Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 9

тема «Символы и строки в Java»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-22-1б Носов Д.О.

Проверил: ассистент каф. ВММБ Нетбай Г.В.

Пермь, 2023

**Содержание**

[Задание 1 5](#_Toc134795130)

[1.1. Постановка задачи 5](#_Toc134795131)

[1.2. Решение задачи, код программы 5](#_Toc134795132)

[1.3. Тестирование работы программы с проверкой 5](#_Toc134795133)

[Задание 2 8](#_Toc134795134)

[2.1. Постановка задачи 8](#_Toc134795135)

[2.2. Решение задачи, код программы 8](#_Toc134795136)

[2.3. Тестирование работы программы с проверкой 8](#_Toc134795137)

[Задание 3 10](#_Toc134795138)

[3.1. Постановка задачи 10](#_Toc134795139)

[3.2. Решение задачи, код программы 10](#_Toc134795140)

[3.3. Тестирование работы программы с проверкой 10](#_Toc134795141)

[Задание 4 12](#_Toc134795142)

[4.1. Постановка задачи 12](#_Toc134795143)

[4.2. Решение задачи, код программы 12](#_Toc134795144)

[4.3. Тестирование работы программы с проверкой 12](#_Toc134795145)

[Задание 5 16](#_Toc134795146)

[5.1. Постановка задачи 16](#_Toc134795147)

[5.2. Решение задачи, код программы 16](#_Toc134795148)

[5.3. Тестирование работы программы с проверкой 16](#_Toc134795149)

[Задание 6 18](#_Toc134795150)

[6.1. Постановка задачи 18](#_Toc134795151)

[6.2. Решение задачи, код программы 18](#_Toc134795152)

[6.3. Тестирование работы программы с проверкой 19](#_Toc134795153)

[Задание 7 24](#_Toc134795154)

[7.1. Постановка задачи 24](#_Toc134795155)

[7.2. Решение задачи, код программы 24](#_Toc134795156)

[7.3. Тестирование работы программы с проверкой 24](#_Toc134795157)

[Задание 8 26](#_Toc134795158)

[8.1. Постановка задачи 26](#_Toc134795159)

[8.2. Решение задачи, код программы 26](#_Toc134795160)

[8.3. Тестирование работы программы с проверкой 26](#_Toc134795161)

[Задание 9 28](#_Toc134795162)

[9.1. Постановка задачи 28](#_Toc134795163)

[9.2. Решение задачи, код программы 28](#_Toc134795164)

[9.3. Тестирование работы программы с проверкой 28](#_Toc134795165)

[Задание 10 30](#_Toc134795166)

[10.1. Постановка задачи 30](#_Toc134795167)

[10.2. Решение задачи, код программы 30](#_Toc134795168)

[10.3. Тестирование работы программы с проверкой 30](#_Toc134795169)

[Задание 11 32](#_Toc134795170)

[11.1. Постановка задачи 32](#_Toc134795171)

[11.2. Решение задачи, код программы 32](#_Toc134795172)

[11.3. Тестирование работы программы с проверкой 32](#_Toc134795173)

[Задание 12 34](#_Toc134795174)

[12.1. Постановка задачи 34](#_Toc134795175)

[12.2. Решение задачи, код программы 34](#_Toc134795176)

[12.3. Тестирование работы программы с проверкой 34](#_Toc134795177)

# Задание 1

## 1.1. Постановка задачи

Дана матрица символов размером N×M. Найти сколько раз по строчкам и по столбцам встречается «no».

## 1.2. Решение задачи, код программы

public class zad\_1 {  
 public void zad\_1() {  
 MyMass arr = new MyMass();  
 System.*out*.println("Задайте матрицу символов: ");  
 String[][] mas = arr.GetStrMas\_x2();  
 int[] no = *no\_count*(mas);  
 arr.PntStrMas\_x2(mas);  
 System.*out*.printf("\nСлово <<no>> встречается %d раз(а) в строке и %d раз в столбце",no[0],no[1]);  
 }  
  
 public static int[] no\_count(String[][] mas) {  
 int[] arr = new int[2];  
 for (int j = 0; j < mas.length; j++) {  
 for (int i = 0; i < mas[0].length - 1; i++) {  
 if (mas[j][i].equals("n") && mas[j][i + 1].equals("o")) {  
 arr[0]++;  
 }  
 if (mas[i][j].equals("n") && mas[i + 1][j].equals("o")) {  
 arr[1]++;  
 }  
 }  
 }  
 return arr;  
 }  
}

