|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Математического обеспечения и стандартизации информационных технологий | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2** | |
| **по дисциплине** | |
| **«**Объектно-ориентированное программирование**»**  **Тема: «Строки в Java. Применение регулярных выражений для поиска заданных слов»** | |
|  | |
| Выполнил студент группы ИКБО-07-18 | Мишанов А.Б. |
| Принял преподаватель | Колесников С.Н. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_201\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. | *(подпись руководителя)* |

**Задание 1**

Применение класса String.

**Требования к заданию:**

1. Создать приложение - основную программу, которое должно иметь интерфейс для демонстрации работы всех трех заданий (два последних пункта будут реализовываться по мере отладки предыдущих). В каждом пункте создаются свои объекты, для решения задач задания.
2. Создать класс с именем, определяющим номер задания, для решения задач варианта.
   1. Данными класса должен быть входной текст и дополнительные строки для промежуточных вычислений.
   2. Основные операции указаны в варианте. Их надо оформить методами созданного класса. Если операция содержит отдельную задачу, то ее так же оформить отдельным методом.
   3. При разработке методов класса использовать объект и методы класса String.
3. Выполнить тестирование класса и приложения.

|  |  |
| --- | --- |
| 4 | Дан художественный текст. По количеству используемых в тексте слов можно с некоторой вероятностью определить автора. Равными словами считать слова, содержащие строчные и прописные буквы (например, сЛоВо и СЛОВО и слово).   * Преобразовать все слова к одному формату: либо строчные буквы, либо прописные. * Сформировать массив из слов. * Сформировать массив вхождений каждого найденного слова в текст. * Определить, сколько слов начинаются с заданного префикса. |

**Вариант 4:**

**Тестирование:**

Тестирование всех функций проводилось на строке: «Шел грека через реку видит греКА В РЕЧке РАК».

Тестирование функции приведения всех символов в нижний регистр:

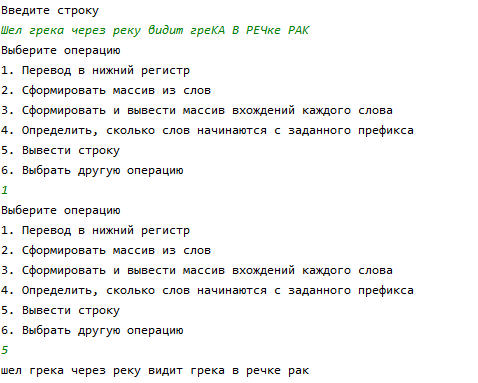


Рисунок 1 – Тестирование функции перевода все символов в нижний регистр

Тестирование функции формирования массива слов, из которых состоит строка:

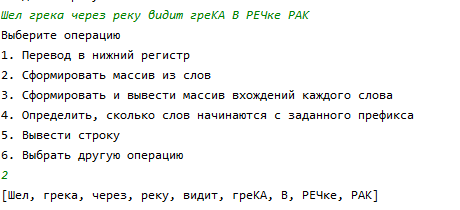


Рисунок 2 – Тестирование функции формирования массива слов

Тестирование функции формирования массива вхождений каждого слова:

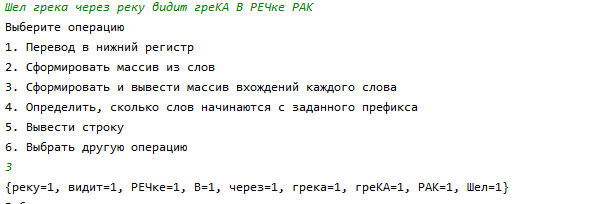


Рисунок 3 – Тестирование функции формирования массива вхождений каждого слова

Тестирование функции определения слов с данным префиксом

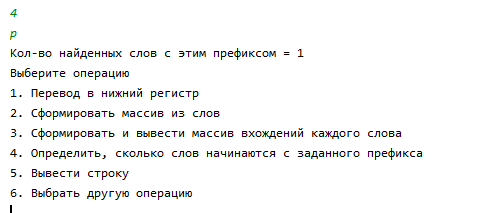


Рисунок 4 – Тестирование функции определения слов с данным префиксом

**Задание 2:**

Применение класса StringBuilder

**Требования к заданию:**

1. Разработать класс для решения задачи варианта.
   1. Данными класса должен быть входной текст и дополнительные строки для промежуточных вычислений.
   2. Основные операции указаны в варианте. Их надо оформить методами созданного класса. Если операция содержит отдельную задачу, то ее так же оформить отдельным методом.
   3. При разработке методов своего класса использовать объекты и методы класса StringBuffer.
2. Реализовать второй пункт консольного интерфейса приложения, созданного в задании 1, включив алгоритм демонстрации выполнения задач варианта.
3. Выполнить тестирование класса и приложения.

