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 группу ИСТ-22-1б Язов М.М.

Проверил: Нетбай Георгий Владимирович

Пермь, 2022

**Содержание**

[Задание 1 3](#_Toc116149685)

[1.1. Постановка задачи 3](#_Toc116149686)

[1.2. Решение задачи, код программы 3](#_Toc116149687)

[1.3. Тестирование работы программы с проверкой 3](#_Toc116149688)

[2.1. Постановка задачи 5](#_Toc116149689)

[2.2. Решение задачи, код программы 5](#_Toc116149690)

[2.3. Тестирование работы программы с проверкой 5](#_Toc116149691)

[3.1. Постановка задачи 6](#_Toc116149692)

[3.2. Решение задачи, код программы 6](#_Toc116149693)

[3.3. Тестирование работы программы с проверкой 6](#_Toc116149694)

[4.1. Постановка задачи 7](#_Toc116149695)

[4.2. Решение задачи, код программы 7](#_Toc116149696)

[4.3. Тестирование работы программы с проверкой 7](#_Toc116149697)

# Задание 1

## 1.1. Постановка задачи

Ввести массив из 10 символов. Если символ массива является прописной буквой латинского алфавита (т. е. буквой от «а» до «z»), замените символ, стоящий перед прописной буквой на пробел

## 1.2. Решение задачи, код программы

import *java.util.Arrays*;  
import *java.util.Random*;  
import *java.util.Scanner*;  
  
public class *exc9\_1* {  
 public static void main(*String*[] *args*) {  
 *Scanner* input = new Scanner(*System*.in);  
 *Random* random = new Random();  
 char[] arr = new char[10];  
 *String* alphabet = "qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM";  
 *String*[] chars = new String[10];  
  
 for (int i = 0; i < 10; i++) {  
 char c = alphabet.charAt(random.nextInt(alphabet.length()));  
 chars[i] = *String*.*valueOf*(c);  
 }  
 *System*.out.println(*Arrays*.*toString*(chars));  
  
 for (int i = 0; i < 10; i++) {  
 arr[i] = chars[i].charAt(0);  
 }  
  
 for (int i = 1; i < arr.length; i++) {  
 if (!*Character*.*isUpperCase*(arr[i])) {  
 arr[i - 1] = ' ';  
 }  
 }  
 *System*.out.println(*Arrays*.*toString*(arr));  
 }  
}

## 1.3. Тестирование работы программы

Таблица 1

Тестирование работы программы и проверка результатов решения

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

**Задача 2**

## 2.1. 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 3-ую систему счисления, записать число в новый массив символов.

## 2.2. Решение задачи, код программы

import *java.util.Scanner*;  
  
public class *exc9\_2* {  
 public static void main(*String*[] *args*) {  
 *Scanner* input = new Scanner(*System*.in);  
 *String* hexNumber;  
  
 while (true) {  
 *System*.out.print("Введите число в 16-ой системе счисления: ");  
 hexNumber = input.nextLine().toUpperCase();  
  
 boolean isValidHex = hexNumber.matches("[0**-**9A**-**F]+");  
 if (isValidHex) {  
 break;  
 } else {  
 *System*.out.println("Неверный ввод. В числе должны быть только символы 0-9 и A-F.");  
 }  
 }  
  
 int decimalNumber = *Integer*.*parseInt*(hexNumber, 16);  
 *String* ternaryNumber = *Integer*.*toString*(decimalNumber, 3);  
  
 *System*.out.println("Число в 3-й системе счисления: " + ternaryNumber);  
 }  
}

## 2.3. Тестирование работы программы

Таблица 2

Тестирование работы программы и проверка результатов решения

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

**Задача 3**

## 3.1. Постановка задачи

1. Дан массив символов. Проанализировать массив и ответить на вопросы:

а) Определить количество строчных русских букв.