## 1.3. Тестирование работы программы с проверкой

Таблица 1.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 2

## 2.1. Постановка задачи

Дан массив символов, содержащий число в 12-ой системе счисления. Проверить правильность ввода этого числа (в его записи должны быть только символы 0, 1, 2,…, 9, A, B). Если число введено неверно, сообщить об этом пользователю и повторить ввод, ввод должен повторяться до тех пор, пока не будет введено число. При правильном вводе перевести число в 16-ую систему счисления, записать число в новый массив символов.

## 2.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class zad\_2 {  
 public void zad\_2() {  
 Scanner scn = new Scanner(System.*in*);  
 boolean fl = false;  
 while (!fl) {  
 System.*out*.print("Введите число в 12-ой системе счисления: ");  
 String input = scn.nextLine().toUpperCase();  
 if (input.matches("[0**-**9A**-**B]+")) {  
 fl = true;  
 String hexString = Integer.*toHexString*(Integer.*parseInt*(input, 12));  
 System.*out*.println("Введенное число в 16-ой системе счисления: " + hexString.toUpperCase());  
 } else {  
 System.*out*.println("Ошибка ввода. В числе могут быть только символы 0-9 и A-B.");  
 }  
 }  
 }  
}

## 2.3. Тестирование работы программы с проверкой

Таблица 2.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 3

## 3.1. Постановка задачи

Дан массив символов. Проанализировать массив и ответить на вопросы:

а) Определить количество латинских прописных букв.

б) Выяснить, входит ли в последовательность буква, введенная с клавиатуры, например «*ю»*, регистр не учитывать*.*

в) Выяснить, верно ли, что среди символов имеются две и более запятые*.*

г) Выяснить, имеется ли среди символов пара соседствующих букв «*во»* или «*ов».*

д) Выяснить, имеется ли среди символов пара соседствующих одинаковых цифр.

е) Выяснить, верно ли. что существуют такие натуральные *i* и *j,* что *1 <i <j < п* и что *si*, *si+1*это одинаковые буквы отличающиеся регистром, a *sj,* *sj+1* это 0.

## 3.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class zad\_3 {  
 public void zad\_3() {  
 MyMass arr = new MyMass();  
 System.*out*.println("Задайте массив символов: ");  
 String[] mas = arr.GetStrMas();  
 Scanner scn = new Scanner(System.*in*);  
 System.*out*.printf("Задайте проверяемую латинскую букву: ");  
 String n = scn.next();  
 *Check*(*Stats*(mas,n),n);  
 }  
  
 public static void Check(int[] mas, String n) {  
 System.*out*.printf("\nРезультаты:\n");  
 System.*out*.printf("Прописных латинских букв %d штук \n", mas[0]);  
 if (mas[1] == 1) { System.*out*.printf("Буква <<%s>> встречается в массиве\n", n);  
 } else { System.*out*.printf("Буква <<%s>> не встречается в массиве\n", n);}  
 if (mas[2] >= 2){ System.*out*.println("Запятая встречается в массиве 2-а или более раз");  
 } else { System.*out*.println("Запятая встречается в массиве менее 2-х раз");}  
 if (mas[3] == 1) { System.*out*.println("Среди элементов встречается <<во>> или <<ов>>");  
 } else { System.*out*.println("Среди букв не встречается <<во>> или <<ов>>");}  
 if (mas[4] == 1) { System.*out*.println("Среди элементов встречаются одинаковый цифры стоящие подряд");  
 } else { System.*out*.println("Среди элементов не встречаются одинаковый цифры стоящие подряд");}  
 if (mas[5] == 1) { System.*out*.println("Среди элементов встречаются одинаковые буквы " +  
 "разного регистра и нули стоящие подряд");  
 } else { System.*out*.println("Среди элементов не встречаются одинаковые буквы " +  
 "разного регистра и нули стоящие подряд");}  
 }  
  
