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 «Линейные алгоритмы в Java»

по дисциплине «Информатика»

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

Проверил: Нетбай Г.В.

Пермь, 2023

**Содержание**

[Задание 1 4](#_Toc136778439)

[1.1. Постановка задачи 4](#_Toc136778440)

[1.2. Решение задачи, код программы 4](#_Toc136778441)

[1.3. Тестирование работы программы с проверкой 4](#_Toc136778442)

[Задание 2 5](#_Toc136778443)

[2.1. Постановка задачи 5](#_Toc136778444)

[2.2. Решение задачи, код программы 5](#_Toc136778445)

[2.3. Тестирование работы программы с проверкой 5](#_Toc136778446)

[Задание 3 6](#_Toc136778447)

[3.1. Постановка задачи 6](#_Toc136778448)

[3.2. Решение задачи, код программы 6](#_Toc136778449)

[3.3. Тестирование работы программы с проверкой 7](#_Toc136778450)

[Задание 4 8](#_Toc136778451)

[4.1. Постановка задачи 8](#_Toc136778452)

[4.2. Решение задачи, код программы 8](#_Toc136778453)

[4.3. Тестирование работы программы 9](#_Toc136778454)

[Задание 5 10](#_Toc136778455)

[5.1. Постановка задачи 10](#_Toc136778456)

[5.2. Решение задачи, код программы 10](#_Toc136778457)

[5.3. Тестирование работы программы с проверкой 10](#_Toc136778458)

[Задание 6.2. 11](#_Toc136778459)

[6.1. Постановка задачи 11](#_Toc136778460)

[6.2. Решение задачи, код программы 11](#_Toc136778461)

[6.3. Тестирование работы программы с проверкой 12](#_Toc136778462)

[Задание 7 13](#_Toc136778463)

[7.1. Постановка задачи 13](#_Toc136778464)

[7.2. Решение задачи, код программы 13](#_Toc136778465)

[7.3. Тестирование работы программы с проверкой 13](#_Toc136778466)

[Задание 8 14](#_Toc136778467)

[8.1. Постановка задачи 14](#_Toc136778468)

[8.2. Решение задачи, код программы 14](#_Toc136778469)

[8.3. Тестирование работы программы с проверкой 14](#_Toc136778470)

[Задание 9 15](#_Toc136778471)

[9.1. Постановка задачи 15](#_Toc136778472)

[9.2. Решение задачи, код программы 15](#_Toc136778473)

[9.3. Тестирование работы программы с проверкой 15](#_Toc136778474)

[Задание 10 16](#_Toc136778475)

[10.1. Постановка задачи 16](#_Toc136778476)

[10.2. Решение задачи, код программы 16](#_Toc136778477)

[10.3. Тестирование работы программы с проверкой 17](#_Toc136778478)

[Задание 11 18](#_Toc136778479)

[11.1. Постановка задачи 18](#_Toc136778480)

[11.2. Решение задачи, код программы 18](#_Toc136778481)

[11.3. Тестирование работы программы с проверкой 18](#_Toc136778482)

[Задание 12 19](#_Toc136778483)

[12.1. Постановка задачи 19](#_Toc136778484)

[12.2. Решение задачи, код программы 19](#_Toc136778485)

[12.3. Тестирование работы программы с проверкой 20](#_Toc136778486)

# Задание 1

## 1.1. Постановка задачи

Введите массив символов из 12 элементов. Замените каждый символ- цифру на символ «!».

## 1.2. Решение задачи, код программы

public class Lab9Tsk1 {  
 public static void main(String[] args) {  
 char[] arr = {'a', '1', 'b', '2', 'c', '3', 'd', '4', 'e', '5', 'f', '6'};  
 for (int i = 0; i < arr.length; i++) {  
 if (Character.*isDigit*(arr[i])) {  
 arr[i] = '!';  
 }  
 }  
 System.*out*.println(arr);  
 }  
 }

## 1.3. Тестирование работы программы с проверкой

Таблица 1

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение MS Excel |
| 1 |  | |

# Задание 2

## 2.1. Постановка задачи

Дан массив символов, содержащий число в двоичной системе счисления. Проверить правильность ввода этого числа (в его записи должны быть только символы 0 и 1). 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 8-ую систему счисления, записать число в новый массив символов.