|  |  |
| --- | --- |
| 4 | Дан текст, состоящий из предложений, заканчивающиеся символами конца предложений. Предложение состоит из слов. Словами могут быть числа.   * Найти в тексте двузначные числа и заменить их строкой из символов + длиной равной двузначному числу. * Добавить в самое короткое предложение текста одно слово КОНЕЦ. * Найти трехзначные числа и заменить из обратными числами, т.е.число 123 заменить на 321. |

**Тестирование:**

Тестирование всех функций проводилось на строке: «Короткое предложение. Не самое короткое предложение 5. Предложение с двухзначным числом 13. Предложение с трехзначным числом 123».

Тестирование функции нахождения двухзначных чисел и их заменой на плюсы:

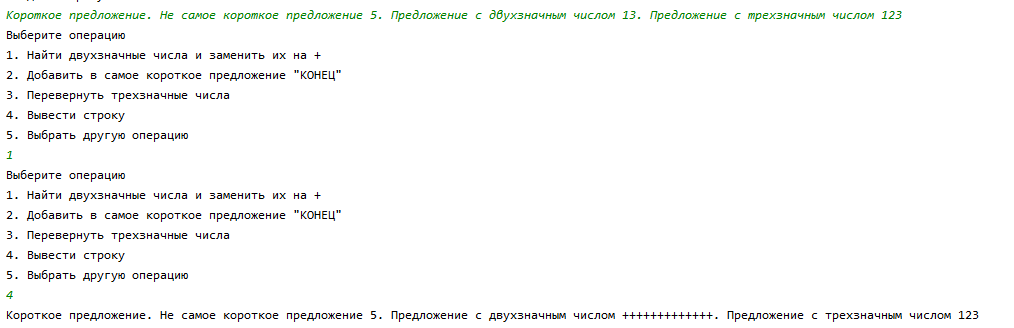


Рисунок 5 – Тестирование функции

Тестирование функции добавления слова «КОНЕЦ» в конец самого короткого предложения

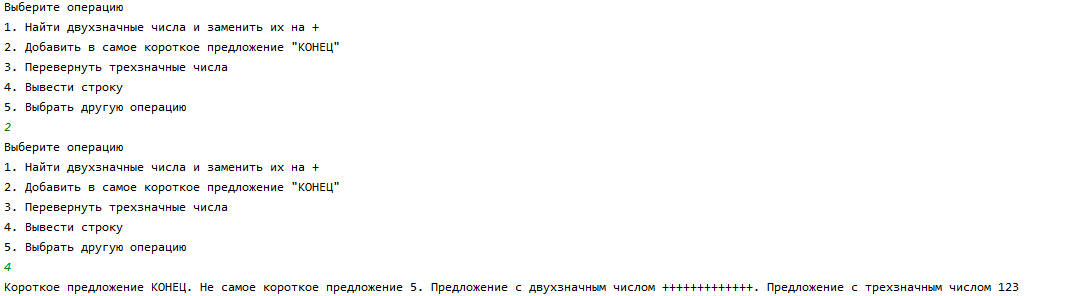


Рисунок 6 – Тестирование функции добавления слова «КОНЕЦ» в конец самого короткого предложения

Тестирование функции переворота трехзначных чисел:

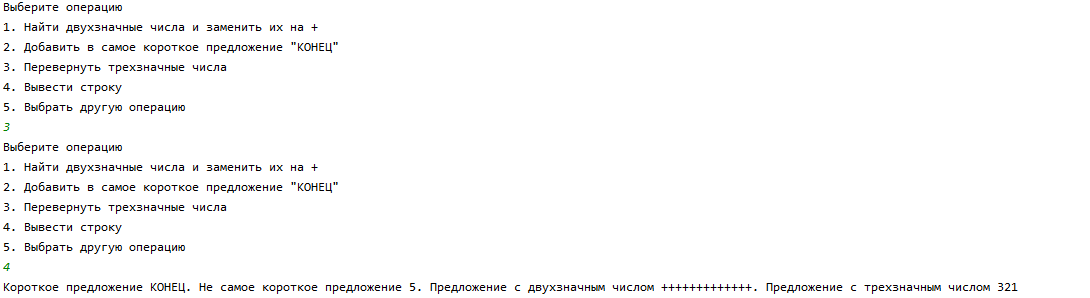


Рисунок 7 – Тестирование функции переворота трехзначных чисел

**Задание 3:**

Регулярные выражения

**Требования к заданию:**

1. Разработать класс для решения задач варианта.
2. Разработать регулярное выражение в соответствии с первой задачей варианта. Разработать метод, реализующий проверку строки на соответствие регулярному выражению и протестировать метод.
3. Разработать метод для решения второй задачи варианта. Выполнить его тестирование.
4. Реализовать третий пункт консольного интерфейса приложения, созданного в задании 1, включив алгоритм демонстрации выполнения задач варианта.

|  |  |
| --- | --- |
| 3 | 1. Определите, что строка является корректной датой в русском и американском формате. 2. Дан текст. Замените все даты в американском формате, датами в русском формате. |

**Тестирование:**

Тестирование функции определения корректности даты в определенном формате для даты «25.02.2015»:

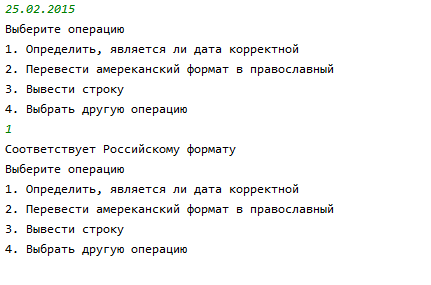


Рисунок 8 –Тестирование функции определения корректности даты в определенном формате

Тестирование функции перевода дат в строке в российский формат:

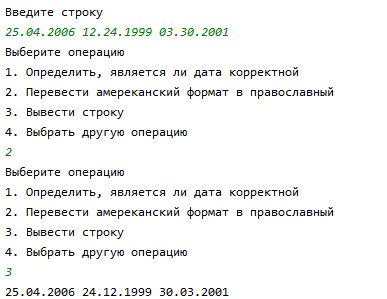


Рисунок 9 – Тестирование функции перевода дат в строке в российский формат

**Исходный код**

Класс String1

**package** com.company;  
**import** java.util.HashMap;  
  
  
**public class** String1 {  
 String **str**;  
 HashMap<String,Integer> **MyMap**;  
 String1(){  
 **str**=**""**;  
 **MyMap** = **new** HashMap<String,Integer>();  
 }  
 String1(String s){  
 **str**=s;  
 **MyMap** = **new** HashMap<String,Integer>();  
 }  
 **public void** toLCase(){  
 **str**=**str**.toLowerCase();  
 }  
  
 **public** String[] SplitReg(){  
 String[] words = **str**.split(**" "**);  
 **return** words;  
 }  
 **public int** PrefCounter(String prefix){  
 **int** count=0;  
 String[] words= **str**.split(**" "**);  
  
 **for** (**int** i=0;i<words.**length**;i++){  
 **if** (words[i].startsWith(prefix))  
 count++;  
 }  
 **return** count;  
 }  
 **public void** WordCounter(){  
 String[] wordArray=**this**.SplitReg();  
 **for** (**int** i=0;i<wordArray.**length**;i++){  
 **if**(**MyMap**.containsKey(wordArray[i])){  
 **MyMap**.put(wordArray[i],**MyMap**.get(wordArray[i])+1);  
 }  
 **else  
 MyMap**.put(wordArray[i],1);  
 }  
 System.***out***.println(**MyMap**.toString());*//Упорно делаем вид, что метод возвращает словарь, а словарь потом выводится в main* }  
 **public** String out(){  
 **return** (**str**);  
 }  
}