 public static int[] Stats(String[] mas, String n) {  
 int arr[] = new int[6];  
 for(int i = 0; i < mas.length; i++){  
 if (mas[i].matches("[а**-**яА**-**Я]+")){  
 if (mas[i].matches("[А**-**яA-z]+")) {  
 arr[0]++;  
 }  
 if((mas[i].toUpperCase()).equals(n.toUpperCase())) {  
 arr[1] = 1;  
 }  
 if((i != mas.length - 1) && ((mas[i].equals("в") && mas[i+1].equals("о"))  
 || (mas[i].equals("о") && mas[i+1].equals("в")))){  
 arr[3] = 1;  
 }  
 if(i > 1 && i != mas.length-1 && !mas[i].equals(mas[i+1]) && (mas[i].toUpperCase().equals(mas[i+1])  
 || mas[i].toLowerCase().equals(mas[i+1]))) {  
 for (int j = i + 1; j < mas.length - 1; j++) {  
 if (mas[j].equals("0") && mas[j + 1].equals("0")) {  
 arr[5] = 1;  
 }  
 }  
 }  
 }else if (mas[i].equals(",")){  
 arr[2]++;  
 }else if (i != mas.length - 1 && mas[i].matches("[0**-**9]+")  
 && mas[i+1].matches("[0**-**9]+") && mas[i].equals(mas[i+1])){  
 arr[4] = 1;  
 }  
 }  
 return arr;  
 }  
}

## 3.3. Тестирование работы программы с проверкой

Таблица 3.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 4

## 4.1. Постановка задачи

Задана матрица символов N×N. Определить построчно и по столбцам, есть ли в них выражения вида («число» + или - «число») / («число» + или - «число») = «число». Вывести выражения данного вида по строкам и по столбцам. Проверить есть ли правильно вычисленные выражения, и подсчитать их количество. Создать ArrayList и записать в него выражения, которые были вычислены не правильно с корректировкой ответа. Вывести информацию на экран.

## 4.2. Решение задачи, код программы

import java.util.ArrayList;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class zad\_4 {  
 public void zad\_4() {  
 MyMass arr = new MyMass();  
  
 System.*out*.println("Задайте массив символов: ");  
 String[][] mas = arr.GetStrMas\_x2();  
 if (mas.length < 13 || mas[0].length < 13) {  
 System.*out*.println("В массиве недостаточно символов!");  
 } else {  
 System.*out*.println(*Array*(mas));  
 }  
 }  
  
 public static ArrayList<String> Array(String[][] mas) {  
 ArrayList<String> list1 = new ArrayList<>();  
 ArrayList<String> list = *List*(mas);  
 Pattern pattern = Pattern.*compile*("[+-]|\\d+");  
 for (String i : list) {  
 Matcher matcher = pattern.matcher(i);  
 while (matcher.find()) {  
 list1.add(i.substring(matcher.start(), matcher.end()));  
 }  
 int[] mas1 = *Check*(list1);  
 if (mas1[2] != Integer.*parseInt*(list1.get(6))){  
 System.*out*.printf("Выражение %s неверно!\n",i);  
 String tr = Integer.*toString*(mas1[2]);  
 list.set(list.indexOf(i),"("+list1.get(0)+list1.get(1)+list1.get(2)+")/("  
 +list1.get(3)+list1.get(4)+list1.get(5)+")="+tr);  
 }  
 list1.clear();  
 }  
 System.*out*.printf("\nИтоговый ArrayList: \n");  
 return list;  
 }  
  
 public static int[] Check(ArrayList<String> list) {  
 int[] mas = new int[3];  
 if (list.get(1).equals("+")) {  
 mas[0] = Integer.*parseInt*(list.get(0)) + Integer.*parseInt*(list.get(2));  
 if (list.get(4).equals("+")) {  
 mas[1] = Integer.*parseInt*(list.get(3)) + Integer.*parseInt*(list.get(5));  
 } else {  
 mas[1] = Integer.*parseInt*(list.get(3)) - Integer.*parseInt*(list.get(5));  
 }  
 } else {  
 mas[0] = Integer.*parseInt*(list.get(0)) - Integer.*parseInt*(list.get(2));  
 if (list.get(4).equals("+")) {  
 mas[1] = Integer.*parseInt*(list.get(3)) + Integer.*parseInt*(list.get(5));  
 } else {  
 mas[1] = Integer.*parseInt*(list.get(3)) - Integer.*parseInt*(list.get(5));  
 }  
 }  
 mas[2] = mas[0]/mas[1];  
 return mas;  
 }  
  
