Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и анализа данных

наименование института

ОТЧЕТ  
к лабораторной работе по дисциплине

Объектно-ориентированное программирование

Разработка консольного приложения 2

Наименование темы

Выполнил студент группы ИСТб 19–2 Первых А.А.

шифр Фамилия И.О.

Проверила Маланова Т.В.

Фамилия И.О.

Содержание отчета на 16 стр.

Иркутск 2020 г.

Содержание

[1 Постановка задачи 3](#_Toc53683192)

[2 Таблица спецификаций 4](#_Toc53683193)

[3 Разработка тестов 6](#_Toc53683194)

[4 Результаты тестирования 7](#_Toc53683195)

[5 Листинг исходного кода 10](#_Toc53683196)

[Заключение 15](#_Toc53683197)

[Список литературы 16](#_Toc53683198)

1 Постановка задачи

Разработать консольное приложение на языке java, реализующее решение задачи в соответствии с индивидуальным заданием. Исходные данные для задачи вводятся как параметры при запуске программы из командной строки, либо – как значения, инициализирующие переменные. Вывод результатов осуществляется в консоль.

Вариант 18.

a. Сгенерируйте числовой ряд длиной 30 элементов, представляющий

квадраты чисел натурального ряда. Сформируйте из полученных

значений два массива: один с четными значениями, другой с нечетными.

b. Подсчитайте, какие символы наиболее часто встречаются в предложении.

c. Выберите из множества числа, удовлетворяющие условию, введенному в виде строке при запуске программы (“>0” или “<10”).

2 Таблица спецификаций

|  |  |  |
| --- | --- | --- |
| **Имя** | **Назначение** | **Тип** |
| **Main** | | |
| Programm1 | Функция решение 1 программы | int |
| Programm2 | Функция решение 2 программы | string |
| Programm3 | Функция решение 3 программы | int |
| Constructor(); | Функция проверки работы конструкторов | string |
| **Program1** | | |
| Digit pr1 | Создание объекта pr1 класса Digit c параметром 30 (размер массива) | int |
| Print(pr1.naturalRyadx2(30)) | Метод создания натурального ряда размером 30 и вывод | int |
| Print(pr1.p1(30, 1)) | Метод вывода четных элементов массива pr1 | int |
| Print(pr1.p1(30, 2)) | Метод вывода нечетных элементов массива pr1 | int |
| **Program2** | | |
| Input input | Создание объекта input класса Input |  |
| str1 = input.inputVerPr2() | Создания строки str1 и ввод методом inputVerPr2 | string |
| Sentenc str | Создание объекта str равному str1 класса Sentenc | string |
| inp2Symb | Метод вывода элементов, встречающихся в строке | int |
| inp1Symb | Метод вывода элемента, встречающегося в строке | int |
| **Program3** | | |
| Input input | Создание объекта input класса Input |  |
| msize = input.inputVer | Ввод числа msize методом inputVer | int |
| s | Ввод строки s | string |
| Sentenc str | Создание объекта str равному s класса Sentenc | string |
| moreThan = str.moreThan | Метод определение знака больше или меньше в веденной строке | int |
| n = str.numStr | Метод определения числа в введенной строке | int |
| Digit digi | Создание объекта digi класса Digit c параметром msize (размер массива) | int |
| inpMass(msize) | Метод задания массива digi размером msize | int |
| printMass(moreThan, n, msize) | Метод вывода чисел массива digi соответствующих условиям moreThan, n | string |
| **Sentenc-класс для строк** | | |
| Поле string | Поле для класса | string |
| Sentenc() | Конструктор без параметра | string |
| Sentenc( String string) | Конструктор объекта Sentenc | string |
| Sentenc(Sentenc str) | Конструктор копирования Sentenc | string |
| upper() | Преобразование букв объекта Sentenc в большие | string |
| Size() | Возвращает размер объекта Sentenc | int |
| Print() | Вывод объекта Sentenc | string |
| chAt() | Метод определения кода ASCII символа строки | int |
| inp2Symb(a, b) | Вывод элементов строки | string |
| inp1Symb(a) | Вывод элемента строки | string |
| moreThan() | Определение знака больше или меньше в строке | int |
| numStr() | Определения числа в введенной строке | int |
| **Digit-класс для одномерного массива** | | |
| Поле digit | Поле для класса Одномерный массив | int |
| Digit (n) | Конструктор объекта Digit (массив размером n) | int |
| Digit (inp) | Конструктор объекта Digit с входной матрицей | int |
| Digit (Digit dig) | Конструктор копирования Digit | int |
| p1(m, k) | Решение программы 1 | int |
| sortEvOd( digit, evOrOd) | Сортировка массива на четный массив и нечетный массив (evOrOd 1 и 2 соотвественно) | int |
| sumEvOd( digit, evOrOd) | Количесвто четных и нечетных элементов массива | int |
| naturalRyadx2(m) | Создание натурального ряда | int |
| Print(source) | Вывод элементов матрицы с параметров | int |
| Print1() | Вывод элементов матрицы digit | int |
| Size() | Размер массива digit | int |
| inpMass(msize) | Ввод массива | int |
| printMass(moreThan, n, msize) | Вывод чисел соотвествующих условия | int |
| **Input - класс для ввода и проверки** | | |
| inputVerPr2() | Ввод и проверка строки для задачи 2 | int |
| inputVer() | Ввод и проверка целого числа | int |

3 Разработка тестов

Так как в задаче 1 при вводе числового массива используется генерация рандомных значений, то на время тестирования будет включен ручной ввод.