б) Выяснить, входит ли в последовательность символ, введенный с клавиатуры, если входит, то посчитать сколько раз*.*

в) Выяснить, верно ли, что среди символов имеются 2 и более цифры, входящие в цифру, введенную с клавиатуры, например *70293.*

г) Выяснить, имеется ли среди символов пара соседствующих скобок «(, ), {, }, [, ]» если скобки присутствуют в последовательности, то выяснить есть ли закрытые пары разного сочетания и вывести ответ*.*

д) Выяснить, имеется ли среди символов более двух подряд идущих пробела.

е) Выяснить, верно ли. что существуют такие натуральные *i* и *j,* что *i<k <j < п* и что *si*, и *si+1* убывающая последовательность цифр, a *sj* и *sj+1* возрастающая последовательность цифр.

## 3.2. Решение задачи, код программы

import *java.util.Arrays*;  
import *java.util.Random*;  
import *java.util.Scanner*;  
  
public class *exc9\_3* {  
 public static void main(*String*[] *args*) {  
 *Scanner* input = new Scanner(*System*.in);  
  
 *System*.out.println("Введите что-нибудь: ");  
 *String* symbol = input.nextLine();  
  
 *System*.out.println("Длина массива: ");  
 int len = input.nextInt();  
  
 *String*[] arr = new String[len];  
 *String* alphabet = "ё1234567890-=йцукенгшщзхъфывапролджэячсмитьбю.!№;%:?\*()\_+, /[]{}@#$^&~ЁЙЦУ" +  
 "КЕНГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБЮqwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM";  
 *Random* random = new Random();  
  
 for (int i = 0; i < arr.length; i++) {  
 char c = alphabet.charAt(random.nextInt(alphabet.length()));  
 arr[i] = *String*.*valueOf*(c);  
 }  
  
 *System*.out.println(*Arrays*.*toString*(arr));  
  
 int counter = 0;  
  
 for (*String* s : arr) {  
 boolean letters = s.matches("[а**-**яa**-**z]");  
 if (letters) counter++;  
 }  
 *System*.out.printf("Кол-во строчных букв %d\n", counter);  
  
 int sym\_count = 0;  
  
 for (*String* s : arr) {  
 if (s.equals(symbol)) sym\_count++;  
 }  
 *System*.out.printf("Символов введеных с клавиатуры: %d\n", sym\_count);  
  
 if (symbol.matches("[0**-**9]+")) {  
 int digit = *Integer*.*parseInt*(symbol);  
 int digitCountInArray = 0;  
 for (*String* s : arr) {  
 if (s.matches("\\d") && *Integer*.*parseInt*(s) == digit) {  
 digitCountInArray++;  
 }  
 }  
 if (digitCountInArray >= 2) {  
 *System*.out.println("Среди символов есть 2 и более цифры, входящие в цифру " + digit);  
 }  
 } else {  
 *System*.out.println("Херня введенная с клавиатруры не число!");  
 }  
  
 int cout\_brackets = 0;  
  
 for (int i = 0; i < arr.length - 1; i++) {  
 if (arr[i].matches("\\(\\[\\{") && arr[i + 1].matches("\\)\\]\\}")) cout\_brackets++;  
 }  
 *System*.out.printf("Кол-во соседних закрытых скобок: %d\n", cout\_brackets);  
  
 int space\_counter = 0;  
  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i].equals(" ")) space\_counter++;  
 else space\_counter = 0;  
 if (space\_counter > 2) {  
 *System*.out.println("Обнаружены более двух подряд идущих пробелов.");  
 break;  
 }  
 }  
  