 public static ArrayList<String> List(String[][] mas) {  
 Pattern pattern = Pattern.*compile*("\\(\\d+[+-]\\d+\\)/\\(\\d+[+-]\\d+\\)=\\d+");  
 ArrayList<String> list = new ArrayList<>();  
 String text1 = "";  
 String text2 = "";  
 for (int j = 0; j < mas.length; j++) {  
 for (int i = 0; i < mas[0].length; i++) {  
 text1 += mas[j][i];  
 text2 += mas[i][j];  
 }  
 Matcher matcher1 = pattern.matcher(text1);  
 while (matcher1.find()){  
 list.add(text1.substring(matcher1.start(), matcher1.end()));  
 }  
 Matcher matcher2 = pattern.matcher(text2);  
 while (matcher1.find()){  
 list.add(text2.substring(matcher2.start(), matcher2.end()));  
 }  
 text1 = "";  
 text2 = "";  
 }  
 return list;  
 }  
}

## 4.3. Тестирование работы программы с проверкой

Таблица 4.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 5

## 5.1. Постановка задачи

Дана строка, зашифровать строку заменой символов: a – @, d – $, n – №, z – \*. Вывести исходный и зашифрованный текст.

## 5.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class zad\_5 {  
 public void zad\_5() {  
 Scanner scn = new Scanner(System.*in*);  
 System.*out*.println("Задайте строку: ");  
 String text = scn.next();  
 System.*out*.printf("\nТекущая строка:\n%s\n\nИзмененная строка:\n",text);  
 System.*out*.println(*Replace\_adnz*(text));  
 }  
  
 public static String Replace\_adnz(String text) {  
 String replace = text.replace('a','@');  
 replace = replace.replace('d','$');  
 replace = replace.replace('n','№');  
 replace = replace.replace('z','\*');  
 return replace;  
 }  
}

## 5.3. Тестирование работы программы с проверкой

Таблица 5.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 6

## 6.1. Постановка задачи

Дан массив строк, который содержит информацию о музыкальных коллективах (каждая ячейка соответствует 1 музыкальному коллективу), не менее 20 строк. Создать две программы для регламентированного и нерегламентированного ввода данных о музыкальных коллективах. Определить в каждой программе: количество музыкальных коллективов по стилю музыки, в которых не менее 40 % участников имеют музыкальное образование и не более 10 % участников имеют театральное образование, есть не более 5 дипломов музыкальных конкурсов, не менее 3 грамот и хотя бы одна премия; количество музыкальных коллективов по странам по разным стилям, в которых у которых средний доход за год более 5 млн. руб. и есть хотя бы 1 участник младше 21 года.

- Первая программа: Сделать регламентированный ввод данных о музыкальных коллективах. Каждая ячейка массива содержит информацию в определенном виде: Страна коллектива; Город коллектива; Стиль, в котором коллектив выступает (рок, джаз, народный и т.д.); Название коллектива; Наличие солиста; Возраст солиста; Количество участников; Количество участников младше 21 года; Количество участников с музыкальным образованием; Количество участников с театральным образованием; Число музыкальных инструментов в собственности коллектива; Средний доход коллектива от выступлений за год; Наличие премий; Количество премий; Наличие дипломов; Количество дипломов; Наличие грамот; Количество грамот.

- Вторая программа: Сделать нерегламентированный ввод данных о музыкальных коллективах. Т.е. пользователи вводят информацию в произвольном виде.

## 6***.2. Решение задачи, код программы***

import java.util.Scanner;  
  