## 2.2. Решение задачи, код программы

import java.util.Scanner;  
public class Lab9Tsk2 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 String binaryNumber;  
 boolean isBinary = false;  
 do {  
 System.*out*.print("Введите число в двоичной системе счисления: ");  
 binaryNumber = in.nextLine();  
 isBinary = true;  
 for (int i = 0; i < binaryNumber.length(); i++) {  
 if (binaryNumber.charAt(i) != '0' && binaryNumber.charAt(i) != '1') {  
 isBinary = false;  
 break;  
 }  
 }  
 if (!isBinary) {  
 System.*out*.println("Ошибка ввода! Введите число, состоящее только из символов 0 и 1.");  
 }  
 } while (!isBinary);  
 int decimalNumber = Integer.*parseInt*(binaryNumber, 2);  
 String octalNumber = Integer.*toOctalString*(decimalNumber);  
 char[] octalArray = octalNumber.toCharArray();  
 System.*out*.print("Число в 8-ой системе счисления: ");  
 System.*out*.println(octalArray);  
 }  
}

## 2.3. Тестирование работы программы с проверкой

Тестирование работы программы и проверка результатов решения

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

# Задание 3

## 3.1. Постановка задачи

Дан массив символов. Проанализировать массив и ответить на вопросы:

а) Определить количество прописных букв русского языка.

б) Выяснить, входит ли в последовательность знак препинания, введенный с клавиатуры, например «:».

в) Выяснить, верно ли, что среди символов имеются хотя бы две буквы, входящие в слово, введенное с клавиатуры, например шина.

г) Выяснить, имеется ли среди символов пара соседствующих букв «то» или «от».

д) Выяснить, имеется ли среди символов три подряд идущих одинаковых символа.

е) Выяснить, верно ли. что существуют такие натуральные i и j, что n/2 <i <j < п и что si, si+1 цифры и равны по модулю, a sj, sj+1 это одинаковые буквы.

## 3.2. Решение задачи, код программы

import java.util.Scanner;  
public class Lab9Tsk3 {  
 public static void main(String[] args) {  
 System.*out*.print("Введите строку : ");  
 Scanner in = new Scanner(System.*in*);  
 String str = in.nextLine();  
 char [] arr = str.toCharArray();  
 int cnt1 = 0;  
 for (char c : arr) {  
 if (Character.*isUpperCase*(c) && Character.UnicodeBlock.*of*(c) == Character.UnicodeBlock.*CYRILLIC*) {  
 cnt1++;  
 }  
 }  
 System.*out*.println("Количество прописных букв русского языка: " + cnt1);  
 System.*out*.print("Введите знак препинания: ");  
 char a = in.next().charAt(0);  
 boolean flag = false;  
 for (char c : arr) {  
 if (c == a) {  
 flag = true;  
 break;  
 }  
 }  
 System.*out*.println("Знак препинания " + a + " присутствует: " + flag);  
 System.*out*.print("Введите слово: ");  
 String word = in.next();  
 int cnt2 = 0;  
 for (char c : arr) {  
 if (word.indexOf(c) != -1) {  
 cnt2++;  
 }  
 if (cnt2 >= 2) {  
 break;  
 }  
 }  
 System.*out*.println("Среди символов имеются хотя бы две буквы из слова: " + (cnt2 >= 2));  
 boolean flag2 = false;  
 for (int i = 0; i < arr.length - 1; i++) {  
 if ((arr[i] == 'т' && arr[i + 1] == 'о') || (arr[i] == 'о' && arr[i + 1] == 'т')) {  
 flag2 = true;  
 break;  
 }  
 }  
 System.*out*.println("Имеется ли среди символов пара соседствующих букв «то» или «от»: " + flag2);  
 boolean flag3 = false;  
 for (int i = 0; i < arr.length - 2; i++) {  
 if (arr[i] == arr[i + 1] && arr[i] == arr[i + 2]) {  
 flag3 = true;  
 break;  
 }  
 }  
 System.*out*.println("Имеется ли среди символов три подряд идущих одинаковых символа: " + flag3);  
 boolean flagE = false;  
 for (int i = arr.length / 2 + 1; i < arr.length - 1; i++) {  
 for (int j = i + 1; j < arr.length - 1; j++) {  
 if (Character.*isDigit*(arr[i]) && Character.*isDigit*(arr[i + 1]) && Math.*abs*(arr[i] - arr[i + 1]) == 1 &&  
 Character.*isLetter*(arr[j]) && Character.*isLetter*(arr[j + 1]) && arr[j] == arr[j + 1]) {  
 flagE = true;  
 break;  
 }  
 }  
 if (flagE) {  
 break;  
 }  
 }  
 System.*out*.println("Верно ли условие е: " + flagE);  
 }  
}