Таблица 2 – Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание | Входные данные | Выходные данные |
| **Задача 1** | | | |
| 1 | Штатный тест на срабатывание программы | source = { 32, 54, 39, 72, 84, 72, 34, 48, 24, 18, 18, 85, 89, 30, 4, 57, 31, 53, 81, 93, 18, 30, 65, 97, 14, 95, 33, 26, 69, 55} | Четные: [1024, 2916, 5184, 7056, 5184, 1156, 2304, 576, 324, 324, 900, 16, 324, 900, 196, 676]  Нечетные: [1521, 7225, 7921, 3249, 961, 2809, 6561, 8649, 4225, 9409, 9025, 1089, 4761, 3025] |
| **Задача 2** | | | |
| 1 | Штатный тест на срабатывание программы | str = “I am stud, on INRTU1.” | A встречается 1 раз(а)  D встречается 1 раз(а)  I встречается 2 раз(а)  M встречается 1 раз(а)  N встречается 2 раз(а)  O встречается 1 раз(а)  R встречается 1 раз(а)  S встречается 1 раз(а)  T встречается 2 раз(а)  U встречается 2 раз(а)  1 встречается 1 раз(а)  Пробел встречается 4 раз(а)  Запятая встречается 1 раз(а)  Точка встречается 1 раз(а) |
| 2 | Тест на проверку программы, при вводе не буквы, цифры, запятой, точки и пробела | str = “I am stud, on INRTU1.\” | Неправильный ввод, повторите ввод.(Буквы, цифры, запятая, точка и пробел.) |
| **Задача 3** | | | |
| 1 | Штатный тест на срабатывание программы (> 0) | msize = 4  s = > 0  arr = {-1 0 1 15} | Сообщение: “Множество чисел удовлетворяющее условию: 1 15 ” |
| 2 | Штатный тест на срабатывание программы (< 10) | msize = 5  s = < 10  arr = {-1 0 1 15} | Сообщение: “Множество чисел удовлетворяющие условию: -11 0 9” |
| 3 | Тест на проверку ввода неправильных значений | msize = qwe  s = > 0  arr = {qwe} | Сообщение: “ Это не число!” |
| 4 | Тест на проверку ввода неправильного значения условия | s = > 5 | Сообщение: “Неправильное условие, повторите ввод  (> 0 || < 10):” |

4 Результаты тестирования

В таблице 3 приведены результаты тестов из таблицы 2.

Таблица 2 – Таблица результатов тестирования.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входные данные | Выходные данные | Результат |
| **Задача 1** | | | |
| 1 | source = { 32, 54, 39, 72, 84, 72, 34, 48, 24, 18, 18, 85, 89, 30, 4, 57, 31, 53, 81, 93, 18, 30, 65, 97, 14, 95, 33, 26, 69, 55} | Четные: [1024, 2916, 5184, 7056, 5184, 1156, 2304, 576, 324, 324, 900, 16, 324, 900, 196, 676]  Нечетные: [1521, 7225, 7921, 3249, 961, 2809, 6561, 8649, 4225, 9409, 9025, 1089, 4761, 3025] | Тест пройден |
| **Задача 2** | | | |
| 1 | str = “I am stud, on INRTU1.” | A встречается 1 раз(а)  D встречается 1 раз(а)  I встречается 2 раз(а)  M встречается 1 раз(а)  N встречается 2 раз(а)  O встречается 1 раз(а)  R встречается 1 раз(а)  S встречается 1 раз(а)  T встречается 2 раз(а)  U встречается 2 раз(а)  1 встречается 1 раз(а)  Пробел встречается 4 раз(а)  Запятая встречается 1 раз(а)  Точка встречается 1 раз(а) | Тест пройден |
| 2 | str = “I am stud, on INRTU1.\” | Неправильный ввод, повторите ввод. (Буквы, цифры, запятая, точка и пробел.) | Тест пройден |
| **Задача 3** | | | |
| 1 | msize = 4  s = > 0  arr = {-1 0 1 15} | Сообщение: “Множество чисел удовлетворяющее условию: 1 15 ” | Тест пройден |
| 2 | msize = 5  s = < 10  arr = {-1 0 1 15} | Сообщение: “Множество чисел удовлетворяющие условию: -11 0 9” | Тест пройден |
| 3 | msize = qwe  s = > 0  arr = {qwe} | Сообщение: “Это не число!” | Тест пройден |
| 4 | s = > 5 | Сообщение: “Неправильное условие, повторите ввод  (> 0 || < 10):” | Тест пройден |

Результаты выполнения тестов с приведены на рисунках с 4 по 9 соответственно.