public class zad\_6 {  
 public void zad\_6() {  
 Scanner scn = new Scanner(System.*in*);  
 String[] stats = {"Страна коллектива","Город коллектива","Стиль","Название коллектива",  
 "Наличие солиста", "Возраст солиста","Количество участников",  
 "Количество участников младше 21 года",  
 "Количество участников с музыкальным образованием",  
 "Количество участников с театральным образованием",  
 "Число музыкальных инструментов в собственности коллектива",  
 "Средний доход коллектива от выступлений за год(млн. руб.)",  
 "Наличие премий","Количество премий","Наличие дипломов",  
 "Количество дипломов","Наличие грамот","Количество грамот"};  
 System.*out*.printf("Введите количество коллективов(не менее 20): ");  
 int count\_teams = scn.nextInt();  
 while (count\_teams < 20){  
 System.*out*.printf("Вы ввели слишком мало коллективов!\n" +  
 "Введите количество коллективов(не менее 20): ");  
 count\_teams = scn.nextInt();  
 }  
 String[] teams = new String[count\_teams];  
 for (int i = 0; i < teams.length; i++) {  
 teams[i] = "";  
 }  
 System.*out*.printf("Какой тип ввода вы веберете?\n" +  
 "1 - Регламентированный\n" +  
 "2 - Нерегламентированный\nОтвет: ");  
 int fl = scn.nextInt();  
 while (fl > 2 || fl < 1) {  
 System.*out*.printf("Недопустимое значение!\n" +  
 "Введите символы \"1\" или \"2\": ");  
 fl = scn.nextInt();  
 }  
 switch (fl) {  
 case 1:  
 *RegInput*(teams,stats);  
 case 2:  
 *NotRegInput*(teams,stats);  
 }  
  
 }  
 public static void Check(String[] teams){  
 String countries = "";  
 int count1 = 0, count2 = 0;  
 for(String team: teams){  
 String[] facts = team.split(";");  
 if ((Integer.*parseInt*(facts[8])\*100/Integer.*parseInt*(facts[6])) >= 40 &&  
 (Integer.*parseInt*(facts[9])\*100/Integer.*parseInt*(facts[6])) <= 10 &&  
 Integer.*parseInt*(facts[15]) <= 5 && Integer.*parseInt*(facts[13]) >= 1 &&  
 Integer.*parseInt*(facts[17]) >= 3){  
 count1++;  
 }  
 if (!countries.matches(".\*"+ facts[0] + ".\*")){  
 countries += facts[0]+";";  
 }  
 }  
 System.*out*.println("Результаты первого условия(среди всех команд): "+count1);  
 for(String country: countries.split(";")){  
 String styles = "";  
 for(String team: teams){  
 String[] facts = team.split(";");  
 if (team.matches(".\*"+country+".\*") && !styles.matches(".\*"+facts[2].toLowerCase()+".\*")){  
 styles += facts[2].toLowerCase()+";";  
 if (Integer.*parseInt*(facts[7]) >= 1 && Integer.*parseInt*(facts[11]) > 5){  
 count2++;  
 }  
 }  
 }  
 }  
 System.*out*.println("Результаты второго условия(по странам): "+count2);  
  
 }  
  
 public static void RegInput(String[] text, String[] stats) {  
 System.*out*.printf("\nВведите данные о командах в массив строк.");  
 Scanner scn = new Scanner(System.*in*);  
 for (int i = 0; i < text.length; i++){  
 System.*out*.printf("\nКоманда №%d:\n",i+1);  
 for(String stat: stats){  
 System.*out*.printf(stat+": ");  
 text[i] += scn.next()+";";  
 }  
 }  
 *Check*(text);  
 }  
 public static void NotRegInput(String[] text, String[] stats) {  
 String[] facts = new String[18];  
 Scanner scn = new Scanner(System.*in*);  
 int count = 0;  
 for(String info: stats){  
 count++;  
 System.*out*.printf("\n%d) %s",count,info);  
 }  
 System.*out*.printf("\n\n");  
 for (int i = 0; i < text.length; i++){  
 for (int j = 0; j < facts.length; j++) {  
 facts[j] = "";  
 }  
 System.*out*.printf("\nКоманда №%d:\n",i+1);  
 for(int j = 0; j < 18;j++){  
 System.*out*.printf("Выберите заполняемые данные: ");  
 int num = scn.nextInt();  
 while(((num > 18) || (num < 1)) || !(facts[num-1].equals(""))) {  
 System.*out*.printf("Введено недопустимое значение! Введите заново: ");  
 num = scn.nextInt();  
 }  
 System.*out*.printf("%s: ", stats[num-1]);  
 facts[num - 1] = scn.next();  
 }  
 for(String fact: facts){  
 text[i] += fact+";";  
 }  
 }  
 *Check*(text);  
 }  
}

## 6.3. Тестирование работы программы с проверкой

Таблица 6.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 7

## 7.1. Постановка задачи

В заданной строке, после каждого слова вида «exp», «ln», «ch» или «sh», поставить «(x-y)».