Класс String2

**package** com.company;  
  
**public class** String2 {  
 StringBuffer **str**;  
 String2(){  
 **str**=**new** StringBuffer(**"Стандартное предложение."**);  
 }  
 String2(StringBuffer userString){  
 **str**=userString;  
 }  
 String2(String userString){  
 **str**=**new** StringBuffer(userString);  
 }  
 **public void** AppendEnd(){  
 **int** min=0, count=0,indexOfMin=-1;  
 **for** (**int** i=0; i<**str**.length();i++){  
 **if** (**str**.charAt(i)==**'.'** || **str**.charAt(i)==**'?'** || **str**.charAt(i)==**'!'**){  
 **if** (min==0){  
 min=count;  
 indexOfMin=i;  
 }  
 **if** (min>count){  
 min=count;  
 indexOfMin=i;  
 }  
 count=0;  
 }  
 count++;  
 }  
 **str**.insert(indexOfMin,**" КОНЕЦ"**);  
 }  
 **public void** InsertPlus(){  
 **int** counter=0;  
 **for** (**int** i=0; i<**str**.length();i++){  
 **if** (Character.*isDigit*(**str**.charAt(i))){  
 **while** (Character.*isDigit*(**str**.charAt(i)) && i<**str**.length()-1){  
 counter++;  
 i++;  
 }  
 **if** (counter==2 && i!=**str**.length()-1){  
 String pluses=**""**;  
 **int** number=Integer.*parseInt*(**str**.substring(i-2,i));  
 **for** (**int** j=0;j<number;j++)  
 pluses=pluses+**"+"**;  
 **str**.replace(i-2,i,pluses);  
 }  
 counter=0;  
 }  
 }  
 **if** (Character.*isDigit*(**str**.charAt(**str**.length()-1)) && Character.*isDigit*(**str**.charAt(**str**.length()-2))&& !Character.*isDigit*(**str**.charAt(**str**.length()-3))){  
 String pluses=**""**;  
 **int** number=Integer.*parseInt*(**str**.substring(**str**.length()-2,**str**.length()));  
 **for** (**int** j=0;j<number;j++)  
 pluses=pluses+**"+"**;  
 **str**.replace(**str**.length()-2,**str**.length(),pluses);  
 }  
 }  
 **public void** ReversNumber(){  
 **int** counter=0;  
 **for** (**int** i=0; i<**str**.length();i++){  
 **if** (Character.*isDigit*(**str**.charAt(i))){  
 **while** (Character.*isDigit*(**str**.charAt(i)) && i<**str**.length()-1){  
 counter++;  
 i++;  
 }  
 **if** (counter==3 && i!=**str**.length()-1){  
 StringBuffer number=**new** StringBuffer(**str**.substring(i-3,i)).reverse();  
 **str**.replace(i-3,i,number.toString());  
 }  
 counter=0;  
 }  
 }  
 **if** (Character.*isDigit*(**str**.charAt(**str**.length()-1)) && Character.*isDigit*(**str**.charAt(**str**.length()-2))&& Character.*isDigit*(**str**.charAt(**str**.length()-3))&& !Character.*isDigit*(**str**.charAt(**str**.length()-4))){  
 StringBuffer number=**new** StringBuffer(**str**.substring(**str**.length()-3,**str**.length())).reverse();  
 **str**.replace(**str**.length()-3,**str**.length(),number.toString());  
 }  
 }  
 **public** String out(){  
 **return str**.toString();  
 }  
}

Класс String3

**package** com.company;  
  
**import** java.util.regex.Matcher;  
**import** java.util.regex.Pattern;  
  
**public class** String3 {  
 String **regAmerDate**;  
 String **regRusDate**;  
 String3(){  
 **regAmerDate**=**"([0][1-9]|[1][0-2])(\\.|-)([0][1-9]|[12][0-9]|[3][01])(\\.|-)[0-2][0-9][0-9][0-9]"**;  
 **regRusDate**=**"([0][1-9]|[12][0-9]|[3][01])(\\.|-)([0][1-9]|[1][0-2])(\\.|-)[0-2][0-9][0-9][0-9]"**;  
 }  
 **public** String Check(String str){  
 String res=**""**;  
 **if** (Pattern.*matches*(**regAmerDate**,str))  
 res=**"Соответствует Американскому формату\n"**;  
 **if** (Pattern.*matches*(**regRusDate**,str))  
 res+=**"Соответствует Российскому формату"**;  
 **if** (res==**""**)  
 **return "Не соответствует ни одному из форматов"**;  
 **return** res;  
 }  
 **public** String ReplaceAmer(String s){  
 StringBuffer str= **new** StringBuffer(s);  
 Pattern pattern=Pattern.*compile*(**regAmerDate**);  
 Matcher matcher=pattern.matcher(str);  
 **while** (matcher.find()){  
 **int** start=matcher.start();  
 **int** end=matcher.end();  
 StringBuffer tempStr= **new** StringBuffer(str.substring(start,end)); String t1=tempStr.substring(0,2), t2=tempStr.substring(3,5);  
 tempStr.replace(0,2,t2); tempStr.replace(3,5,t1);  
 str.replace(start,end,tempStr.toString());  
 }  
 **return** str.toString();  
 }  
}

Класс Main

**package** com.company;  
**import** java.util.Scanner;  
**import** java.util.Arrays;  
**import** java.util.regex.Pattern;  
  
**public class** Main {  
  
 **public static void** main(String[] args) {  
 **int** n = 0, num = 0;  
 String str;  
 Scanner in = **new** Scanner(System.***in***);  
 System.***out***.println(**"Выберите задание"**);  
 System.***out***.println(**"1. Задание №1\n2. Задание №2\n3. Задание №3"**);  
 **while** (num != 7) {  
  
 num = in.nextInt();  
 **switch** (num) {  
 **case** 1:  
 n = 0;  
 System.***out***.println(**"Введите строку"**);  
 in.nextLine();  
 str = in.nextLine();  
 String1 str1 = **new** String1(str);  
 **while** (n != 6) {  
 System.***out***.println(**"Выберите операцию\n1. Перевод в нижний регистр"**);  
 System.***out***.println(**"2. Сформировать массив из слов\n3. Сформировать и вывести массив вхождений каждого слова"**);  
 System.***out***.println(**"4. Определить, сколько слов начинаются с заданного префикса\n5. Вывести строку\n6. Выбрать другую операцию"**);  
 n = in.nextInt();  
 **switch** (n) {  
 **case** 1:  
 str1.toLCase();  
 **break**;  
 **case** 2:  
 String[] strArray = str1.SplitReg();  
 System.***out***.println(Arrays.*toString*(strArray));  
 **break**;  
 **case** 3:  
 str1.WordCounter();  
 **break**;  
 **case** 4:  
 in.nextLine();  
 String prefix = in.nextLine();  
 System.***out***.println(**"Кол-во найденных слов с этим префиксом = "** + str1.PrefCounter(prefix));  
 **break**;  
 **case** 5:  
 System.***out***.println(str1.out());  
 **break**;  
 **default**:  
 System.***out***.println(**"Неверное значение"**);  
 **break**;  
 }  
 }  
 **break**;  
 **case** 2:  
 n = 0;  
 System.***out***.println(**"Введите строку"**);  
 in.nextLine();  
 str = in.nextLine();  
 String2 str2 = **new** String2(str);  
 **while** (n != 5) {  
  
 System.***out***.println(**"Выберите операцию\n1. Найти двухзначные числа и заменить их на +"**);  
 System.***out***.println(**"2. Добавить в самое короткое предложение \"КОНЕЦ\"\n3. Перевернуть трехзначные числа"**);  
 System.***out***.println(**"4. Вывести строку\n5. Выбрать другую операцию"**);  
 n = in.nextInt();  
 **switch** (n) {  
 **case** 1:  
 str2.InsertPlus();  
 **break**;  
 **case** 2:  
 str2.AppendEnd();  
 **break**;  
 **case** 3:  
 str2.ReversNumber();  
 **break**;  
 **case** 4:  
 System.***out***.println(str2.out());  
 **break**;  
 **default**:  
 *// System.out.println("Неверно");* }  
 }  
  
 **break**;  
 **case** 3:  
 n = 0;  
 System.***out***.println(**"Введите строку"**);  
 in.nextLine();  
 str = in.nextLine();  
 String3 str3 = **new** String3();  
 **while** (n != 4) {  
 System.***out***.println(**"Выберите операцию\n1. Определить, является ли дата корректной"**);  
 System.***out***.println(**"2. Перевести амереканский формат в православный\n3. Вывести строку"**);  
 System.***out***.println(**"4. Выбрать другую операцию"**);  
 n = in.nextInt();  
 **switch** (n) {  
 **case** 1:  
 System.***out***.println(str3.Check(str));  
 **break**;  
 **case** 2:  
 str = str3.ReplaceAmer(str);  
 **break**;  
 **case** 3:  
 System.***out***.println(str);  
 ;  
 **break**;  
 }  
 }  
 **break**;  
  
  
 }  
 }  
 }  
}