## 3.3. Тестирование работы программы с проверкой

Таблица 3

Тестирование работы программы и проверка результатов решения

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

# 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») = «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.Arrays;  
public class Lab9Tsk4u {  
 public static void main(String[] args) {  
 char[][] array = {  
 {'1', '+', '2', '/', '2', '=', '2'},  
 {'3', '-', '4', '/', '2', '=', '1'},  
 {'5', '+', '6', '/', '3', '=', '7'},  
 {'8', '-', '9', '/', '3', '=', '5'},  
 {'1', '+', '1', '/', '1', '=', '3'}  
 };  
 int cnt = 0;  
 ArrayList<String> incorrectExpressions = new ArrayList<>();  
 for (char[] row : array) {  
 String exp = new String(row);  
 if (*RegexCheck*(exp)) {  
 if (*MathCheck*(exp)) {  
 cnt++;  
 } else {  
 incorrectExpressions.add(*Fix*(exp));  
 }  
 }  
 }  
 for (int i = 0; i < array[0].length; i++) {  
 StringBuilder expressionBuilder = new StringBuilder();  
 for (char[] row : array) {  
 expressionBuilder.append(row[i]);  
 }  
 String exp = expressionBuilder.toString();  
 if (*RegexCheck*(exp)) {  
 if (*MathCheck*(exp)) {  
 cnt++;  
 } else {  
 incorrectExpressions.add(*Fix*(exp));  
 }  
 }  
 }  
 System.*out*.println("Количество правильно вычисленных выражений: " + cnt);  
 System.*out*.println("Неправильно вычисленные выражения с корректировкой ответа: " + Arrays.*toString*(incorrectExpressions.toArray()));  
 }  
  
 private static boolean RegexCheck(String exp) {  
 return exp.matches("\\d[+\\-]\\d/\\d=\\d");  
 }  
  
 private static boolean MathCheck(String exp) {  
 int num1 = Character.*getNumericValue*(exp.charAt(0));  
 char sing = exp.charAt(1);  
 int num2 = Character.*getNumericValue*(exp.charAt(2));  
 int num3 = Character.*getNumericValue*(exp.charAt(4));  
 int result = Character.*getNumericValue*(exp.charAt(6));  
 int resultC;  
 if (sing == '+') {  
 resultC = num1 + num2 / num3;  
 } else {  
 resultC = num1 - num2 / num3;  
 }  
 return resultC == result;  
 }  
  
 private static String Fix(String exp) {  
 int num1 = Character.*getNumericValue*(exp.charAt(0));  
 char sing = exp.charAt(1);  
 int num2 = Character.*getNumericValue*(exp.charAt(2));  
 int num3 = Character.*getNumericValue*(exp.charAt(4));  
 int resultC;  
 if (sing == '+') {  
 resultC = num1 + num2 / num3;  
 } else {  
 resultC = num1 - num2 / num3;  
 }  
 return exp.substring(0, 6) + resultC;  
 }  
}

## 4.3. Тестирование работы программы

Таблица 4

Тестирование работы программы и проверка результатов решения

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

# Задание 5

## 5.1. Постановка задачи

Определить, сколько раз повторяется в тексте каждое слово, которое встречается в нем.

## 5.2. Решение задачи, код программы

import java.util.HashMap;  
import java.util.Scanner;  
public class Lab9Tsk5u {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.print("Введите текст: ");  
 String str = in.nextLine();  
 str = str.replaceAll("[^a-zA-Zа-яА-Я ]", "").toLowerCase();  
 String[] words = str.split("\\s+");  
 HashMap<String, Integer> wordsCnt = new HashMap<>();  
 for (String word : words) {  
 if (wordsCnt.containsKey(word)) {  
 wordsCnt.put(word, wordsCnt.get(word) + 1);  
 } else {  
 wordsCnt.put(word, 1);  
 }  
 }  
 for (String word : wordsCnt.keySet()) {  
 System.*out*.println(word + ": " + wordsCnt.get(word));  
 }  
 }  
}

## 5.3. Тестирование работы программы с проверкой

Таблица 5

Тестирование работы программы и проверка результатов решения

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

# Задание 6.2.

## 6.1. Постановка задачи

Вторая программа: Сделать нерегламентированный ввод данных по проектам. Т.е. пользователи вводят информацию в произвольном виде.

## 6.2. Решение задачи, код программы

import java.util.\*;  
public class Lab9Tsk62 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.print("input n ");  
 int n = in.nextInt();  
 in.nextLine();  
 String[][] arr = new String[n][11];  
 for(int j = 0; j < n; j++){  
 for (int i = 0; i < 11; i++) {  
 System.*out*.print("Введите строку " + (i + 1) + ": ");  
 arr[j][i] = in.nextLine();  
 }  
 System.*out*.println("Вы ввели " + j + 1 + " массив информации");  
 }  
 int multiCntF = 0;  
 Map<String, Integer> fClassCnt = new HashMap<>();  
 Map<String, Integer> bClassCnt = new HashMap<>();  
 for (String [] array : arr) {  
 String[] result = *findInfoByKeywords*(array, "Адрес", "Корпус", "Номер аудитории", "Тип аудитории", "Количество посадочных мест", "Наличие мультимедийного оборудования",  
 "Наличие компьютеров", "Количество компьютеров", "Принадлежность факультету", "Принадлежность кафедре", "Ответственное лицо");  
 Set<String> rPersom = new HashSet<>();  
 String f = result[8];  
 String b = result[1];  
 String rPerson = result[10];  
 boolean hasMulti = Boolean.*parseBoolean*(result[5]);  
 boolean hasComp = Boolean.*parseBoolean*(result[6]);  
 int compCnt = Integer.*parseInt*(result[7]);  
 int seatCnt = Integer.*parseInt*(result[4]);  
 if (hasMulti && hasComp && compCnt >= 15 && seatCnt >= 30) {  
 multiCntF++;  
 rPersom.add(rPerson);  
 fClassCnt.put(f, fClassCnt.getOrDefault(f, 0) + 1);  
 bClassCnt.put(b, bClassCnt.getOrDefault(b, 0) + 1);  
 }  
 }  
 System.*out*.println("Количество компьютерных классов по факультетам с мультимедиа: " + multiCntF);  
 System.*out*.println(fClassCnt);  
 System.*out*.println(bClassCnt);  
 }  
  
 public static String[] findInfoByKeywords(String[] arr, String... keywords) {  
 List<String> orderedResult = new ArrayList<>();  
 for (String keyword : keywords) {  
 for (String s : arr) {  
 if (s.contains(keyword)) {  
 int index = s.indexOf(keyword) + 1;  
 orderedResult.add(s.substring(index + keyword.length()));  
 }  
 }  
 }  
 return String.*join*("\n", orderedResult).split("\n");  
 }  
}

## 6.3. Тестирование работы программы с проверкой

Таблица 6

Тестирование работы программы и проверка результатов решения

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

# Задание 7

## 7.1. Постановка задачи

В заданной строке посчитать количество разных символов, входящих в эту строку. Создать новую строку, состоящую из букв первой строки и количества их вхождений в нее.

## 7.2. Решение задачи, код программы

import java.util.HashMap;  
import java.util.Scanner;  
public class Lab9Tsk7 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.print("Введите строку: ");  
 String str = in.nextLine();  
 HashMap<Character, Integer> charCnt = new HashMap<>();  
 for (int i = 0; i < str.length(); i++) {  
 char c = str.charAt(i);  
 if (charCnt.containsKey(c)) {  
 charCnt.put(c, charCnt.get(c) + 1);  
 } else {  
 charCnt.put(c, 1);  
 }  
 }  
 StringBuilder stringBuilder = new StringBuilder();  
 for (char c : charCnt.keySet()) {  
 stringBuilder.append(c);  
 stringBuilder.append(charCnt.get(c));  
 }  
 System.*out*.println("Количество разных символов: " + charCnt.size());  
 System.*out*.println("Новая строка: " + stringBuilder);  
 }  
}

Таблица 7

## 7.3. Тестирование работы программы с проверкой

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

# Задание 8

## 8.1. Постановка задачи

Распаковать текст, сжатый по правилу: каждая группа из двух или более рядом стоящих символов заменяется на один символ, за которым следует количество его вхождений в группу. К примеру, строка hellowoooorld должна сжиматься в hel2owo4rld.

## 8.2. Решение задачи, код программы

public class Lab9Tsk8 {  
 public static void main(String[] args) {  
 String str = "hellowoooorld";  
 String compressed = *compress*(str);  
 System.*out*.println(compressed);  
 }  
 public static String compress(String str) {  
 StringBuilder stringBuilder = new StringBuilder();  
 int cnt = 1;  
 for (int i = 0; i < str.length(); i++) {  
 if (i == str.length() - 1 || str.charAt(i) != str.charAt(i + 1)) {  
 stringBuilder.append(str.charAt(i));  
 if (cnt > 1) {  
 stringBuilder.append(cnt);  
 }  
 cnt = 1;  
 } else {  
 cnt++;  
 }  
 }  
 return stringBuilder.toString();  
 }  
}

Таблица 8

## 8.3. Тестирование работы программы с проверкой

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

# 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.regex.Matcher;  
import java.util.regex.Pattern;  
public class Lab9Tsk9 {  
 public static void main(String[] args) {  
 String text = "txt // one string coment ; /\* multi string coment \*/ /\*\* documentary coment \*/";  
 if (!text.contains("/")) {  
 System.*out*.println("text doesn't contains any coments");  
 System.*exit*(0);  
 }  
 text = text.replaceAll("//.\*?;", "");  
 Pattern pattern = Pattern.*compile*("/\\\*.\*?\\\*/", Pattern.*DOTALL*);  
 Matcher matcher = pattern.matcher(text);  
 text = matcher.replaceAll("");  
 text = text.replaceAll("/\\\*\\\*.\*?\\\*/", "");  
 if (text.trim().isEmpty()) {  
 System.*out*.println("no any symbols besides coments");  
 } else {  
 System.*out*.println(text);  
 }  
 }  
}

## 9.3. Тестирование работы программы с проверкой

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 | String text = "txt // one string coment ; /\* multi string coment \*/ /\*\* documentary coment \*/"; |

# Задание 10

## 10.1. Постановка задачи

Дан текст, который содержит данные определенной структуры, разделителем данных об одном объекте является «;», разделитель между объектами «.». «Баскетболист»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номер телефона; название команды; номер в команде; амплуа; результативность (количество очков); количество игр. Разбить данные по объектам на массив строк. Посчитать количество людей с ростом от 200 до 220 см, при этом вес которых от 90 до 110 кг. Вывести сведения про баскетболистов, которых забросили за свою команду больше 150 очков.

## 10.2. Решение задачи, код программы

import java.util.ArrayList;  
public class Lab9Tsk10 {  
 public static void main(String[] args ){  
 String str = "Баскетболист: Иванов; Иван; Иванович; М; Россиянин; 205; 100; 1995, 10, 15; +7(999)123-45-67; Команда1; 5; Центровой; 200; 20. "  
 +"Баскетболист: Петров; Петр; Петрович; М; Россиянин; 190; 80; 1996, 5, 20; +7(999)234-56-78; Команда2; 10; Форвард; 150; 30. "  
 +"Баскетболист: Сидорова; Мария; Ивановна; Ж; Россиянка; 210; 110; 1994, 3, 1; +(999)345-67-89; Команда1; 7; Центровой; 250; 25. "  
 +"Баскетболист: Смирнов; Алексей; Петрович; М; Россиянин; 195; 95; 1993, 12, 5; +7(999)456-78-90; Команда2; 8; Форвард; 100; 40.\n";  
 String[] parts = str.trim().split("\\.");  
 int cnt1 = 0;  
 ArrayList<String> highScore = new ArrayList<>();  
 for (String data : parts) {  
 String[] stats = data.split(";");  
 int height = Integer.*parseInt*(stats[5].trim());  
 int weight = Integer.*parseInt*(stats[6].trim());  
 int points = Integer.*parseInt*(stats[12].trim());  
 if (height >= 200 && height <= 220 && weight >= 90 && weight <= 110) {  
 cnt1++;  
 }  
 if (points > 150) {  
 highScore.add(data);  
 }  
 }  
 System.*out*.println("Количество людей с ростом от 200 до 220 см и весом от 90 до 110 кг: " + cnt1);  
 System.*out*.println("Сведения про баскетболистов, которых забросили за свою команду больше 150 очков:");  
 for (String highScorer : highScore) {  
 System.*out*.println(highScorer);  
 }  
 }  
  