## 7.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class zad\_7 {  
 public void zad\_7() {  
 Scanner scn = new Scanner(System.*in*);  
 System.*out*.println("Задайте строку: ");  
 String text = scn.nextLine();  
 System.*out*.printf("\nТекущая строка:\n%s\n\nИзмененная строка:\n",text);  
 System.*out*.println(*Arguments*(text));  
 }  
  
 public static String Arguments(String text) {  
 String ln = "ln", exp= "exp", ch = "ch", sh = "sh", arg ="(x-y)";  
 String replace = text.replace(ln,ln+arg);  
 replace = replace.replace(exp,exp+arg);  
 replace = replace.replace(ch,ch+arg);  
 replace = replace.replace(sh,sh+arg);  
 return replace;  
 }  
}

## 7.3. Тестирование работы программы с проверкой

Таблица 5.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 8

## 8.1. Постановка задачи

Дан текст, зашифровать слова заменив m и ь на 0, j и о на 1, k и л на 2, l и д на 3, u и г на 4, i и ш на 5, o (латинская буква) и щ на 6. Подсчитать сколько слов было зашифровано.

## 8.2. Решение задачи, код программы

public class zad\_8 {  
 public void zad\_8() {  
 String text = "В стеклянной тумбочке под телевизором — видеомагнитофон Toshiba " +  
 "модели super-high-band Beta со встроенной функцией редактирования, " +  
 "включая буквенный генератор с восьмистраничной памятью, high-band-запись," +  
 "обратное воспроизведение и трехнедельное предпрограммирование на восемь " +  
 "записей. По углам стоят галогеновые лампы. Тонкие белые жалюзи закрывают " +  
 "все восемь окон — от пола до потолка. Перед диваном — журнальный столик " +  
 "от Turchin (стеклянный верх и дубовые ножки), на котором расставлены " +  
 "стеклянные зверюшки от Steuben и дорогие хрустальные пепельницы от " +  
 "Fortunoff, хоть я и не курю.";  
 String OLDtext = text;  
 text = *Replace\_Chars*(text);  
 if (!text.equals(OLDtext)){  
 System.*out*.println("Текст зашифрован!");  
 System.*out*.printf("В тексте произошло %d замен", *Count*(OLDtext));  
 }else{  
 System.*out*.println("В тексте нет подходящих замен");  
 }  
 System.*out*.println(text);  
 }  
 public static int Count(String text){  
 String[] words = text.split("\\s+");  
 int count = 0;  
 for (int i = 0; i < words.length; i++){  
 if (!words[i].equals(*Replace\_Chars*(words[i]))){  
 count++;  
 }  
 }  
 return count;  
 }  
  
 public static String Replace\_Chars(String text) {  
 String replace = text.replaceAll("[mь]","0");  
 replace = replace.replaceAll("[jо]","1");  
 replace = replace.replaceAll("[lд]","3");  
 replace = replace.replaceAll("[uг]","4");  
 return replace;  
 }  
}

## 8.3. Тестирование работы программы с проверкой

Таблица 8.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 9

## 9.1. Постановка задачи

Дан текст, который содержит комментарии. Из текста удалить все комментарии (\_ … \_, - … -,   
= … =), если комментарий нет, то сообщить об этом.

## 9.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class zad\_9 {  
 public void zad\_9() {  
 Scanner scn = new Scanner(System.*in*);  
 System.*out*.println("Введите текст: ");  
 String text = scn.nextLine();  
 String replace = *Replace\_Coms*(text);  
 if (replace.equals(text)) {  
 System.*out*.println("Комментарий в тексте нет");  
 } else {  
 System.*out*.printf("\nКомментарии удалены!\n\nИзмененный текст:\n%s", replace);  
 }  
 }  
  
 public static String Replace\_Coms(String text) {  
 String replace = text.replaceAll("-.+-", "");  
 replace = replace.replaceAll("\_.+\_", "");  
 replace = replace.replaceAll("=.+=", "");  
 return replace;  
 }  
}

## 9.3. Тестирование работы программы с проверкой

Таблица 9.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 10

## 10.1. Постановка задачи

Дан текст, который содержит данные определенной структуры, разделителем данных об одном объекте является «;», разделитель между объектами «.». «Футболист»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номер телефона; название команды; номер в команде; амплуа; результативность (количество голов); количество игр. Разбить данные по объектам на массив строк. Посчитать количество людей с номерами телефона, начинающимися на +38 и заканчивающимися на цифру, соответствующую возрасту футболиста. Вывести сведения про футболистов, которые провели за свою команду больше 50 матчей с результативностью более 20 голов.