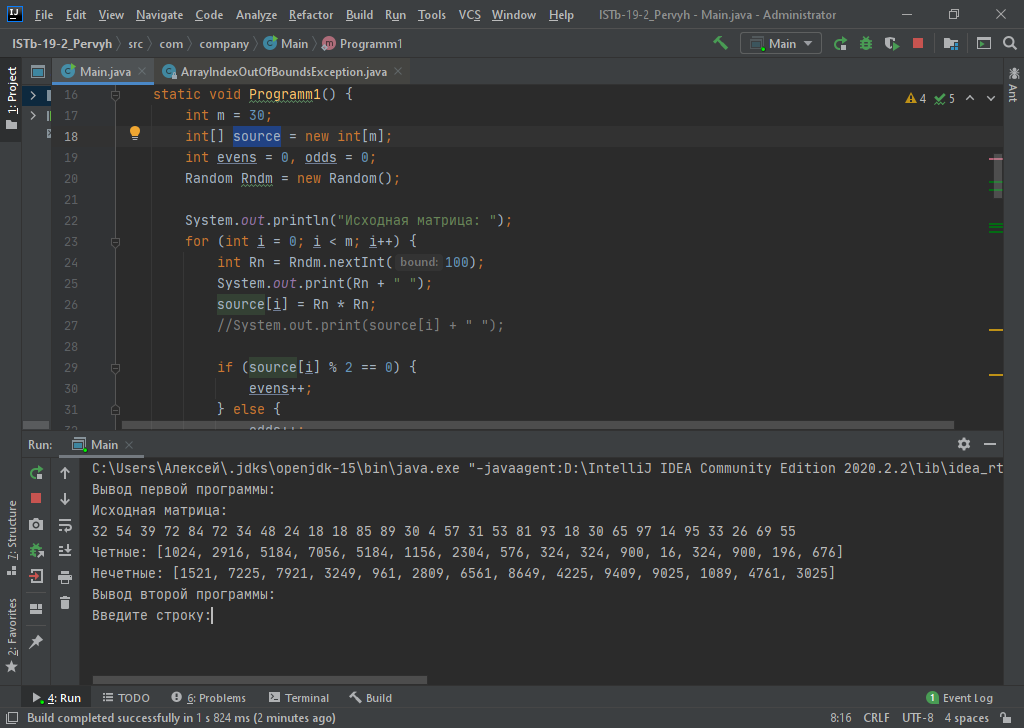


Рисунок 4 — Результат выполнения теста 1 задачи 1

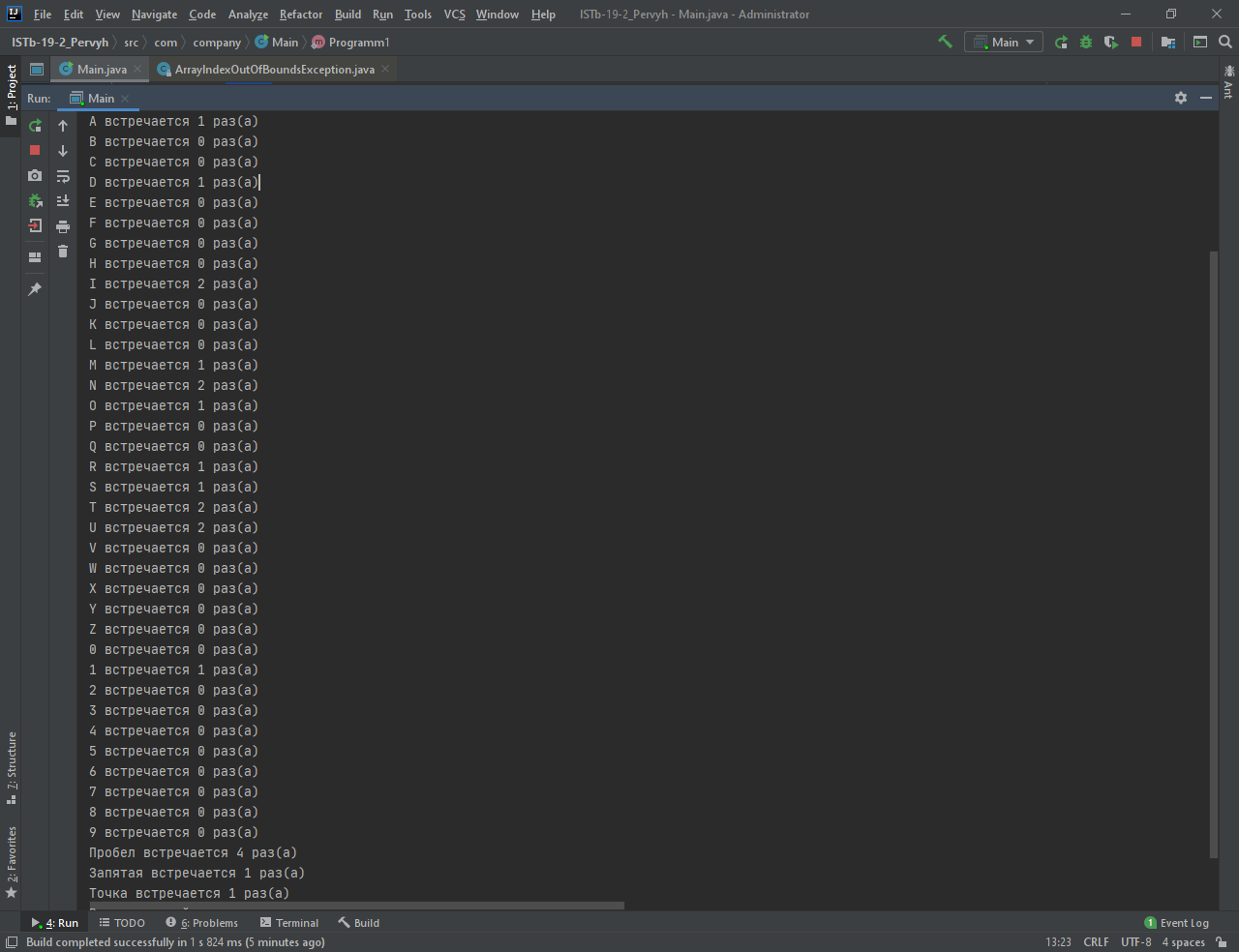


Рисунок 5 — Результат выполнения теста 1 задачи 2

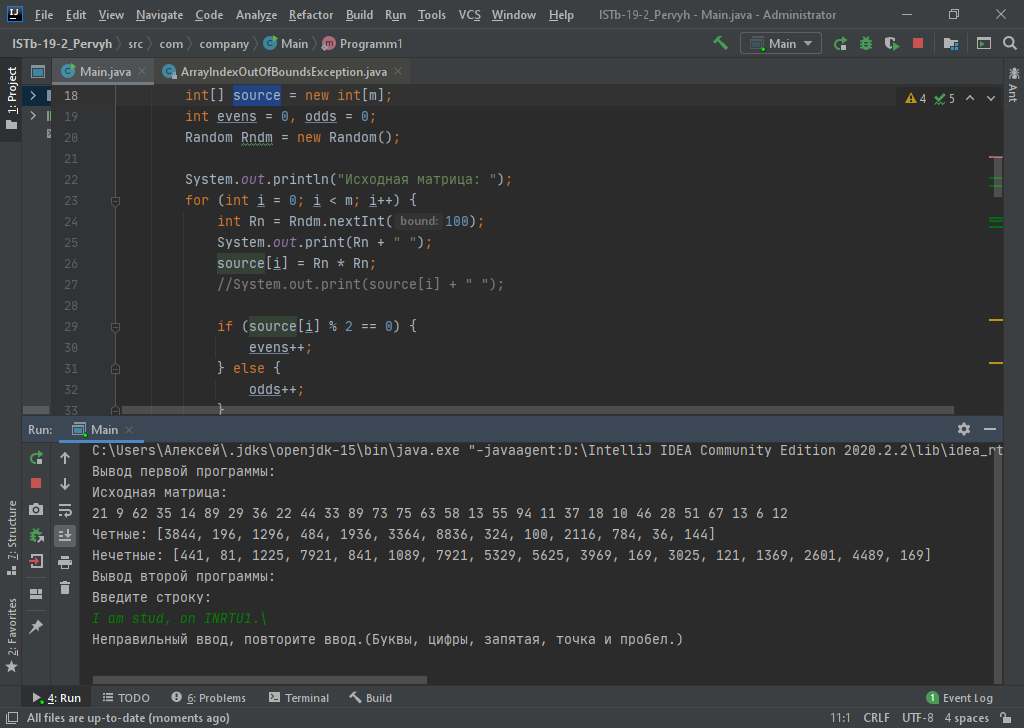


Рисунок 6 — Результат выполнения теста 2 задачи 2

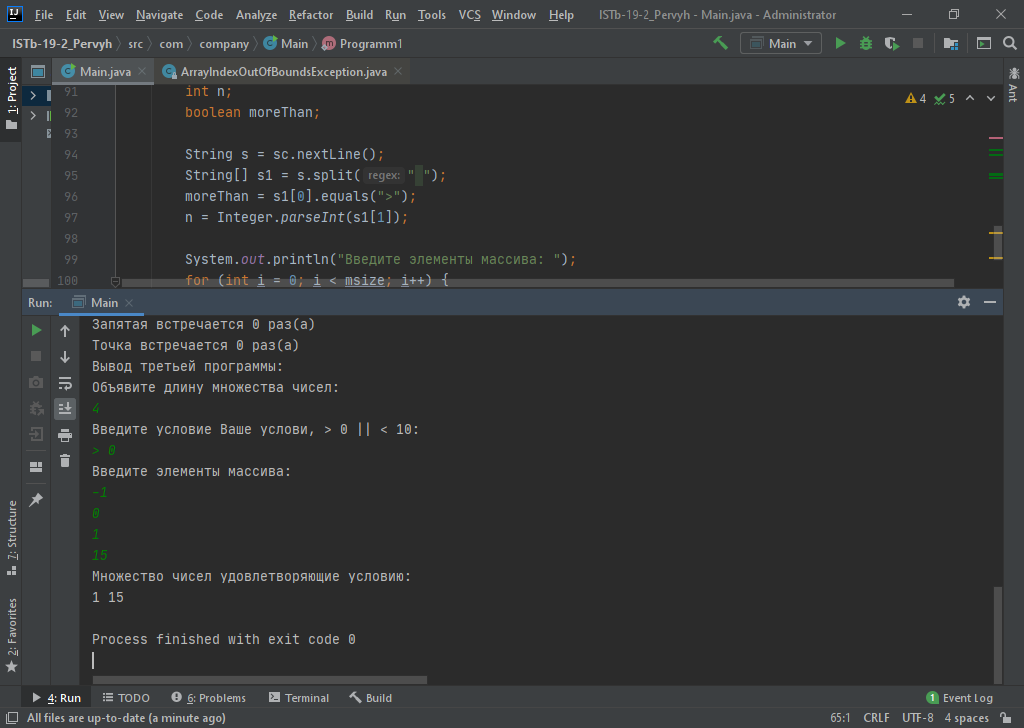


Рисунок 7 — Результат выполнения теста 1 задачи 3

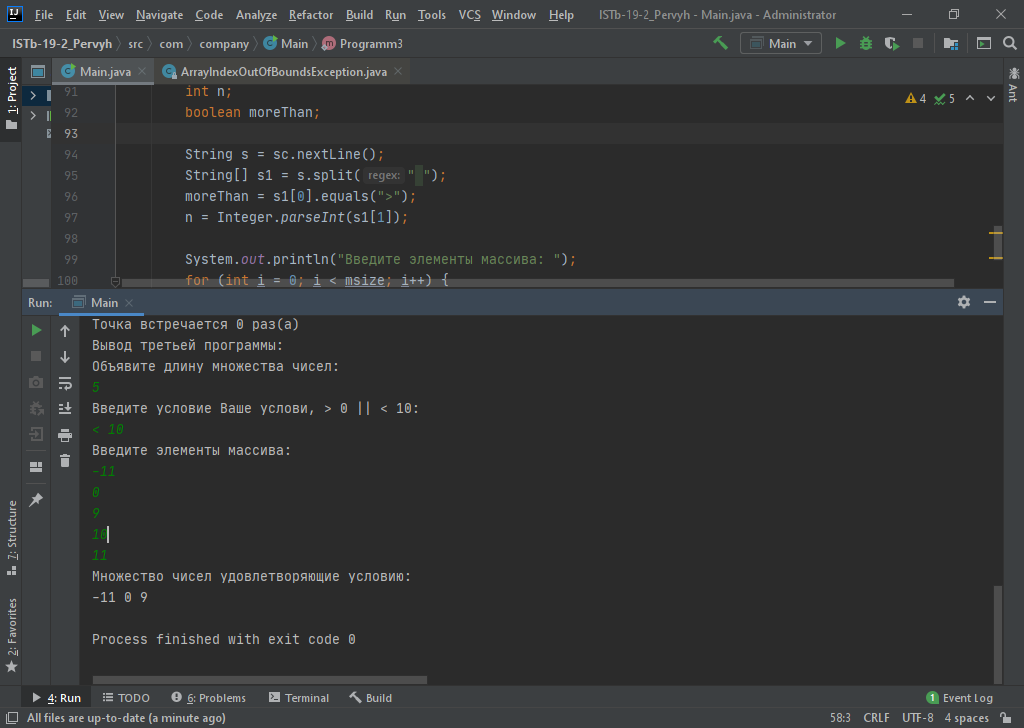


Рисунок 8 — Результат выполнения теста 2 задачи 3

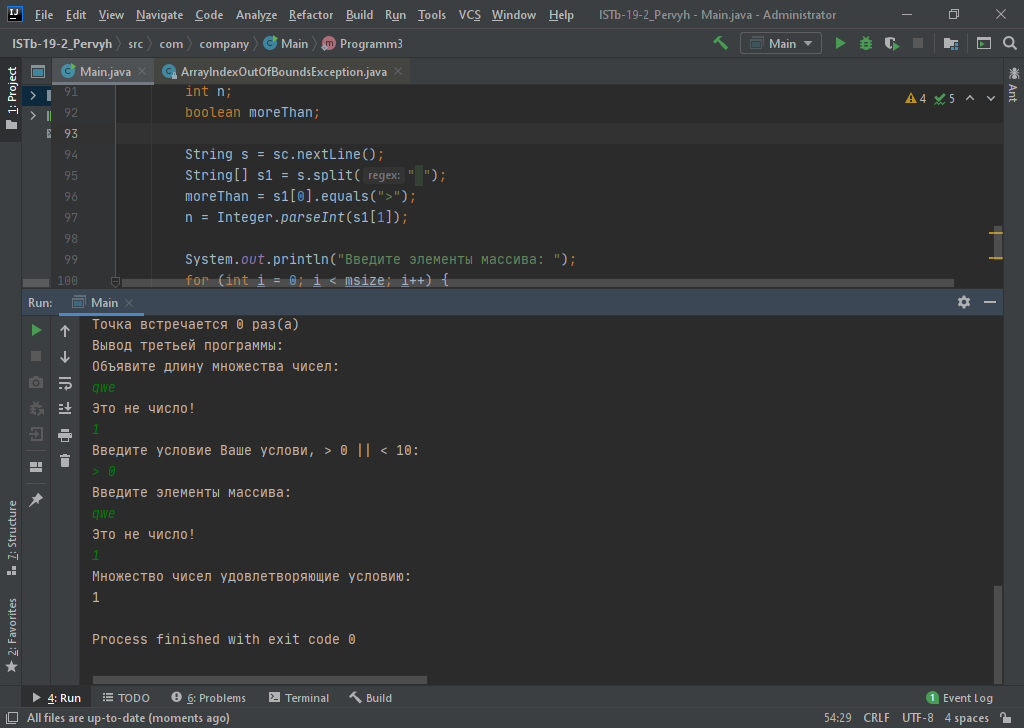


Рисунок 9 — Результат выполнения теста 3 задачи 3

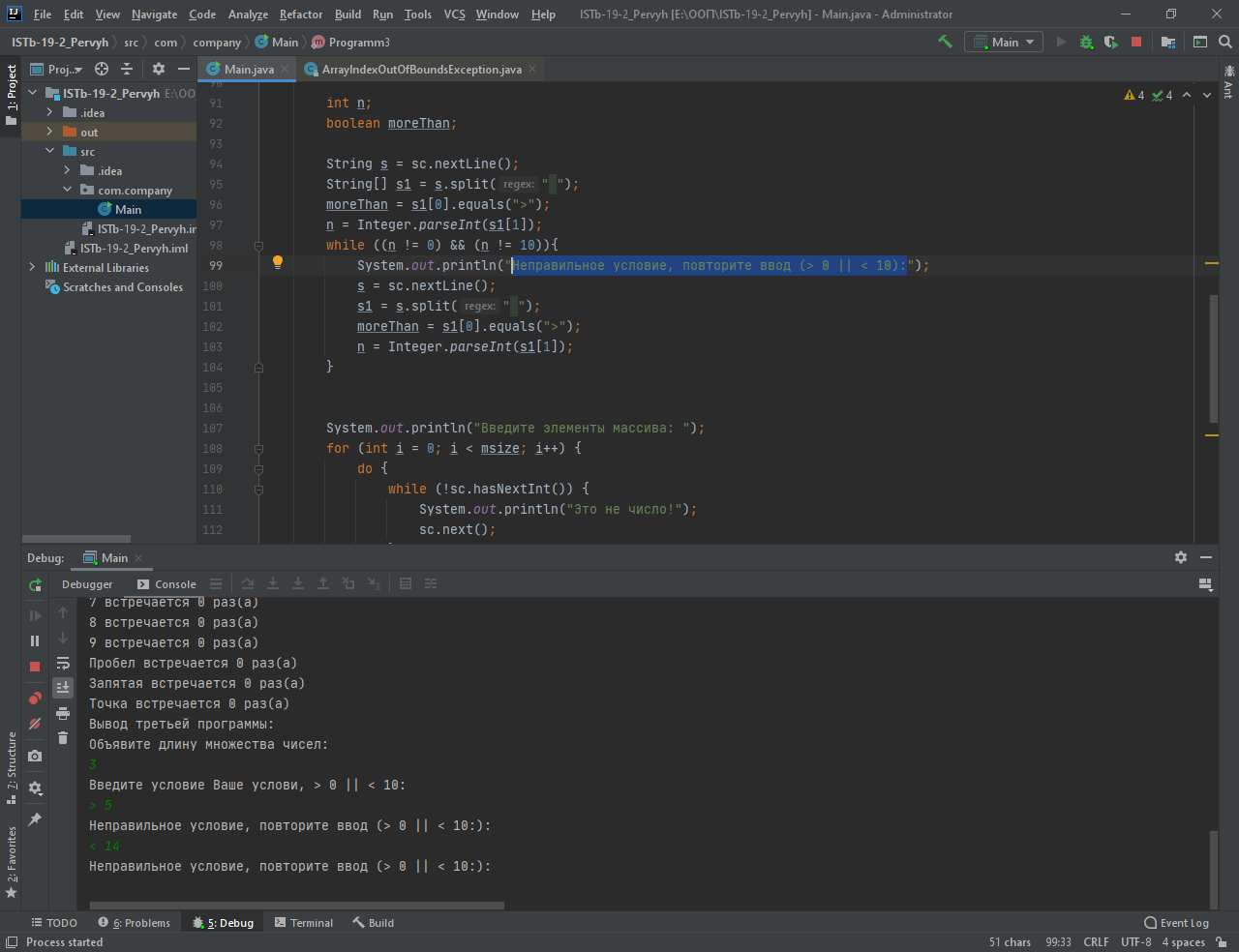


Рисунок 9 — Результат выполнения теста 4 задачи 3

5 Листинг исходного кода

**Класс Main**

public static void main(String[] args) {

System.out.println("Вывод первой программы: ");

Programm1();

System.out.println();

System.out.println("Вывод второй программы: ");

Programm2();

System.out.println();

