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»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-21-1б Евсеев В.С.

Проверил: доцент каф. ВММБ Каменских А.А.

Пермь, 2022

Содержание

[Задание 1 4](#_Toc102336967)

[1.1. Общая постановка задачи 4](#_Toc102336968)

[1.2. Решение задачи, код программы 4](#_Toc102336969)

[1.3. Тестирование программы с проверкой 4](#_Toc102336970)

[Задание 2 6](#_Toc102336971)

[2.1. Общая постановка задачи 6](#_Toc102336972)

[2.2. Решение задачи, код программы 6](#_Toc102336973)

[2.3. Тестирование программы с проверкой 9](#_Toc102336974)

[3 Задание 11](#_Toc102336975)

[3.1. Общая постановка задачи 11](#_Toc102336976)

[3.2. Решение задачи, код программы 12](#_Toc102336977)

[3.3. Тестирование программы с проверкой 14](#_Toc102336978)

[Задание 4 15](#_Toc102336979)

[4.1. Общая постановка задачи 15](#_Toc102336980)

[4.2. Решение задачи, код программы 15](#_Toc102336981)

[4.3. Тестирование программы с проверкой 18](#_Toc102336982)

[Задание 5 20](#_Toc102336983)

[5.1. Общая постановка задачи 20](#_Toc102336984)

[5.2. Решение задачи, код программы 20](#_Toc102336985)

[5.3. Тестирование программы с проверкой 20](#_Toc102336986)

[Задание 6 22](#_Toc102336987)

[6.1. Общая постановка задачи 22](#_Toc102336988)

[6.2. Решение задачи, код программы 23](#_Toc102336989)

[6.2.1. Решение задачи, код первой программы 23](#_Toc102336990)

[6.2.2. Решение задачи, код второй программы 26](#_Toc102336991)

[6.3. Тестирование программы с проверкой 28](#_Toc102336992)

[6.3.1. Тестирование первой программы с проверкой 28](#_Toc102336993)

[6.3.2. Тестирование второй программы с проверкой 30](#_Toc102336994)

[7 Задание 32](#_Toc102336995)

[7.1. Общая постановка задачи 32](#_Toc102336996)

[7.2. Решение задачи, код программы 32](#_Toc102336997)

[7.3. Тестирование программы с проверкой 33](#_Toc102336998)

[Задание 8 34](#_Toc102336999)

[8.1. Общая постановка задачи 34](#_Toc102337000)

[8.2. Решение задачи, код программы 34](#_Toc102337001)

[8.3. Тестирование программы с проверкой 35](#_Toc102337002)

[Задание 9 36](#_Toc102337003)

[9.1. Общая постановка задачи 36](#_Toc102337004)

[9.2. Решение задачи, код программы 36](#_Toc102337005)

[9.3. Тестирование программы с проверкой 37](#_Toc102337006)

[Задание 10 38](#_Toc102337007)

[10.1. Общая постановка задачи 38](#_Toc102337008)

[10.2. Решение задачи, код программы 38](#_Toc102337009)

[10.3. Тестирование программы с проверкой 39](#_Toc102337010)

[11 Задание 41](#_Toc102337011)

[11.1. Общая постановка задачи 41](#_Toc102337012)

[11.2. Решение задачи, код программы 41](#_Toc102337013)

[11.3. Тестирование программы с проверкой 41](#_Toc102337014)

[12 Задание 42](#_Toc102337015)

[12.1. Общая постановка задачи 42](#_Toc102337016)

[12.2. Решение задачи, код программы 42](#_Toc102337017)

[12.3. Тестирование с проверкой 43](#_Toc102337018)

# Задание 1

## 1.1. Общая постановка задачи

Введите массив символов из 15 элементов. Измените регистр символа: прописные заменить на строчные, а строчные на прописные.

## 1.2. Решение задачи, код программы

**import static** java.lang.System.***out***;  
  
**import** java.util.Scanner;  
  
**public class** Task91 {  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
 **public static void** main(String[] args) {  
 ***out***.println(**"Введите размерность массива:"**);  
 **int** n = *scanner*.nextInt();  
 **char**[] masChar = **new char**[n];  
 ***out***.println(**"Вводите строку по одному символу:"**);  
 **for** (**int** i = 0; i < n; i++) {  
 masChar[i] = *input*();  
 }  
 *check*(masChar);  
 **for** (**int** i = 0; i < masChar.**length**; i++) {  
 ***out***.print(masChar[i] + **" "**);  
 }  
 }  
  
 **public static char** input() {  
 **char** ch = **' '**;  
 String string = *scanner*.next();  
 **if** (string.length() == 1) {  
 ch = string.charAt(0);  
 }  
 **else** {  
 **while** (string.length() != 0) {  
 ***out***.println(**"Вы ввели не 1 символ, повторите ввод"**);  
 string = *scanner*.next();  
 **if** (string.length() == 1)  
 ch = string.charAt(0);  
 }  
 }  
 **return** ch;  
 }  
  
 **public static void** check(**char**[] masChar) {  
 **for** (**int** i = 0; i < masChar.**length**; i++){  
 **if** (Character.*isLowerCase*((masChar[i]))) {  
 masChar[i] = Character.*toUpperCase*(masChar[i]);  
 }  
 **else** {  
 masChar[i] = Character.*toLowerCase*(masChar[i]);  
 }  
 }  
 }  
}

## 1.3. Тестирование программы с проверкой

Для проверки в MS Excel создана таблица данных, которая заполнена буквами разного регистра. С помощью функций ПРОПИСН() и СТРОЧН(), заменяем регистр на строчный и прописной соответственно.

В таблице 1 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 1

Тестирование работы программы и проверка решения задачи 1

|  |  |
| --- | --- |
| № п/п | Решение Java |
| 1 |  |
| 2 |  |
| № п/п | Решение MS Excel |
| 1 |  |
| 2 |  |

Сравнение решения задачи с использованием двух прикладных пакетов показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 2

## 2.1. Общая постановка задачи

Дан массив символов, содержащий число в 16-ой системе счисления. Проверить правильность ввода этого числа (в его записи должны быть только символы 0, 1, 2,…, 9, A, B, C, D, E, F). 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 2-ую систему счисления, записать число в новый массив символов.

## 2.2. Решение задачи, код программы

**import static** java.lang.System.***out***;  
  
**import** java.util.Scanner;  
  
**public class** Task92 {  
 **public static void** main(String[] args) {  
 ***out***.println(**"------------------------------------------------------------------------------------------------"**);  
 ***out***.println(**"Условия ввода:"**);  
 ***out***.println(**"В одну строку записывается только один символ"**);  
 ***out***.println(**"Символы, которые могут использоваться при записи 16-го числа: 0, 1, 2,…, 9, A, B, C, D, E, F"**);  
 ***out***.println(**"------------------------------------------------------------------------------------------------"**);  
 *input*();  
 }  
  
 **public static void** input() {  
 Scanner scanner = **new** Scanner(System.***in***);  
 ***out***.println(**"Введите размерность массива:"**);  
 **int** n = scanner.nextInt();  
 **if** (n == 0) {  
 **throw new** IllegalArgumentException(**"Размерность массива не может быть ровна 0, введите n >= 1:"**);  
 }  
 **char**[] masChar = **new char**[n];  
 ***out***.println(**"Введите шестнадцатеричное число:"**);  
 **for** (**int** i = 0; i < n; i++) {  
 masChar[i] = *inputMasChar*(i, n);  
 }  
 *conversion*(masChar, n);  
 }  
   