## 10.2. Решение задачи, код программы

import java.text.SimpleDateFormat;  
import java.util.Date;  
  
public class zad\_10 {  
 public void zad\_10() {  
 String text = "Иванов;Иван;Иванович;М;Украинец;180;75;1986,3,12;+380971234567;Динамо;9;нападающий;30;10." +  
 "Петров;Петр;Петрович;М;Русский;175;70;1987,5,23;+380931234565;Шахтер;5;защитник;5;70." +  
 "Сидоров;Сергей;Александрович;М;Белорус;190;90;1988,8,5;+375291234564;Динамо;2;вратарь;1000;2000." +  
 "Коваленко;Ольга;Николаевна;Ж;Украинка;170;65;1990,1,20;+380991234563;Спартак;8;полузащитник;150;70." +  
 "Шевченко;Андрей;Михайлович;М;Украинец;185;80;1976,9,29;+380631234566;Милан;7;нападающий;120;300.";  
 String[] persons = text.split("\\.");  
 int countNum = 0;  
 Date date = new Date();  
 SimpleDateFormat DateFormat = new SimpleDateFormat("yyyy/MM/dd/");  
 String[] NowDate = DateFormat.format(date).split("/");  
 System.*out*.println("Спортсмены с более 50 игр и более 20 голов: ");  
 for (String persInfo : persons) {  
 String[] info = persInfo.split(";");  
 String[] BF = info[7].split(",");  
 int year;  
 if ((Integer.*parseInt*(NowDate[1]) >= Integer.*parseInt*(BF[1])) &&  
 (Integer.*parseInt*(NowDate[2]) >= Integer.*parseInt*(BF[2]))) {  
 year = Integer.*parseInt*(NowDate[0]) - Integer.*parseInt*(BF[0]);  
 } else {  
 year = Integer.*parseInt*(NowDate[0]) - Integer.*parseInt*(BF[0]) - 1;  
 }  
 if (info[8].matches("\\+38\\d+" + String.*valueOf*(year % 10))) {  
 countNum++;  
 }  
 if (Integer.*parseInt*(info[13]) > 50 && Integer.*parseInt*(info[12]) > 20) {  
 System.*out*.println(persInfo);  
 }  
 }  
 System.*out*.printf("\nВсего %d спортсменов имет номер начинающийся с +38 и заканчивающиеся последней цифрой возраста", countNum);  
 }  
}

## 10.3. Тестирование работы программы с проверкой

Таблица 10.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 11

## 11.1. Постановка задачи

Создать программу форматирования массива строк используя три разных регулярных выражения, т.е. регулярные выражения разного типа, созданные при использовании разных метасимволов и их сочетаний.

## 11.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class zad\_11 {  
 public void zad\_11() {  
 MyMass arr = new MyMass();  
 String[] text = arr.GetStrMas();  
 String[] NEWtext = *Format*(text);  
 arr.PntStrMas(NEWtext);  
 }  
  
 public static String[] Format(String[] text) {  
 MyMass arr = new MyMass();  
 Scanner scn = new Scanner(System.*in*);  
 String[] txt = text;  
 System.*out*.printf("\nЗадайте тип форматирования.\n" +  
 "1 - удаление всех букв и пробелов из строки\n" +  
 "2 - удаление всех символов кроме букв, цифр и нижнего подчеркивания\n" +  
 "3 - добавить тройное восклицание\nОтвет: ");  
 int fl = scn.nextInt();  
 while (fl > 3 || fl < 1) {  
 System.*out*.printf("Недопустимое значение!\n" +  
 "Введите символы \"1\",\"2\" или \"3\": ");  
 fl = scn.nextInt();  
 }  
 switch (fl) {  
 case 1:  
 for (int i = 0; i < txt.length; i++) {  
 txt[i] = txt[i].replaceAll("[A**-**Za**-**zА**-**Яа**-**я ]+", "");  
 }  
 break;  
 case 2:  
 for (int i = 0; i < txt.length; i++) {  
 txt[i] = txt[i].replaceAll("[^\\wА**-**Яа**-**я]", "");  
 }  
 break;  
 case 3:  
 for (int i = 0; i < txt.length; i++) {  
 txt[i] = txt[i].replaceAll("!{1,2}", "!!!");  
 }  
 break;  
 }  
 return txt;  
 }  
}