System.out.println("Вывод третьей программы: ");

Programm3();

System.out.println();

Constructor();

}

static void Programm1() {

Digit pr1 = new Digit(30);

System.out.println("Данный ряд:");

pr1.Print(pr1.naturalRyadx2(30));// Создание натурального ряда

System.out.println();

System.out.println("Четные: ");

pr1.Print(pr1.p1(30, 1));// Вывод четных элементов массива pr1

System.out.println();

System.out.println("Нечетные: ");

pr1.Print(pr1.p1(30, 2));// Вывод нечетных элементов массива pr1

}

static void Programm2() {

Input input = new Input();

String str1 = input.inputVerPr2();//Ввожу строку

Sentenc str = new Sentenc(str1);// Объявляю объект str класса Sentenc

str.inp2Symb(65, 91);

str.inp2Symb(48, 58);

str.inp1Symb(32);

str.inp1Symb(44);

str.inp1Symb(46);

}

static void Programm3() {

Scanner sc = new Scanner(System.in);

int n, moreThan;

Input input = new Input();

System.out.println("Введите длину множества чисел: ");

int msize = input.inputVer();

System.out.println("Введите условие Ваше услови, > 0 || < 10: ");

String s = sc.nextLine();

Sentenc str = new Sentenc(s);

moreThan = str.moreThan();// Определение знака больше или меньше в веденной строке

n = str.numStr();// Определения числа в введенной строке

Digit digi = new Digit(msize);

digi.inpMass(msize);//Задаю массива

System.out.println("Множество чисел удовлетворяющие условию: ");

digi.printMass(moreThan, n, msize);

}

static void Constructor() {

Sentenc str = new Sentenc();

str.Print();

Sentenc str1 = new Sentenc("Конструтор с параметром");

str1.Print();

Sentenc str2 = new Sentenc(str1);//Конструктор копирования

str2.Print();

int msize = 5;

Digit digi = new Digit(msize);

Digit digi1 = new Digit(new int[] {84, 72, 5, 34, 2, 2, 43});

digi1.Print1();

System.out.println();

Digit digi2 = new Digit(digi1);

digi2.Print1();

}

**Класс Sentenc**

private String string;

public Sentenc() {

this.string = "I can, program in java. And write 12345678";

}

public Sentenc(String string) {

this.string = string;

}

public Sentenc(Sentenc str){

this.string = str.string;

}

// Преобразование букв в большие

public String upper(){

this.string = string.toUpperCase();

return string;

}

// Размер строки

public int Size(){

int Size;

Size = this.string.length();

return Size;

}

// Вывод обьекта класса Sentenc

public void Print(){

System.out.println(this.string);

}

// Методы задачи 2

// Определния кода ASCII символа строки

public int[] chAt(){

int[] arr = new int[255];

for (int i = 0; i < string.length(); i++) {

arr[this.string.charAt(i)]++;

}

return arr;

}

// Вывод элементов массива в интервале

public void inp2Symb(int a, int b){

int[] arr = new int[255];

Sentenc str = new Sentenc(this.string);

str.upper();

arr = str.chAt();

for (int i = a; i < b; i++) {

if (arr[i] != 0) {

System.out.println((char) i + " встречается " + arr[i] + " раз(а)");

}

}

}

// Вывод 1 элемента массива

public void inp1Symb(int a){

int[] arr = new int[255];

Sentenc str = new Sentenc(this.string);

str.upper();

arr = str.chAt();

System.out.println("Пробел встречается " + arr[a] + " раз(а)");

}

// Задача 3

//Определение знака больше или меньше в веденной строке

public int moreThan(){

int moreThan;

String[] s1 = string.split(" ");

if (s1[0].equals(">")){moreThan = 1;}

else {moreThan = 2;}

return moreThan;

}

//Определения числа в введенной строке

public int numStr(){

int n;

String[] s1 = string.split(" ");

n = Integer.parseInt(s1[1]);

return n;

}

**Класс Digit**

private int[] digit;

public Digit(int n) {

this.digit = new int[n];

}

public Digit(int[] inp) {

this.digit = new int[Size(inp)];

// копируем элементы, поэлементно

for (int i = 0; i < Size(inp); ++i)

{

this.digit[i] = inp[i];

}

}

public Digit(Digit dig) {

this.digit = dig.digit;

}

// решение программы 1

public int[] p1(int m, int k){

int[] nat = naturalRyadx2(m);

int[] masEven = new int[sumEvOd(nat, 1)];

int[] masOdd = new int[sumEvOd(nat, 2)];

masEven = sortEvOd(nat, 1);

masOdd = sortEvOd(nat, 2);

if (k == 1){return masEven;}

else {return masOdd;}

}

//сортировка массива на четный массив и нечетный массив (evOrOd 1 и 2 соотвественно)

public int[] sortEvOd(int[] source, int evOrOd){

int[] masEven = new int[sumEvOd(source, 1)];

int[] masOdd = new int[sumEvOd(source, 2)];

int index1 = 0, index2 = 0;

for (int i = 0; i < Size(source); i++) {

if (source[i] % 2 == 0) {

masEven[index1++] = source[i];

} else {

masOdd[index2++] = source[i];

}

}

if (evOrOd == 1){return masEven;}

else {return masOdd;}

}

//Количесвто четных и нечетных элементов массива

public int sumEvOd(int[] source, int evOrOd){

int evens = 0, odd = 0;

for (int i = 0; i < Size(source); i++) {

if (source[i] % 2 == 0){evens++;}

else {odd++;}

}

if (evOrOd == 1){return evens;}

else {return odd;}

}

// Создание натурального ряда

public int[] naturalRyadx2(int m){

int[] source = new int[m];

int in = 1;

for (int i = 0; i < m; i++) {

source[i] = in \* in;

in++;

}

return source;

}

// Вывод массива

public void Print(int[] source){

for (int i = 0; i < Size(source); i++){

System.out.print(source[i] + ", ");

}

}

public void Print1(){

for (int i = 0; i < Size(this.digit); i++){

System.out.print(this.digit[i] + ", ");

}

}

// Размер массива

public int Size(int[] source){

return source.length;

}

// Ввод массива

public void inpMass(int msize){

Digit digi = new Digit(msize);

System.out.println("Введите элементы массива: ");

for (int i = 0; i < msize; i++) {

System.out.println("Элемент " + (i+1) + " массива: ");

Input input = new Input();

this.digit[i] = input.inputVer();

}

}

// Вывод чисел соотвествующих условия

public void printMass(int moreThan, int n, int msize){

for (int i = 0; i < msize; i++) {

if ((moreThan == 1) && this.digit[i] > n)

System.out.print(this.digit[i]+ " ");

else if ((moreThan == 2) && this.digit[i] < n)

System.out.print(this.digit[i] + " ");

}

}

**Класс Input**

// Ввод и проверка строки для задачи 2

public static String inputVerPr2(){

Scanner sc = new Scanner(System.in);

String str;

System.out.println("Введите строку: (Буквы, цифры, запятая, точка и пробел.)");

str = sc.nextLine();

while (!str.matches("[A-Za-z0-9,. ]+")){

System.out.println("Неправильный ввод, повторите ввод.(Буквы, цифры, запятая, точка и пробел.)");

str = sc.nextLine();

}

return str;

}

// Ввод целого числа

public static int inputVer(){

Scanner sc = new Scanner(System.in);

int scan;

do {

while (!sc.hasNextInt()) {

System.out.println("Это не число!");

sc.next();

}

scan = sc.nextInt();

} while ((scan % 1) != 0);// Если эквивалентно целому числу.

return scan;

}

Заключение

В процессе выполнения лабораторной работы 1: “Разработка консольного приложения” были разработаны алгоритмы решения индивидуального задания. Было сформировано 7 тестов, покрывающих все ситуации. Составленные на языке Java программы, реализующие разработанные алгоритмы, прошла все тесты успешно.

В ходе лабораторной работы были получены практические навыки использования конструкций языка Java, а также знакомство с платформой.

Все цели практики достигнуты.

Список литературы

1. Java. Экспресс-курс [электронный ресурс] // Сайт Александра Климова [сайт], URL: http://developer.alexanderklimov.ru/android/java/java.php

2. API Specification for the Java 7 SE. [официальный сайт] URL: http://docs.oracle.com/javase/7/docs/api/

3. The Java Tutorials. SE [электронный ресурс], URL: http://docs.oracle.com/javase/tutorial/index.html

4. Гради Буч, Роберт А. Максимчук, Майкл У. Энгл, Бобби Дж. Янг, Джим Коналлен, Келли А. Хьюстон. Объектно-ориентированны анализ и проек-тирование с примерами приложений. Третье издание. М.: "Вильямс", 2010.

5. Хабибуллин И.Ш. Java 7: для программистов / И. Ш. Хабибуллин. – Санкт-Петербург : БХВ–Петербург, 2014.