 **public static char** inputMasChar(**int** i, **int** n ) {  
 Scanner scannerMas = **new** Scanner(System.***in***);  
 ***out***.println(**"masChar[ "** + i + **" ] = "**);  
 **char** ch = **' '**;  
 String string = scannerMas.next();  
 **if** (string.length() == 1) {  
 ch = string.charAt(0);  
 **if** (ch == **'0'** & (i == 0 || i == n-1)) { *// первый и последний символ не может быть равен 0* **while** (ch == **'0'**) {  
 ***out***.println(**"Вы ввели не верно A[0], оно не может быть равно 0"**);  
 ***out***.println(**"A[ "** + i + **" ] = "**);  
 string = scannerMas.next();  
 ch = string.charAt(0);  
 }  
 }  
 **while** (*check*(ch)) { *// проверка на правильность ввода* ***out***.println(**"Пожалуйста посмотрите условия ввода и введите только разрешенные символы:"**);  
 ***out***.println(**"masChar[ "** + i + **" ] = "**);  
 string = scannerMas.next();  
 ch = string.charAt(0);  
 **while** (string.length() != 1) { *// проверка на ввод 1 символа* ***out***.println(**"Вы ввели не 1 символ, повторите ввод:"**);  
 ***out***.println(**"masChar[ "** + i + **" ] = "**);  
 string = scannerMas.next();  
 }  
 **if** (ch == **'0'** & (i == 0 || i == n-1)) { *// первый и последний символ не может быть равен 0* **while** (ch == **'0'**) {  
 ***out***.println(**"Вы ввели не верно A[0], оно не может быть равно 0"**);  
 ***out***.println(**"A[ "** + i + **" ] = "**);  
 string = scannerMas.next();  
 ch = string.charAt(0);  
 }  
 }  
 }  
 }  
 **else** {  
 **while** (string.length() != 1) { *// проверка на ввод 1 символа* ***out***.println(**"Вы ввели не 1 символ, повторите ввод:"**);  
 ***out***.println(**"masChar[ "** + i + **" ] = "**);  
 string = scannerMas.next();  
 **if** (string.length() == 1) {  
 ch = string.charAt(0);  
 **if** (ch == **'0'** & (i == 0 || i == n-1)) { *// первый и последний символ не может быть равен 0* **while** (ch == **'0'**) {  
 ***out***.println(**"Вы ввели не верно A[0], оно не может быть равно 0"**);  
 ***out***.println(**"A[ "** + i + **" ] = "**);  
 string = scannerMas.next();  
 ch = string.charAt(0);  
 }  
 }  
 **while** (*check*(ch)) { *// проверка на правильность ввода* ***out***.println(**"Пожалуйста посмотрите условия ввода и введите только разрешенные символы:"**);  
 ***out***.println(**"masChar[ "** + i + **" ] = "**);  
 string = scannerMas.next();  
 ch = string.charAt(0);  
 }  
 }  
 }  
 }  
 **return** ch;  
 }  
  
 **public static boolean** check(**char** ch) { *// проверка на правильность ввода посредством* **int** count = 0; *// перебирания кода символа Unicode в десятичном исчислении* **for** (**int** k = 0; k <= 47; k++) {  
 **if** (ch == k) {  
 count++;  
 }  
 }  
 **for** (**int** k = 58; k <= 64; k++) {  
 **if** (ch == k) {  
 count++;  
 }  
 }  
 **for** (**int** k = 71; k <= 1000; k++) {  
 **if** (ch == k) {  
 count++;  
 }  
 }  
 **return** count >= 1; *// если символ совпал с неразрешенным, то возвращает true* }  
  
 **public static void** conversion(**char**[] masChar, **int** n) {  
 **int** count = 0;  
 **int** temp = 0;  
 **for** (**int** i = n - 1, j = 0; i >= 0; i--, j++) { *// перевод в 10-ю систему счисления* ***out***.println(**"temp1 = "** + temp);  
 ***out***.println(**"masChar["** + i + **"] = "** + masChar[i]);  
 temp += Character.*digit*(masChar[i], 16) \* Math.*pow*(16, j);  
 ***out***.println(**"temp2 = "** + temp);  
 }  
 ***out***.println(**"Число в 16-ой системе в 10-ой имеет вид: "** + temp);  
 **int** k = temp;  
 **while** (k > 0) { *// считаем размер массива masTwo* k /= 2;  
 count++;  
 }  
 **int**[] masTwo = **new int**[count];  
 **for** (**int** i = count - 1; i >= 0 ; i--) { *// перевод в 2-ю систему счисления* masTwo[i] = temp % 2;  
 temp /= 2;  
 ***out***.println(temp / 2);  
 }  
 *printMasTwo*(masTwo);  
 }  
  
 **public static void** printMasTwo(**int**[] masTwo) {  
 ***out***.print(**"Введенное вами число в 2-ой системе счисления имеет вид: "**);  
 **for** (**int** i = 0; i < masTwo.**length**; i++) {  
 ***out***.print(masTwo[i]);  
 }  
 }  
}

## 2.3. Тестирование программы с проверкой

В таблице 2 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 2

Тестирование работы программы и проверка решения задачи 2

|  |  |
| --- | --- |
| № п/п | Решение Java |
| 1 |  |
| 2 |  |
| 3 |  |
| № п/п | Решение в онлайн калькуляторе |
| 1 |  |
| 2 |  |
| 3 |  |

<https://calculatori.ru/perevod-chisel.html>

Сравнение решения задачи с использованием трех прикладных пакетов показало, что решения задачи в Java и в онлайн калькуляторе совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# 3 Задание

## 3.1. Общая 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 static** java.lang.System.***out***;  
  
**import** java.util.Scanner;  
  
**public class** Task93 {  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
 **public static void** main(String[] args) {  
 ***out***.println(**"Введите размерность массива:"**);  
 **int** n = *scanner*.nextInt();  
 ***out***.println(**"Введите любой массив символов по одному в строку:"**);  
 **char**[] masChar = **new char**[n];  
 **for** (**int** i = 0; i < n; i++) {  
 masChar[i] = *input*();  
 }  
 *check*(masChar, n);  
 }  
  
 **public static char** input() {  
 **char** ch = **' '**;  
 String string = *scanner*.next();  
 **if** (string.length() == 1) {  
 ch = string.charAt(0);  
 }  
 **else** {  
 **while** (string.length() != 1) {  
 ***out***.println(**"Вы ввели не один символ, повторите ввод"**);  
 string = *scanner*.next();  
 **if** (string.length() == 1) {  
 ch = string.charAt(0);  
 }  
 }  
 }  
 **return** ch;  
 }  
 **public static void** check(**char**[] masChar, **int** n) {  
 **int** countLat = 0, countCommas = 0, parameterI = 0, parameterJ = 0;  
 **boolean** checkU = **false**;  
 **boolean** checkBo = **false**;  
 **boolean** checkOb = **false**;  
 **boolean** checkDigit = **false**;  
 **boolean** checkSiSi1 = **false**;  
 **boolean** checkSjSj1 = **false**;  
 **for** (**int** i = 0; i < masChar.**length**; i++) { *// по длине всего массива* **for** (**int** k = 97; k <= 122; k++) { *// вхождение прописных латинских букв* **if** (masChar[i] == k) {  
 countLat++;  
 }  
 }  
 **if** (masChar[i] == **'ю'** | masChar[i] == **'Ю'**) { *// вхождение буквы "ю"* checkU = **true**;  
 }  
 **if** (masChar[i] == **','**) { *// вхождение запятой* countCommas++;  
 }  
 }  
 **for** (**int** i = 0; i < n - 1; i++) { *// идёт до предпоследнего символа* **if** (masChar[i] == **'в'** & masChar[i + 1] == **'о'**) { *// вхождение 'во'* checkBo = **true**;  
 }  
 **if** (masChar[i] == **'о'** & masChar[i + 1] == **'в'**) {*// вхождение 'ов'* checkOb = **true**;  
 }  
 **if** (Character.*isDigit*(masChar[i]) & Character.*isDigit*(masChar[i + 1])) { *// пара соседних одинаковых чисел* **if** (masChar[i] == masChar[i + 1]) {  
 checkDigit = **true**;  
 }  
 }  
 }  
 **for** (**int** k = 2; k < n - 1; k++) { *// идёт до предпоследнего символа* **if** ((Character.*isLowerCase*(masChar[k]) & Character.*isUpperCase*(masChar[k + 1])) |  
 (Character.*isUpperCase*(masChar[k]) & Character.*isLowerCase*(masChar[k + 1]))) {  
 *// si, si+1 это одинаковые буквы отличающиеся регистром* checkSiSi1 = **true**;  
 parameterI = k;  
 **break**;  
 }  
 }  
 ***out***.println(**"k = "** + parameterI);  
 **for** (**int** c = parameterI + 1; c < n - 1; c++) { *// идёт до предпоследнего символа* **if** (Character.*isDigit*(masChar[c])) {  
 **if** ((Character.*digit*(masChar[c], 10) == 0) & (Character.*digit*(masChar[c + 1], 10) == 0)) {  
 *// sj, sj+1 это 0* checkSjSj1 = **true**;  
 parameterJ = c;  
 }  
 }  
 }  
 *printAns*(countLat, checkU, countCommas, checkBo, checkOb, checkDigit,  
 checkSiSi1, checkSjSj1, parameterI, parameterJ);  
 }  
  