## 11.3. Тестирование работы программы с проверкой

Таблица 11.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 12

## 12.1. Постановка задачи

Внук написал бабушке послание, но забыл заменить слова современного молодежного сленга на нормальные, а так же допустил ряд ошибок. Текст оформлен в виде массива строк. Помогите бабушке понять внука. Она выделила то, что нужно исправить: 1. Повторение слов через один или несколько пробелов; 2. Есть слова паразиты «будем говорить», «ну», «туто», «короче», «типа»; 3. Молодежный сленг «катка», «кекать», «краш», «дноклы», «лол», «нуб», «падра», «ОМГ», «стримить», «флексить», «чилить». В цикле с использование регулярных выражений выполнить автозамену информации с использованием массивов регулярных выражений. Исходную и измененную информацию вывести построчно: сначала исходная строка, затем измененная.

## 12.2. Решение задачи, код программы

public class zad\_12 {  
 public void zad\_12() {  
 MyMass arr = new MyMass();  
 System.*out*.println("Задайте массив сообщений для бабушки.");  
 String[] messages = arr.GetStrMas();  
 *replaces*(messages);  
 }  
  
 public static void replaces(String[] messages) {  
 String[] patterns = {"[бБ]удем говорить", "[нН]у", "[тТ]уто", "[Кк]ороче", "[Тт]ипа"};  
 String[] sleng = {"[кК]ат[ак][а**-**я]+", "[кК]ек[а**-**я]+", "[кК]раш[а**-**я]\*", "[лЛ]ол[а**-**я]\*",  
 "[нН]уб[а**-**я]\*", "ОМГ", "[Сс]трим[а**-**я]\*", "[фФ]лекс[а**-**я]\*",  
 "[чЧ]ил[а**-**я]\*", "[пП]адр[а**-**я]+", "[дД]нокл[а**-**я]\*"};  
 String[] trans = {"(играть)", "(смеяться)", "(пример подражания)", "(очень смешно)",  
 "(новичек)", "(крайнее удивление)", "(трансляция)", "(быть увлеченным)",  
 "(отдых)", "(подруга)", "(одноклассник)"};  
 for (String message : messages) {  
 System.*out*.printf("\nИсходный текст: %s", message);  
 message = message.replaceAll("\\s{2,}", " ");  
 //String[] words = message.split("\\s+");  
 for (String pattern : patterns) {  
 message = message.replaceAll(pattern + "[.,!?]?\\s\*", "");  
 }  
 String[] words = message.split("\\s+");  
 for (String word : words) {  
 String clearWord = word.replaceAll("[^А**-**Яа**-**я]", "");  
 message = message.replaceAll(clearWord + ",?\\s+,?" + clearWord, clearWord);  
 message = message.replaceAll(clearWord.substring(0, 1).toUpperCase()  
 + clearWord.substring(1) + ",?\\s+,?" + clearWord, clearWord);  
 message = message.replaceAll(clearWord + ",?\\s+,?" + clearWord.substring(0, 1).toUpperCase()  
 + clearWord.substring(1), clearWord.substring(0, 1).toUpperCase()  
 + clearWord.substring(1));  
 }  
 words = message.split("\\s+");  
 String NEWmessage = "";  
 for (int i = 0; i < words.length; i++) {  
 String word = words[i];  
 if (word.equals(words[0])) {  
 word = word.substring(0, 1).toUpperCase() + word.substring(1);  
 } else if (words[i - 1].charAt(words[i - 1].length() - 1) == '.' ||  
 words[i - 1].charAt(words[i - 1].length() - 1) == '?' ||  
 words[i - 1].charAt(words[i - 1].length() - 1) == '!') {  
 word = word.substring(0, 1).toUpperCase() + word.substring(1);  
 }  
 String clearWord = word.replaceAll("[^А**-**Яа**-**я]", "");  
 for (int k = 0; k < trans.length; k++) {  
 word = word.replaceAll(sleng[k], clearWord + trans[k]);  
 }  
 NEWmessage += word + " ";  
 }  
 System.*out*.printf("\nИсправленный текст: %s\n", NEWmessage);  
 }  
 }  
}

## 12.3. Тестирование работы программы с проверкой

Таблица 12.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |