new** StringBuffer(base);  
 sb.insert(k, alt);  
 System.***out***.println(**"Полученный результат:\n"** + sb);  
  
 }  
}

**Проверка правильности работы:**



**Вывод**: Программа в тексте после k-го символа вставляет заданную подстроку. Программа работает корректно.

Вариант 2

6. Напечатать без повторения слова текста, у которых первая и последняя буквы совпадают.

**Код**:

**package** com.company;

**import** java.util.Scanner;

**public class** Main {

**public static void** main(String[] args) {

Scanner cmd = **new** Scanner(System.***in***);

String base =

**"Тестовый текст, используемый в работе приложения.\n"** +

**"Текст содержит набор повторяющихся слов, используемых в работе приложения.\n"** +

**"А так же слова 'Артемида', 'сенокос', 'олово' для иных тестов."**;

System.***out***.println(**"Используемый текст:\n"** + base);

System.***out***.println(**"\nСлова, в которых первая и последняя буква совпадают:"**);

String[] S = base.split(**"[\\s.,'\"!?0-9]"**);

**for**(String s : S){

**if** (s.length() > 1 && s.charAt(0) == s.charAt(s.length()-1))

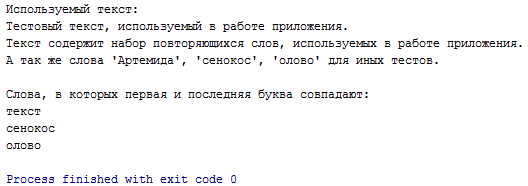
System.***out***.println(s);

}

}

}

**Проверка правильности работы:**



**Вывод**: Программа выводит без повторения слова текста, у которых первая и последняя буквы совпадают. Программа работает корректно.

Вариант 3

1. Найти такое слово в первом предложении, которого нет ни в одном из остальных предложений.

**Код**:

**package** com.company;

**import** java.util.Scanner;

**public class** Main {

**public static void** main(String[] args) {

Scanner cmd = **new** Scanner(System.***in***);

String base =

**"Тестовый текст, используемый в работе приложения.\n"** +

**"Текст содержит набор повторяющихся слов, используемых в работе приложения.\n"** +

**"А так же слова 'Артемида', 'сенокос', 'олово' для иных тестов."**;

System.***out***.println(**"Используемый текст:\n"** + base);

System.***out***.println(**"\nСлова первого предложения, отсутствующие в последующих:"**);

String[] S1 = base.split(**"[.!?]"**);

String[] S2 = S1[0].split(**"[\\s.,'\":;!@#$%^&()]"**);

**for**(String s : S2) {

**boolean** f = **false**;

**for** (**int** i = 1; i < S1.**length**; i++)

**if** (S1[i].toLowerCase().contains(s.toLowerCase())) {

f = **true**;

**break**;

}

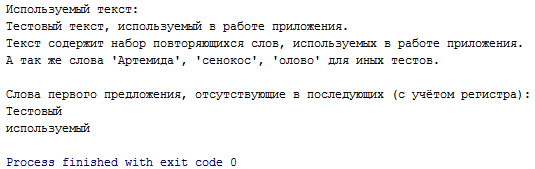
**if** (!f) System.***out***.println(s);

}

}

}

**Проверка правильности работы:**



**Вывод**: Программа находит такое слово в первом предложении, которого нет ни в одном из остальных предложений. Программа работает корректно.

Вариант 4

1. Подсчитать, сколько раз заданное слово входит в текст.

**Код**:

**package** com.company;

**import** java.util.Scanner;

**public class** Main {

**public static void** main(String[] args) {

Scanner cmd = **new** Scanner(System.***in***);

String base =

**"Тестовый текст, используемый в работе приложения.\n"** +

**"Текст содержит набор повторяющихся слов, используемых в работе приложения.\n"** +

**"А так же слова 'Артемида', 'сенокос', 'олово' для иных тестов."**;

System.***out***.println(**"Используемый текст:\n"** + base);

System.***out***.print(**"\nВведите искомое слово:"**);

String f = cmd.next();

String[] S = base.split(**"[\\s.,'\":;!@#$%^&()]"**);

**int** counter = 0;

**for**(String s : S) {

**if** (s.toLowerCase().equals(f.toLowerCase()))

counter++;

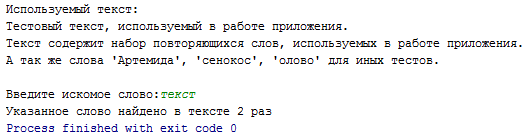
}

System.***out***.printf(**"Указанное слово найдено в тексте %d раз"**, counter);

}

}

**Проверка правильности работы:**



**Вывод**: Программа считает, сколько раз заданное слово входит в текст. Программа работает корректно.