 **public static void** printAns(**int** countLat, **boolean** checkU, **int** countCommas, **boolean** checkBo, **boolean** checkOb,  
 **boolean** checkDigit, **boolean** checkSiSi1, **boolean** checkSjSj1,  
 **int** parameterI, **int** parameterJ) {  
 ***out***.println(**"Количество латинских прописных букв: "** + countLat);  
 **if** (checkU) {  
 ***out***.println(**"В массив входит буква 'ю'"**);  
 }  
 **if** (countCommas >= 2) {  
 ***out***.println(**"В массиве имеются две или более запятые"**);  
 }  
 **if** (checkBo) {  
 ***out***.println(**"В массив входит последовательность 'во'"**);  
 }  
 **if** (checkOb) {  
 ***out***.println(**"В массив входит последовательность 'ов'"**);  
 }  
 **if** (checkDigit) {  
 ***out***.println(**"В массив входит пара соседствующих одинаковых цифр"**);  
 }  
 **if** (checkSiSi1 & checkSjSj1) {  
 ***out***.println(**"В массиве есть такие натуральные i и j, что 1 < i < j < n и что si, si+1 "** +  
 **"это одинаковые буквы отличающиеся регистром, a sj, sj+1 это 0"**);  
 ***out***.println(**"Первое вхождение i = "** + parameterI);  
 ***out***.println(**"Первое вхождение i = "** + parameterJ);  
 }  
 }  
}

## 3.3. Тестирование программы с проверкой

Для проверки задания в MS Excel создана таблица данных. В таблицу B2:O3 записаны индексы массива и их значения. Далее по строкам идёт проверка A5, A13, A16 – аналитическая, A7, A9, A11 – с помощью формулы,

A7=ЕСЛИ(ИЛИ(B3="Ю"; B3="ю");1;0)

A9=ЕСЛИ(B3=",";1;0)

A11=ЕСЛИ(ИЛИ(И(B3="в";C3="о");И(B3="о";C3="в"));1;0)

В таблице 3 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 3

Тестирование работы программы и проверка решения задачи 3

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

Сравнение решения задачи с использованием трех прикладных пакетов показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 4

## 4.1. Общая постановка задачи

Задана матрица символов N×N. Определить построчно и по столбцам, есть ли в них выражения вида сos(«число»\* «число») = «число» или последовательность чисел. 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 static** java.lang.System.***out***;  
  
**import** java.util.Scanner;  
**import** java.util.ArrayList;  
  
**public class** Task94 {  
  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
  
 **static** ArrayList<Integer> *list* = **new** ArrayList<>();  
  
 **public static void** main(String[] args) {  
 ***out***.println(**"Введите размерность n квадратной матрицы:"**);  
 **int** n = *scanner*.nextInt();  
 **if** (n < 10) {  
 **throw new** IllegalArgumentException(**"n не может быть меньше 10, пожалуйста повторите ввод"**);  
 }  
 ***out***.println(**"Введите данные:"**);  
 **char**[][] masChar = **new char**[n][n];  
 **for** (**int** i = 0; i < n; i++) {  
 **for** (**int** j = 0; j < n; j++) {  
 masChar[i][j] = *inputMasChar*();  
 }  
 }  
 *printMasChar*(masChar, n);  
 *checkConstruction*(masChar, n);  
 }  
  
 **public static char** inputMasChar() {  
 **char** ch = **' '**;  
 String string = *scanner*.next();  
 **if** (string.length() == 1) {  
 ch = string.charAt(0);  
 }  
 **else** {  
 **while** (string.length() != 1) {  
 ***out***.println(**"Вы ввели не один символ, повторите ввод"**);  
 string = *scanner*.next();  
 **if** (string.length() == 1) {  
 ch = string.charAt(0);  
 }  
 }  
 }  
 **return** ch;  
 }  
  
 *// Ищет наличие конструкции вида: сos(«число»\* «число») = «число»* **public static void** checkConstruction (**char**[][] masChar, **int** n) {  
 **int** countForMethodAnswer = 0; *// для того, чтобы countLines, countColumns записывались в list 1 раз* **for** (**int** i = 0; i < n; i++) {  
 **for** (**int** j = 0; j < n - 9; j++) {  
 *// проверка по строкам на присутствие конструкции* **if** (masChar[i][j] == **'c'** & masChar[i][j+1] == **'o'** & masChar[i][j+2] == **'s'** & masChar[i][j+3] == **'('** &  
 masChar[i][j+7] == **')'** ) {  
 **if** (Character.*isDigit*(masChar[i][j+4]) & Character.*isDigit*(masChar[i][j+6]) &  
 Character.*isDigit*(masChar[i][j+9])) {  
 **if** (masChar[i][j+5] == **'\*'** & masChar[i][j+8] == **'='**) {  
 countForMethodAnswer++;  
 *printLineColumn*(masChar, i, j, 1, 0);  
 *checkTheCorrectnessOfTheConstruction*(Character.*digit*(masChar[i][j + 4], 10),  
 Character.*digit*(masChar[i][j + 6], 10),  
 Character.*digit*(masChar[i][j + 9], 10), 1, 0,  
 countForMethodAnswer);  
 }  
 }  
 }  
 *// проверка по столбцам на присутствие конструкции* **if** (masChar[j][i] == **'c'** & masChar[j+1][i] == **'o'** & masChar[j+2][i] == **'s'** & masChar[j+3][i] == **'('** &  
 masChar[j+7][i] == **')'** ) {  
 **if** (Character.*isDigit*(masChar[j+4][i]) & Character.*isDigit*(masChar[j+6][i]) &  
 Character.*isDigit*(masChar[j+9][i])) {  
 **if** (masChar[j+5][i] == **'\*'** & masChar[j+8][i] == **'='**) {  
 *printLineColumn*(masChar, i, j, 0, 1);  
 countForMethodAnswer = 2;  
 *checkTheCorrectnessOfTheConstruction*(Character.*digit*(masChar[j + 4][i], 10),  
 Character.*digit*(masChar[j + 6][i], 10),  
 Character.*digit*(masChar[j + 9][i], 10),0, 1,  
 countForMethodAnswer);  
 }  
 }  
 }  
 }  
 }  
 *answer*();  
 }  
  
 *// Проверяет правильность вычисления выражения сos(«число»\* «число») = «число»* **public static void** checkTheCorrectnessOfTheConstruction (**int** k, **int** m, **int** c, **int** isLine,  
 **int** isColumn, **int** countForMethodAnswer) {  
 **int** countLines = 0, countColumns = 0;  
 **if** (countForMethodAnswer == 1) {  
 *list*.add(countLines);  
 *list*.add(countColumns);  
 }  
 **if** (isLine == 1) {  
 **if** (Math.*cos*(k \* m) == c) {  
 countLines++;  
 *list*.set(0, countLines);  
 } **else** {  
 *list*.add(k);  
 *list*.add(m);  
 *list*.add(c);  
 }  
 }  
 **if** (isColumn == 1) {  
 **if** (Math.*cos*(k \* m) == c) {  
 countColumns++;  
 *list*.set(1, countColumns);  
 } **else** {  
 *list*.add(k);  
 *list*.add(m);  
 *list*.add(c);  
 }  
 }  
 }  
  