 boolean foundSequence = false;  
 for (int i = 0; i < arr.length - 3; i++) {  
 *// проверка условий для последовательности* if (*Character*.*isDigit*(arr[i].charAt(0)) &&  
 *Character*.*isDigit*(arr[i + 1].charAt(0)) &&  
 *Character*.*isDigit*(arr[i + 2].charAt(0))) {  
 int num1 = *Integer*.*parseInt*(arr[i]);  
 int num2 = *Integer*.*parseInt*(arr[i + 1]);  
 int num3 = *Integer*.*parseInt*(arr[i + 2]);  
 if (num1 > num2 && num2 > num3) { *// первые две цифры убывающие, последняя возрастающая* foundSequence = true;  
 break;  
 }  
 }  
 }  
 if (foundSequence) {  
 *System*.out.println("Существуют такие натуральные i и j, что i<k<j<п и " +  
 "si, и si+1 убывающая последовательность цифр, a sj и sj+1 возрастающая последовательность цифр.");  
 } else {  
 *System*.out.println("Не найдено таких натуральных i и j, что i<k<j<п и " +  
 "si, и si+1 убывающая последовательность цифр, a sj и sj+1 возрастающая последовательность цифр.");  
 }  
  
 }  
}

## 3.3. Тестирование работы программы

Таблица 3

Тестирование работы программы и проверка результатов решения

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

**Задача 4**

## 4.1. Постановка задачи

Задана матрица символов N×N. Определить построчно и по столбцам, есть ли в них выражения вида («число» + «число» \* («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.*\*;  
  
import static *java.lang.Math*.\*;  
  
public class *exc9\_4* {  
 public static int some\_line(*Character*[][] *array*, *ArrayList*<*Integer*> *lst*) {  
 int counter = 0;  
  
 for (int i = 0; i < *array*.length - 4; i++) {  
 for (int j = 0; j < *array*[i].length; j++) {  
 if (48 <= *array*[i][j] & *array*[i][j] <= 57 & 48 <= *array*[i + 1][j] & *array*[i + 1][j] <= 57  
 & 48 <= *array*[i + 2][j] & *array*[i + 2][j] <= 57 & 48 <= *array*[i + 3][j] & *array*[i + 3][j] <= 57  
 & 48 <= *array*[i + 4][j] & *array*[i + 4][j] <= 57) {  
 if (((int) (*array*[i][j] - 48) + (int) (*array*[i + 1][j] - 48) \*  
 ((int) (*array*[i + 2][j] - 48) - (int) (*array*[i + 3][j] - 48))) == (int) (*array*[i + 4][j] - 48)) {  
 *System*.out.println((int) (*array*[i][j] - 48) + " + " + (int) (*array*[i + 1][j] - 48) + " \* (" + (int) (*array*[i + 2][j] - 48) + " - " +  
 (int) (*array*[i + 3][j] - 48) + ") = " + ((int) *array*[i + 4][j] - 48) + " (Правильный столбик)");  
 counter += 1;  
 } else {  
 *System*.out.println((int) (*array*[i][j] - 48) + " + " + (int) (*array*[i + 1][j] - 48) + " \* (" + (int) (*array*[i + 2][j] - 48) + " - " +  
 (int) (*array*[i + 3][j] - 48) + ") = " + ((int) *array*[i + 4][j] - 48) + " (Неправильный столбик)");  
 *lst*.add((int) (*array*[i][j] - 48));  
 *lst*.add((int) (*array*[i + 1][j] - 48));  
 *lst*.add((int) (*array*[i + 2][j] - 48));  
 *lst*.add((int) (*array*[i + 3][j] - 48));  
 }  
 }  
 }  
 }  
 for (*Character*[] characters : *array*) {  
 for (int j = 0; j < characters.length - 4; j++) {  
 if (48 <= characters[j] & characters[j] <= 57 & 48 <= characters[j + 1] & characters[j + 1] <= 57  
 & 48 <= characters[j + 2] & characters[j + 2] <= 57 & 48 <= characters[j + 3] & characters[j + 3] <= 57  
 & 48 <= characters[j + 4] & characters[j + 4] <= 57) {  
 if ((int) (characters[j] - 48) + ((int) (characters[j + 1] - 48)  
 \* ((int) (characters[j + 2] - 48)) - (int) (characters[j + 3] - 48)) == (int) (characters[j + 4] - 48)) {  
 *System*.out.println((int) (characters[j] - 48) + " + " + ((int) (characters[j + 1] - 48) + " \* ("  
 + ((int) (characters[j + 2] - 48)) + " - " + (int) (characters[j + 3] - 48))  
 + ") =" + (int) (characters[j + 4] - 48) + " (Правильная строка)");  
 counter += 1;  
 } else {  
 *System*.out.println((int) (characters[j] - 48) + " + " + ((int) (characters[j + 1] - 48) + " \* ("  
 + ((int) (characters[j + 2] - 48)) + " - " + (int) (characters[j + 3] - 48))  
 + ") = " + (int) (characters[j + 4] - 48) + " (Неправильная строка)");  
 *lst*.add((int) (characters[j] - 48));  
 *lst*.add((int) (characters[j + 1] - 48));  
 *lst*.add((int) (characters[j + 2] - 48));  
 *lst*.add((int) (characters[j + 3] - 48));  
 }  
 }  
 }  
 }  
 return counter;  
 }  
  