}

Таблица 10

## 10.3. Тестирование работы программы с проверкой

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

# 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.Arrays;  
public class Lab9Tsk11 {  
 public static void main(String[] args) {  
 String[] strings = {" Every ", " elk ", " go to forest "};  
 for (int i = 0; i < strings.length; i++) {  
 strings[i] = strings[i].trim();  
 }  
 for (int i = 0; i < strings.length; i++) {  
 strings[i] = strings[i].replaceAll("**\s**+", " ");  
 }  
 for (int i = 0; i < strings.length; i++) {  
 strings[i] = strings[i].substring(0, 1).toUpperCase() + strings[i].substring(1);  
 }  
 System.*out*.println(Arrays.*toString*(strings));  
 }  
}

Таблица 11

## 11.3. Тестирование работы программы с проверкой

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

# Задание 12

## 12.1. Постановка задачи

Дан массив строк по месяцам о численности и успеваемости учащихся группы, не менее 24 строк. В цикле с использование регулярных выражений выполнить автозамену информации, так как текст записан без учета склонений, которые автозамена и должна исправить. Исходную и измененную информацию вывести построчно: сначала исходная строка, затем измененная.

## 12.2. Решение задачи, код программы

import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
public class Lab9Tsk12 {  
 public static void main(String[] args) {  
 String[] array = {  
 "Январь: 20 учащихся, 15 отличников",  
 "Февраль: 22 учащихся, 12личников",  
 "Март: 25 учащихся, 18 отличников",  
 "Апрель: 24 учащихся, 14 отличников",  
 "Май: 26 учаихся, 20 отличников",  
 "Июнь: 23 учащихся, 16 отличников",  
 "Июль: 25 учащихся, 19 отличников",  
 "Август: 24 учащихся, 17личников",  
 "Сентябрь: 26 учащихся, 21 отличник",  
 "Октябрь: 23 учащихся, 15 отличников",  
 "Ноябрь: 25 учащся, 18 отличников",  
 "Декабрь: 24 учащихся, 16 отличников"  
 };  
 Pattern pattern = Pattern.*compile*("(\\d+)\\s+учащихся,\\s+(\\d+)\\s+отличник");  
 for (String month : array) {  
 System.*out*.println("Исходная строка: " + month);  
 Matcher matcher = pattern.matcher(month);  
 if (matcher.find()) {  
 int students = Integer.*parseInt*(matcher.group(1));  
 int excellent = Integer.*parseInt*(matcher.group(2));  
 String newMonth = month.replaceAll("(\\d+)\\s+учащихся", *getDeclension*(students, "учащийся", "учащихся", "учащих"))  
 .replaceAll("(\\d+)\\s+отличник", *getDeclension*(excellent, "отличник", "отличника", "отличников"));  
  
 System.*out*.println("Измененная строка: " + newMonth);  
 } else {  
 System.*out*.println("Ошибка: не найдены данные о численности учащихся и отличников");  
 }  
 }  
 }  
 private static String getDeclension(int number, String one, String two, String five) {  
 int mod10 = number % 10;  
 int mod100 = number % 100;  
 if (mod10 == 1 && mod100 != 11) {  
 return number + " " + one;  
 } else if (mod10 >= 2 && mod10 <= 4 && (mod100 < 10 || mod100 >= 20)) {  
 return number + " " + two;  
 } else {  
 return number + " " + five;  
 }  
 }  
}

Таблица 12

## 12.3. Тестирование работы программы с проверкой

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