 *// Выводит выражения данного вида по строкам и по столбцам.* **public static void** printLineColumn (**char**[][] array, **int** k, **int** m, **int** isLine, **int** isColumn) {  
 ***out***.println();  
 **if** (isLine == 1) {  
 ***out***.println(**"В строчке "** + k + **" содержится конструкция:"**);  
 **for** (**int** i = m; i <= m + 9; i++) {  
 ***out***.print(array[k][i] + **"\t"**);  
 }  
 ***out***.println();  
 }  
 **else if** (isColumn == 1) {  
 ***out***.println(**"В столбце "** + k + **" содержится конструкция:"**);  
 **for** (**int** i = m; i <= m + 9; i++) {  
 ***out***.print(array[i][k] + **"\t"**);  
 }  
 }  
 }  
  
 **public static void** printMasChar (**char**[][] masChar, **int** n) {  
 ***out***.println(**"Матрица имеет вид:"**);  
 **for** (**int** i = 0; i < n; i++) {  
 **for** (**int** j = 0; j < n; j++) {  
 ***out***.print(masChar[i][j] + **"\t"**);  
 }  
 ***out***.println();  
 }  
 }  
  
 **public static void** answer() {  
 ***out***.println();  
 ***out***.println();  
 ***out***.println(**"Всего по строкам содержится "** + *list*.get(0) + **" правильно вычисленных конструкций."**);  
 ***out***.println(**"Всего по столбцам содержится "** + *list*.get(1) + **" правильно вычисленных конструкций."**);  
 **if** (*list*.get(0) == 0| *list*.get(1) == 0) {  
 ***out***.println(**"Выражения, которые были вычислены не правильно с корректировкой ответа:"**);  
 **for** (**int** i = 2; i < *list*.size() - 2; i = i + 3) {  
 ***out***.println(**"cos("** + *list*.get(i) + **" \* "** + *list*.get(i + 1) + **")"** + **" = "** + *list*.get(i + 2));  
 ***out***.println(**"cos("** + *list*.get(i) + **" \* "** + *list*.get(i + 1) + **")"** + **" = "** +  
 Math.*cos*(*list*.get(i) \* *list*.get(i + 1)));  
 ***out***.println();  
 }  
 }  
 }  
}

## 4.3. Тестирование программы с проверкой

Для проверки в MS Excel создана таблица данных. В ячейки B2:K9 записана сам массив данных. В ячейки B16, B17 – формулы для проверки правильности вычисления выражения. В ячейки G16, G17 – формулы для исправления вычисления выражения.

B16=ЕСЛИ(COS(F2\*H2)=K2; "Выражение вычислено правильно"; "Выражение вычислено неправильно")

B17=ЕСЛИ(COS(B6\*B8)=B11; "Выражение вычислено правильно"; "Выражение вычислено неправильно")

На рисунке 1 приведен вид решения в MS Excel.

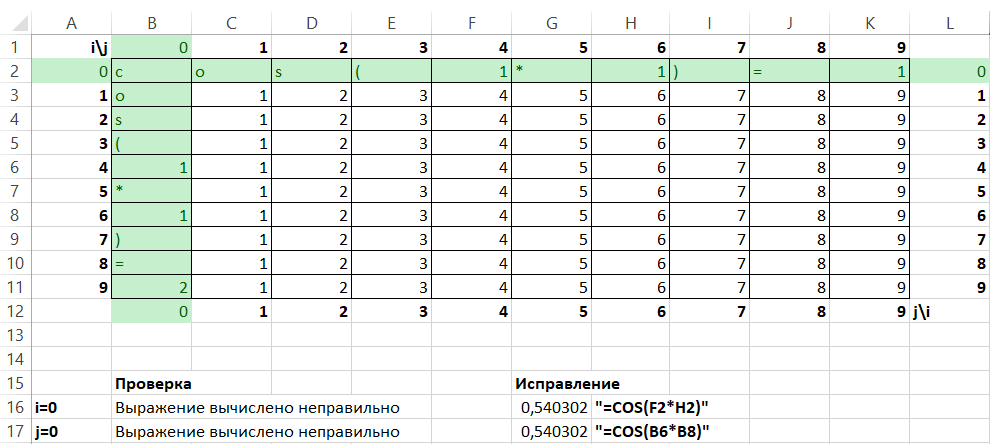


Рисунок 1. Вид решения в MS Excel задания 4

В таблице 4 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 4

Тестирование работы программы и проверка решения задачи 4

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

Сравнение решения задачи с использованием одного прикладных пакетов показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 5

## 5.1. Общая постановка задачи

В заданной строке посчитать количество символов цифр (от «0» до «9»).

## 5.2. Решение задачи, код программы

**import static** java.lang.System.***out***;  
  
**import** java.util.Scanner;  
  
**public class** Task95 {  
  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
  
 **public static void** main (String[] args) {  
 ***out***.println(**"Введите любую строку:"**);  
 String string = *scanner*.nextLine();  
 **int** count = 0;  
 **char**[] masString = string.toCharArray();  
 **for** (**char** s : masString) {  
 **if** (Character.*isDigit*(s)) {  
 count++;  
 }  
 }  
 ***out***.println(**"В ведённой строчке содержатся числа в количестве: "** + count);  
 }  
}

## 5.3. Тестирование программы с проверкой

Для проверки в MS Excel создана таблица данных, содержащая массив символов.

В таблице 5 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 5

Тестирование работы программы и проверка решения задачи 5

|  |  |
| --- | --- |
| № п/п | Решение Java |
| 1 |  |
| 2 |  |
| 3 |  |
| № п/п | Решение MS Excel |
| 1 |  |
| 2 |  |
| 3 |  |

Сравнение решения задачи с использованием трех прикладных пакетов показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 6

## 6.1. Общая постановка задачи

Дан массив строк, который содержит информацию о ветеринарных клиниках (каждая ячейка соответствует 1 клинике), не менее 20 строк. Создать две программы для регламентированного и нерегламентированного ввода данных по ветеринарным клиникам. Определить в каждой программе: количество клиник с номером; количество клиник по районам, в которых есть не менее 2 докторов высшей квалификации, повышение квалификации прошли более 30 % врачей за последние 5 лет с численностью пациентов более 100; количество клиник, в которых есть рентген с общей стоимостью оборудования клиники более 5 млн. руб.

- Первая программа: Сделать регламентированный ввод данных по ветеринарным клиникам. Каждая ячейка массива содержит информацию в определенном виде: Район города; Номер/Название ветеринарной клиники; Количество работников; Количество врачей; Количество врачей высшей категории; Количество врачей прошедших повышение квалификации за последние 5 лет; Общее число пациентов в базе; Число пациентов собак; Число пациентов кошек; Наличие рентгена; Наличие компьютеров; Количество компьютеров; Число клеток для передержки животных; Средний суммарный доход клиники; Общая стоимость оборудования клиники.