 public static void main(*String*[] *args*) {  
 int n = 6 + (int) (*random*() \* 10);  
 *Character*[][] array = new Character[n][n];  
 *ArrayList*<*Integer*> list = new ArrayList<>();  
  
 for (int i = 0; i < array.length; i++) {  
 for (int j = 0; j < array[i].length; j++) {  
 array[i][j] = (char) (48 + (int) (*random*() \* (64 - 48)));  
 }  
 }  
 for (*Character*[] characters : array) {  
 for (*Character* character : characters) {  
 *System*.out.print(character + " ");  
 }  
 *System*.out.print("\n");  
 }  
 *System*.out.print("Количество правильных: " + *some\_line*(array, list) + "\nНеправильные:\n");  
  
 for (int i = 0; i < list.size() - 3; i += 4) {  
 *System*.out.println(list.get(i) + " + " + list.get(i + 1) + " \* (" + list.get(i + 2) + " - "  
 + list.get(i + 3) + ") = " +  
 (list.get(i) + list.get(i + 1) \* (list.get(i + 2) - list.get(i + 3))));  
 }  
 }  
}

## 4.3. Тестирование работы программы

Таблица 4

Тестирование работы программы и проверка результатов решения

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

**Задача 5**

## 5.1. Постановка задачи

## В заданном тексте найти произведение всех встречающихся цифр.

## 5.2. Решение задачи, код программы

import *java.util.Scanner*;  
  
public class *exc9\_5* {  
 public static void main(*String*[] *args*) {  
 *Scanner* input = new Scanner(*System*.in);  
 *System*.out.println("Введите текст который содержит цифры: ");  
 *String* text = input.nextLine();  
  
 int mult = 1;  
  
 for (int i = 0; i < text.length(); i++) {  
 char ch = text.charAt(i);  
 if (*Character*.*isDigit*(ch)) mult \*= ((int) (ch) - 48);  
 }  
  
 *System*.out.println(mult);  
  
 }  
}

## 5.3. Тестирование работы программы с проверкой

Таблица 5

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 | 7 \* 8 \* 9 = 56 \* 9 = 504 |

**Задача 7**

## 7.1. Постановка задачи

В заданной строке заменить каждый символ «!» числом, равным индексу этого символа в строке.

## 7.2. Решение задачи, код программы

import *java.util.Scanner*;  
  
public class *exc9\_7* {  
 public static void main(*String*[] *args*) {  
 *Scanner* input = new Scanner(*System*.in);  
 *System*.out.println("Текст с такой херней --> ! <-- : ");  
 *String* text = input.nextLine();  
  
 for (int i = 0; i < text.length(); i++) {  
 char ch = text.charAt(i);  
 if (ch == '!') {  
 text = text.substring(0, i) + *Integer*.*toString*(i) + text.substring(i + 1);  
 }  
 }  
 *System*.out.println(text);  
 }  
}

## 7.3. Тестирование работы программы с проверкой

Таблица 5

Тестирование работы программы и проверка результатов решения

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