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

## 6.2. Решение задачи, код программы

### 6.2.1. Решение задачи, код первой программы

**import static** java.lang.System.***out***;  
  
**import** java.util.ArrayList;  
**import** java.util.Arrays;  
**import** java.util.Scanner;  
  
**public class** Task961 {  
  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
  
 **static** ArrayList<String> *listInf* = **new** ArrayList<>(); *// конструкция* **static** ArrayList<String> *listDistrict* = **new** ArrayList<>(); *// районы* **static** ArrayList<Integer> *listDistrictInf* = **new** ArrayList<>(); *// данные по районам* **public static void** main(String[] args) {  
 String[] masList = {**"Район города"**, **"Номер/Название ветеринарной клиники"**, **"Количество работников"**,  
 **"Количество врачей"**, **"Количество врачей высшей категории"**,  
 **"Количество врачей прошедших повышение квалификации за последние 5 лет"**,  
 **"Общее число пациентов в базе"**, **"Число пациентов собак"**, **"Число пациентов кошек"**, **"Наличие рентгена"**,  
 **"Наличие компьютеров"**, **"Количество компьютеров"**, **"Число клеток для передержки животных"**,  
 **"Средний суммарный доход клиники"**, **"Общая стоимость оборудования клиники"**};  
  
 *listInf*.addAll(Arrays.*asList*(masList));  
  
 String[] vetClinics = {**"vetClinic1"**, **"vetClinic2"**, **"vetClinic3"**, **"vetClinic4"**, **"vetClinic5"**, **"vetClinic6"**,  
 **"vetClinic7"**, **"vetClinic8"**, **"vetClinic9"**, **"vetClinic10"**, **"vetClinic11"**, **"vetClinic12"**, **"vetClinic13"**,  
 **"vetClinic14"**, **"vetClinic15"**, **"vetClinic16"**, **"vetClinic17"**, **"vetClinic18"**, **"vetClinic19"**, **"vetClinic20"**};  
 **int** n = 20;  
 String[] infAboutVetClinic = **new** String[n];  
 ***out***.println(**"Введите данные о ветеринарных клиниках (всё в одну строку) в виде:"**);  
 ***out***.println(**"""  
 Район города; Номер/Название ветеринарной клиники; Количество работников; Количество врачей;\s  
 Количество врачей высшей категории;\s  
 Количество врачей прошедших повышение квалификации за последние 5 лет;\s  
 Общее число пациентов в базе; Число пациентов собак; Число пациентов кошек; Наличие рентгена;\s  
 Наличие компьютеров; Количество компьютеров; Число клеток для передержки животных;\s  
 Средний суммарный доход клиники; Общая стоимость оборудования клиники."""**);  
  
 **for** (**int** i = 0; i < n; i++) {  
 infAboutVetClinic[i] = *checkInfAboutVetClinics*();  
 }  
  
 **for** (**int** i = 0; i < n; i++) {  
 ***out***.println(vetClinics[i] + **": "** + infAboutVetClinic[i]);  
 }  
 *analyzeInfAboutVetClinics*(infAboutVetClinic, n);  
 }  
  
 *// анализируем данные и выделяем то, что нужно по условию задачи* **public static void** analyzeInfAboutVetClinics (String[] infAboutVetClinic, **int** n) {  
 String[] splitInf;  
 **int**[] answer = **new int**[2];  
 **for** (**int** i = 0; i < n; i++) {  
 splitInf = infAboutVetClinic[i].split(**";"**);  
 **if** (splitInf[1].contains(**"/"**)) { *// если строка содержит "/", то это автоматически значит, что есть номер* answer[0]++;  
 }  
  
 **int** numberDocHighCategory = Integer.*parseInt*(splitInf[4]);  
 **int** numberDocHighCategoryFive = Integer.*parseInt*(splitInf[5]);  
 **int** totalNumberDoc = Integer.*parseInt*(splitInf[2]);  
 **int** totalNumberPatient = Integer.*parseInt*(splitInf[6]);  
 **boolean** conditions = numberDocHighCategory >= 2 & numberDocHighCategoryFive > totalNumberDoc \* 3.0 / 10 &  
 totalNumberPatient > 100;  
 **if** (i == 0) {  
 *listDistrict*.add(splitInf[0]);  
 **if** (conditions) {  
 *listDistrictInf*.add(1);  
 }  
 }  
 **else** {  
 **if** (!*listDistrict*.contains(splitInf[0])) {  
 *listDistrict*.add(splitInf[0]);  
 **if** (conditions) {  
 *listDistrictInf*.add(1);  
 }  
 } **else if** (*listDistrict*.contains(splitInf[0])) {  
 **int** index = *listDistrict*.indexOf(splitInf[0]);  
 **int** newElement = *listDistrictInf*.get(index) + 1;  
 *listDistrictInf*.set(index, newElement);  
 }  
 }  
 **int** totalCostEquipment = Integer.*parseInt*(splitInf[14]);  
 **if** ((splitInf[9].equals(**"есть"**) | splitInf[9].equals(**"Есть"**)) & totalCostEquipment > 5000000) {  
 answer[1]++;  
 }  
 }  
 *output*(answer);  
 }  
  
 *// проверка на структуру данных* **public static** String checkInfAboutVetClinics() {  
 String string = *scanner*.nextLine();  
 String[] strMasVetClinic = string.split(**";"**); *// создаем массив, в каждой ячейке - инф. в опр. виде* **while** (*cycleCheckInfAboutVetClinics*(strMasVetClinic)) { *// проверка праавильности ввода конструкции* ***out***.println(**"Пожалуйста, повторите ввод данных:"**);  
 string = *scanner*.nextLine();  
 strMasVetClinic = string.split(**";"**);  
 }  
  
 **return** string;  
 }  
  
 *// сам цикл по проверке* **public static boolean** cycleCheckInfAboutVetClinics(String[] strMasVetClinic) {  
 **boolean** marker = **false**;  
 **for** (**char** ch : strMasVetClinic[0].toCharArray()) {  
 **if** (Character.*isDigit*(ch)) {  
 ***out***.println(**"В позиции 1, где указывается "** + *listInf*.get(0) +  
 **", вы ввели не буквы, там должна быть буквенная строка!"**);  
 marker = **true**;  
 }  
 }  
  
 String[] strMasVetClinicBoxTwo = strMasVetClinic[1].split(**"/"**);  
 **if** (strMasVetClinic[1].contains(**"/"**)) {  
 **int** countTwoFirst = 0;  
 **int** countTwoSecond = 0;  
 **for** (**char** ch : strMasVetClinicBoxTwo[0].toCharArray()) {  
 **if** (!Character.*isDigit*(ch)) {  
 marker = **true**;  
 countTwoFirst++;  
 }  
 }  
 **for** (**char** ch : strMasVetClinicBoxTwo[1].toCharArray()) {  
 **if** (!Character.*isLetter*(ch)) {  
 marker = **true**;  
 countTwoSecond++;  
 }  
 }  
 **if** (countTwoFirst > 0) {  
 ***out***.println(**"В позиции 2, где указывается номер клиники, вы ввели не число!"**);  
 }  
 **if** (countTwoSecond > 0) {  
 ***out***.println(**"В позиции 2, где указывается название клиники, вы ввели не буквы!"**);  
 }  
 }  
 **else if** (!strMasVetClinic[1].contains(**"/"**)) {  
 **int** countTwo = 0;  
 **for** (**char** ch : strMasVetClinic[1].toCharArray()) {  
 **if** (!Character.*isLetter*(ch)) {  
 countTwo++;  
 }  
 }  
 **if** (countTwo > 0) {  
 ***out***.println(**"В позиции 2, где указывается название клиники, вы ввели не буквы!"**);  
 }  
 }  
  
 **for** (**int** i = 2; i < 9; i++) {  
 **int** countThreeToNine = 0;  
 **for** (**char** ch : strMasVetClinic[i].toCharArray()) {  
 **if** (Character.*isLetter*(ch)) {  
 countThreeToNine++;  
 }  
 }  
 **if**(countThreeToNine > 0) {  
 marker = **true**;  
 ***out***.println(**"В позиции "** + (i + 1) + **" ("** + *listInf*.get(i) + **") "** + **"вы ввели не число!"**);  
 }  
 }  
  
 **for** (**int** i = 9; i < 11; i++) {  
 **int** countTenToEleven = 0;  
 **for** (**char** ch : strMasVetClinic[i].toCharArray()) {  
 **if** (!Character.*isLetter*(ch)) {  
 countTenToEleven++;  
 }  
 }  
 **if** (countTenToEleven > 0) {  
 marker = **true**;  
 ***out***.println(**"В позиции "** + (i + 1) + **" ("** + *listInf*.get(i) + **") "** + **"вы ввели не буквы, "** +  
 **"укажите наличие (есть/нет)!"**);  
 }  
 **if**(countTenToEleven == 0 & !strMasVetClinic[i].equals(**"есть"**) & !strMasVetClinic[i].equals(**"Есть"**) &  
 !strMasVetClinic[i].equals(**"нет"**) & !strMasVetClinic[i].equals(**"Нет"**)) {  
 ***out***.println(**"В позиции "** + (i + 1) + **" ("** + *listInf*.get(i) + **") "** + **"укажите наличие (есть/нет)"**);  
 }  
 }  
  
 **for** (**int** i = 11; i < 15; i++) {  
 **int** countTwelveToFifteen = 0;  
 **for** (**char** ch : strMasVetClinic[i].toCharArray()) {  
 **if** (!Character.*isDigit*(ch)) {  
 countTwelveToFifteen++;  
 }  
 }  
 **if** (countTwelveToFifteen > 0) {  
 marker = **true**;  
 ***out***.println(**"В позиции "** + (i + 1) + **" ("** + *listInf*.get(i) + **") "** + **"вы ввели не число!"**);  
 }  
 }  
  
 **return** marker;  
 }  
  
 **public static void** output(**int**[] answer) {  
 ***out***.println(**"Количество клиник с номером: "** + answer[0]);  
 **for** (**int** i = 0; i < *listDistrict*.size(); i++) {  
 ***out***.println(**"Район: "** + *listDistrict*.get(i) + **", количество клиник, удовлетворяющих условию: "** +  
 *listDistrictInf*.get(i));  
 }  
 ***out***.println(**"Количество клиник, где есть рентген с общей стоимостью оборудования клиники более 5 млн. руб: "** +  
 answer[1]);  
 }  
}

### 6.2.2. Решение задачи, код второй программы

**import static** java.lang.System.***out***;  
  
**import** java.util.ArrayList;  
**import** java.util.Scanner;  
  
**public class** Task962 {  
  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
  
 **static** ArrayList<String> *list* = **new** ArrayList<>();  
  
 **public static void** main(String[] args) {  
 String[] vetClinics = {**"vetClinic1"**, **"vetClinic2"**, **"vetClinic3"**, **"vetClinic4"**, **"vetClinic5"**, **"vetClinic6"**,  
 **"vetClinic7"**, **"vetClinic8"**, **"vetClinic9"**, **"vetClinic10"**, **"vetClinic11"**, **"vetClinic12"**, **"vetClinic13"**,  
 **"vetClinic14"**, **"vetClinic15"**, **"vetClinic16"**, **"vetClinic17"**, **"vetClinic18"**, **"vetClinic19"**, **"vetClinic20"**};  
 **int** n = 20;  
 String[] infAboutVetClinic = **new** String[n];  
 ***out***.println(**"Введите данные о ветеринарных клиниках (всё в одну строку):"**);  
 ***out***.println(**"""  
 Район города; Номер/Название ветеринарной клиники; Количество работников; Количество врачей;\s  
 Количество врачей высшей категории;\s  
 Количество врачей прошедших повышение квалификации за последние 5 лет;\s  
 Общее число пациентов в базе; Число пациентов собак; Число пациентов кошек; Наличие рентгена;\s  
 Наличие компьютеров; Количество компьютеров; Число клеток для передержки животных;\s  
 Средний суммарный доход клиники; Общая стоимость оборудования клиники."""**);  
  
 **for** (**int** i = 0; i < n; i++) {  
 infAboutVetClinic[i] = *scanner*.nextLine();  
 }  
  
 **for** (**int** i = 0; i < n; i++) {  
 ***out***.println(vetClinics[i] + **": "** + infAboutVetClinic[i]);  
 }  
 *checkInfAboutVetClinic*(infAboutVetClinic, n);  
 }  
  
 **public static void** checkInfAboutVetClinic (String[] infAboutVetClinic, **int** n) {  
 **int** c = 15;  
 **int**[] answer = **new int**[3];  
 String[] splitInf;  
 String[] masDistrict = {**"Рай."**, **"рай."**, **"р."**, **"Р."**, **"Р-н"**, **"р-н"**};  
 String[] masCostEquipment = {**"стоимость"**, **"Стоимость"**, **"Стоимость оборудования"**, **"стоимость оборудования"**};  
 **for** (**int** i = 0; i < n; i++) {  
 splitInf = infAboutVetClinic[i].split(**";"**);  
 **for** (**int** j = 0; j < masDistrict.**length**; j++) {  
 **for** (**int** k = 0; k < c; k++) {  
 **if** (splitInf[k].contains(masDistrict[j])) {  
 **if** (i == 0) {  
 *list*.add(splitInf[k]);  
 }  
 **else if** (!*list*.contains(splitInf[k])) {  
 *list*.add(splitInf[k]);  
 answer[1]++;  
 }  
 }  
 }  
 }  
  
 **int** countNameVetClinic = 0;  
 **for** (**int** j = 0; j < c; j++) {  
 **if** (splitInf[j].contains(**"/"**)) {  
 answer[0]++;  
 }  
 **for** (**char** ch : splitInf[j].toCharArray()) {  
 **if** (Character.*isDigit*(ch)) {  
 countNameVetClinic++;  
 }  
 }  
 **if** (countNameVetClinic > 1) {  
 answer[0]++;  
 }  
 }  
  
 **for** (**int** j = 0; j < masCostEquipment.**length**; j++) {  
 **for** (**int** k = 0; k < c; k++) {  
 **if** ((splitInf[k].contains(**"Есть"**) | splitInf[k].contains(**"есть"**)) &  
 splitInf[k].contains(masCostEquipment[j])) {  
 answer[2]++;  
 }  
 }  
 }  
 }  
 *output*(answer);  
 }  
  
 **public static void** output(**int**[] answer) {  
 ***out***.println(**"Количество клиник с номером: "** + answer[0]);  
 **for** (**int** i = 0; i < *list*.size(); i++) {  
 ***out***.println(**"Район: "** + *list*.get(i) + **", количество клиник, удовлетворяющих условию: "** + answer[1]);  
 }  
 ***out***.println(**"Количество клиник, где есть рентген с общей стоимостью оборудования клиники более 5 млн. руб: "** +  
 answer[2]);  
 ***out***.println(*list*.size());  
 }  
  
}

## 6.3. Тестирование программы с проверкой

### 6.3.1. Тестирование первой программы с проверкой

Для проверки в MS Excel создана таблица данных, которая содержит информацию о ветеринарных клиниках.

В таблице 6 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 6

Тестирование работы первой программы и проверка решения задачи 6

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

Сравнение решения задачи с использованием одного прикладного пакета показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

### 6.3.2. Тестирование второй программы с проверкой

Для проверки в MS Excel создана таблица данных, которая содержит информацию о ветеринарных клиниках.

В таблице 7 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 7

Тестирование работы второй программы и проверка решения задачи 6

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

Сравнение решения задачи с использованием одного прикладного пакета показало, невозможно однозначно определить то, что просят в условии задачи, при нерегламентированном выводе, поэтому всегда следует использовать регламентированный ввод данных.

# 7 Задание

## 7.1. Общая постановка задачи

В заданной строке заменить каждый символ «\*» числом, соответствующим номеру по порядку вхождения этого символа в строку.

## 7.2. Решение задачи, код программы

**import static** java.lang.System.***out***;  
  
**import** java.util.ArrayList;  
**import** java.util.Scanner;  
  
**public class** Task97 {  
  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
  
 **static** ArrayList<Integer> *listIndex* = **new** ArrayList<>();  
 **static** ArrayList<String> *listString* = **new** ArrayList<>();  
  
 **public static void** main(String[] args) {  
 ***out***.println(**"Введите любую строку:"**);  
 String string = *scanner*.nextLine();  
 String[] masString = string.split(**""**);  
  
 **int** count = 0;  
 **for** (**int** i = 0; i < masString.**length**; i++) {  
 *listString*.add(masString[i]);  
 **if** (masString[i].equals(**"\*"**)) {  
 count++;  
 *listIndex*.add(count);  
 }  
 }  
  
 **for** (**int** i = 0; i < *listIndex*.size(); i++) {  
 *listString*.set(*listIndex*.get(i) - 1, *listIndex*.get(i).toString());  
 }  
 ***out***.println(**"Преобразованная строка имеет вид:"**);  
 **for** (String str :*listString*) {  
 ***out***.print(str);  
 }  
 }  
}

## 7.3. Тестирование программы с проверкой

Для проверки в MS Excel создана таблица данных, содержащая исходную и преобразованную строку.

В таблице 8 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 8

Тестирование работы программы и проверка решения задачи 7

|  |  |
| --- | --- |
| № п/п | Решение Java |
| 1 |  |
| 2 |  |
| 3 |  |
| № п/п | Решение MS Excel |
| 1 |  |
| 2 |  |
| 3 |  |

Сравнение решения задачи с использованием трех прикладных пакетов показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 8

## 8.1. Общая постановка задачи

Вычеркнуть из текста минимальное количество предложений так, чтобы у любых двух оставшихся предложений было хотя бы одно общее слово.

## 8.2. Решение задачи, код программы

**import** java.util.ArrayList;  
**import** java.util.Arrays;  
**import** java.util.Collections;  
**import** java.util.Scanner;  
  
**import static** java.lang.System.***out***;  
  
  
**public class** Task98 {  
  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
  
 **static** ArrayList<String[]> *listSentence* = **new** ArrayList<>();  
 **static** ArrayList<String> *listCommonWords* = **new** ArrayList<>();  
  
 **public static void** main(String[] args) {  
 ***out***.println(**"Введите любой текст:"**);  
 String text = *scanner*.nextLine();  
 String[] masSplitText = text.split(**"\\. "**);  
 String[] masSentence;  
 **for** (String s : masSplitText) {  
 masSentence = s.split(**"\s"**);  
 *listSentence*.addAll(Collections.*singleton*(masSentence));  
 }  
  
 **for** (**int** i = 0; i < *listSentence*.size() - 1; i++) {  
 **int** countMin = 0;  
 **int** countMax = 1;  
 **int** a;  
 **for** (**int** j = 0; j < *listSentence*.get(i).**length**; j++) {  
 **for** (**int** k = i + 1; k < *listSentence*.size(); k++) {  
 **for** (**int** m = 0; m < *listSentence*.get(k).**length**; m++) {  
 **if** (*listSentence*.get(i)[j].equalsIgnoreCase(*listSentence*.get(k)[m])) {  
 countMin++;  
 **if** (countMin > countMax) { *// выбираем слово, которое чаще всего встречается  
 listCommonWords*.add(*listSentence*.get(i)[j]);  
 a = countMax;  
 countMax = countMin;  
 countMin = a;  
 }  
 }  
 }  
 }  
 }  
 }  
  
 **for** (**int** i = 0; i < *listSentence*.size(); i++) {  
 **for** (**int** j = 0; j < *listSentence*.get(i).**length**; j++) { *// если слово есть в предложении, то выводим* **if** (*listSentence*.get(i)[j].equalsIgnoreCase(*listCommonWords*.get(*listCommonWords*.size() - 1))) {  
 *output*(i);  
 }  
 }  
 }  
 }  
  
 **public static void** output(**int** i) {  
 ***out***.println(Arrays.*toString*(*listSentence*.get(i)));  
 }  
}

## 8.3. Тестирование программы с проверкой

Для проверки в MS Excel создана таблица данных с исходным текстом и преобразованным.

В таблице 9 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 9

Тестирование работы программы и проверка решения задачи 8

|  |  |
| --- | --- |
| № п/п | Решение Java |
| 1 |  |
| 2 |  |
| 3 |  |
| № п/п | Решение MS Excel |
| 1 |  |
| 2 |  |
| 3 |  |

Сравнение решения задачи с использованием трех прикладных пакетов показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 9

## 9.1. Общая 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 static** java.lang.System.***out***;  
  
**import** java.util.Scanner;  
**import** java.util.regex.\*;  
  
**public class** Task99 {  
  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
  
 **public static void** main(String[] args) {  
 ***out***.println(**"Введите любой текст:"**);  
 String text = *scanner*.nextLine();  
 String[] masSplitText = text.split(**"\\.\s"**);  
 **int** count = 0;  
  
 **for** (**int** i = 0; i < masSplitText.**length**; i++) {  
 Pattern patternFirst = Pattern.*compile*(**"%.+?%"**);  
 Pattern patternSecond = Pattern.*compile*(**"%.+?\\$"**);  
 Pattern patternThird = Pattern.*compile*(**"\\$.+?%"**);  
 Matcher matcherFirst = patternFirst.matcher(masSplitText[i]);  
 Matcher matcherSecond = patternSecond.matcher(masSplitText[i]);  
 Matcher matcherThird = patternThird.matcher(masSplitText[i]);  
 **if** (matcherFirst.find()) {  
 count++;  
 masSplitText[i] = matcherFirst.replaceAll(**""**);  
 }  
 **if** (matcherSecond.find()) {  
 count++;  
 masSplitText[i] = matcherSecond.replaceAll(**""**);  
 }  
 **if** (matcherThird.find()) {  
 count++;  
 masSplitText[i] = matcherThird.replaceAll(**""**);  
 }  
 }  
  
 ***out***.println();  
 **if** (count >= 1) {  
 **for** (String str : masSplitText) {  
 ***out***.print(str + **". "**);  
 }  
 }  
 **if** (count == 0) {  
 ***out***.println(**"Комментариев в тексте не найдено."**);  
 }  
 }  
}

## 9.3. Тестирование программы с проверкой

Для проверки в MS Excel создана таблица данных с исходным текстом и преобразованным текстом.

В таблице 10 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 10

Тестирование работы программы и проверка решения задачи 9

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

Сравнение решения задачи с использованием одного прикладного пакета показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 10

## 10.1. Общая постановка задачи

Дан текст, который содержит данные определенной структуры, разделителем данных об одном объекте является «;», разделитель между объектами «.». «Международная компания»: название; интернет сайт; телефон; факс; адрес главного офиса (почтовый индекс, страна, область, район, город, улица, дом, квартира) продолжительность пребывания на мировом рынке; количество сотрудников; количество филиалов в Европе. Разбить данные по объектам на массив строк. Посчитать количество компаний, в которых первые 5 цифр факса совпадает с номером телефона, а последняя цифра телефона совпадает с последней цифрой номера дома (адреса главного офиса). Вывести международные компании, количество сотрудников у которых больше 10000.

## 10.2. Решение задачи, код программы

**import** java.util.ArrayList;  
**import** java.util.Collections;  
**import** java.util.Scanner;  
  
**import static** java.lang.System.***out***;  
  
**public class** Task910 {  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
  
 **static** ArrayList<String[]> *listOfCompanies* = **new** ArrayList<>();  
 **static** ArrayList<String> *listNameOfCompaniesCondition2* = **new** ArrayList<>();  
 **static** ArrayList<Integer> *listNumbOfEmployee* = **new** ArrayList<>();  
 **static** ArrayList<String> *listNameOfCompaniesCondition1* = **new** ArrayList<>();  
  
 **public static void** main(String[] args) {  
 ***out***.println(**"Введите текст:"**);  
 String text = *scanner*.nextLine();  
 String[] masSplitText = text.split(**"\\. "**);  
 String[] masSentence;  
 **for** (String s : masSplitText) {  
 masSentence = s.split(**";\s"**);  
 *listOfCompanies*.addAll(Collections.*singleton*(masSentence));  
 }  
  
 **int** count = 0;  
 **int** numbOfEmployees = 10000;  
 String firstFivePhoneNumb = **""**;  
 String firstFiveFaxNumb = **""**;  
 String lastDigitPhoneNumb = **""**;  
 String[] masAddress;  
 String lastDigitNumbBuilding = **""**;  
 **int** numbOfEmployeesInCompany = 0;  
 **for** (**int** i = 0; i < *listOfCompanies*.size(); i++) {  
 **for** (**int** j = 0; j < *listOfCompanies*.get(i).**length**; j++) {  
 lastDigitPhoneNumb = *listOfCompanies*.get(i)[2].substring(11);  
 masAddress = *listOfCompanies*.get(i)[4].split(**",\s"**);  
 lastDigitNumbBuilding = masAddress[6].substring(masAddress[6].length() - 1);  
 firstFivePhoneNumb = *listOfCompanies*.get(i)[2].substring(2, 7);  
 firstFiveFaxNumb = *listOfCompanies*.get(i)[3].substring(2, 7);  
 numbOfEmployeesInCompany = Integer.*parseInt*(*listOfCompanies*.get(i)[6]);  
 }  
 **if** (firstFivePhoneNumb.equals(firstFiveFaxNumb) & lastDigitPhoneNumb.equals(lastDigitNumbBuilding)) {  
 count++;  
 *listNameOfCompaniesCondition1*.add(*listOfCompanies*.get(i)[0]);  
 }  
 **if** (numbOfEmployeesInCompany > numbOfEmployees) {  
 *listNameOfCompaniesCondition2*.add(*listOfCompanies*.get(i)[0]);  
 *listNumbOfEmployee*.add(numbOfEmployeesInCompany);  
 }  
 }  
 *output*(count);  
 }  
  
 **public static void** output(**int** count) {  
 ***out***.println();  
 ***out***.printf(**"Количество компаний, в которых первые 5 цифр факса совпадает с номером телефона, "** +  
 **"\nа последняя цифра телефона совпадает с последней цифрой номера дома: %d"**, count);  
 ***out***.println();  
 ***out***.println(**"Это компании:"**);  
 **for** (**int** i = 0; i < *listNameOfCompaniesCondition1*.size(); i++) {  
 ***out***.println(*listNameOfCompaniesCondition1*.get(i));  
 }  
 ***out***.println();  
 ***out***.println(**"Международные компании, количество сотрудников у которых больше 10000, представлены ниже:"**);  
 **for** (**int** i = 0; i < *listNameOfCompaniesCondition2*.size(); i++) {  
 ***out***.printf(*listNameOfCompaniesCondition2*.get(i) + **", количество сотрудников: %d"**, *listNumbOfEmployee*.get(i));  
 ***out***.println();  
 }  
 }  
}

## 10.3. Тестирование программы с проверкой

Для проверки в MS Excel создана таблица данных, содержащая информацию о международных компаниях.

В таблице 11 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 11

Тестирование работы программы и проверка решения задачи 10

|  |  |
| --- | --- |
| № п/п | Решение Java |
| 1 |  |
| № п/п | Решение MS Excel |
| 2 |  |

Сравнение решения задачи с использованием одного прикладного пакета показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# 11 Задание

## 11.1. Общая 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;  
**import** java.util.regex.Matcher;  
**import** java.util.regex.Pattern;  
  
**import static** java.lang.System.***out***;  
  
**public class** Task911 {  
  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
  
 **public static void** main(String[] args) {  
 ***out***.println(**"Введите текст:"**);  
 String text = *scanner*.nextLine();  
 String[] splitText = text.split(**"\s"**);  
 *check*(splitText);  
 }  
  
 **public static void** check(String[] masText) {  
 String[] pattern = {**"!.+!"**, **"\\?.+\\?"**, **"\\.\\.\\..+\\."**};  
 String[] replace = {**"!"**, **"?"**, **"..."**};  
 Matcher matcher;  
 Pattern patternNow;  
  
 **for** (**int** i = 0; i < masText.**length**; i++) {  
 **for** (**int** j = 0; j < pattern.**length**; j++) {  
 patternNow = Pattern.*compile*(pattern[j]);  
 matcher = patternNow.matcher(masText[i]);  
 **if** (matcher.find()) {  
 masText[i] = matcher.replaceAll(replace[j]);  
 }  
 }  
 }  
  
 ***out***.println();  
 ***out***.print(**"Отредактированный текст:"**);  
 ***out***.println();  
 **for** (String string : masText) {  
 ***out***.print(string + **"\s"**);  
 }  
 }  
}

## 11.3. Тестирование программы с проверкой

Для проверки в MS Excel создана таблица данных, содержащая исходные и преобразованный данные.

В таблице 12 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 12

Тестирование работы программы и проверка решения задачи 11

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

Сравнение решения задачи с использованием одного прикладного пакета показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# 12 Задание

## 12.1. Общая постановка задачи

Дан массив строк содержащих информацию об участниках математического клуба (возраст, пол, IQ, средний балл, место учебы, место жительства, численность семьи), строки заполнены не структурировано, зоны могут быть заполнены многоточиями. В цикле с использование регулярных выражений (массив данных) выполнить автозамену информации: многоточия заменить на фразу «Заполнить», ввести сокращение женского и мужского пола, ввести сокращения слова город и его различных записей до «г.», ввести сокращение край или область и их различных вариантов записи до «кр.» или «обл.». Исходную и измененную информацию вывести построчно: сначала исходная строка, затем измененная.

## 12.2. Решение задачи, код программы

**import static** java.lang.System.***out***;  
  
**import** java.util.Scanner;  
**import** java.util.regex.Matcher;  
**import** java.util.regex.Pattern;  
  
**public class** Task912 {  
  
 **static** Scanner *scanner* = **new** Scanner(System.***in***);  
  
 **public static void** main(String[] args) {  
 ***out***.println(**"Введите количество участников математического клуба:"**);  
 **int** n = *scanner*.nextInt();  
 ***out***.println(**"Введите данные об учениках в виде:"**);  
 ***out***.println(**"возраст, пол, IQ, средний балл, место учебы, место жительства, численность семьи"**);  
 String[] infAboutStudents = **new** String[n];  
 *scanner*.nextLine();  
 **for** (**int** i = 0; i < n; i++) {  
 infAboutStudents[i] = *scanner*.nextLine();  
 }  
 *check*(infAboutStudents);  
 }  
  
 **public static void** check(String[] infAboutStudents) {  
 ***out***.println();  
 String[][] masPatternAll = {{**"Мужской"**, **"мужской"**, **"мужск"**, **"Мужск"**, **"Муж"**, **"муж"**},  
 {**"Женский"**, **"женский"**, **"женск"**, **"Женск"**, **"Жен"**, **"жен"**}, {**"Город"**, **"город"**, **"Гор"**, **"гор"**},  
 {**"Область"**, **"область"**, **"Обл"**, **"обл"**}, {**"Край"**, **"край"**, **"Кр"**, **"кр"**}, {**"\\.\\.\\."**}};  
 String[] replaceAll = {**"м\\."**, **"ж\\."**, **"г\\."**, **"обл\\."**, **"кр\\."**, **"Заполнить"**};  
 Pattern pattern;  
 Matcher matcher;  
 **for** (**int** k = 0; k < infAboutStudents.**length**; k++) {  
 ***out***.println((k + 1) + **" ученик:"**);  
 ***out***.println(infAboutStudents[k]);  
 **for** (**int** i = 0; i < masPatternAll.**length**; i++) {  
 **for** ( **int** j = 0; j < masPatternAll[i].**length**; j++) {  
 pattern = Pattern.*compile*(masPatternAll[i][j]);  
 matcher = pattern.matcher(infAboutStudents[k]);  
 **if** (matcher.find()) {  
 infAboutStudents[k] = matcher.replaceAll(replaceAll[i]);  
 **break**;  
 }  
 }  
 }  
 ***out***.println(infAboutStudents[k]);  
 ***out***.println(**"----------------------------------------------------------------------------------"**);  
 }  
 }  
}

## 12.3. Тестирование с проверкой

Для тестирования в MS Excel создана таблица данных, содержащая исходные строки и преобразованные.

В таблице 13 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением в MS Excel.

Таблица 13

Тестирование работы программы и проверка решения задачи 12

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

Сравнение решения задачи с использованием трех прикладных пакетов показало, что решения задачи в Java и в MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.