### 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

Центр программной инженерии

ОТЧЕТ

к лабораторной работе №1 по дисциплине:

|  |
| --- |
| «Объектно-ориентированное программирование» |
| Разработка консольного приложения |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: | ИСТб 19-2 |  |  |  | Козлов М.В. |
|  | шифр группы |  | подпись |  | Фамилия ИО |
| Проверил: | доцент |  |  |  | Маланова Т.В. |
|  | должность |  | подпись |  | Фамилия ИО |

Иркутск. 2020 г.

Оглавление

[1 Постановка задачи 4](#_Toc51357312)

[2 Блок-схема алгоритма 5](#_Toc51357313)

[3 Таблица спецификации 17](#_Toc51357314)

[4 Таблица тестов 19](#_Toc51357315)

[5 Результаты тестирования 21](#_Toc51357316)

[6 Исходный код 25](#_Toc51357317)

[7 Список использованных источников 32](#_Toc51357318)

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 № 9.

a. Сгенерируйте числовой ряд длиной не более 50, представляющий собой числа Фибоначчи (каждый последующий элемент представляют суммы двух предыдущих, первые два элемента равны 1). Выведите на печать первые 10, отмечая четные числа каким-нибудь символом.

b. Дан массив чисел, знак операции и операнд. Выведите на печать массив, полученный в результате арифметического действия, совершенного над элементами массива.

c. Дана строка содержащая ряд чисел в двоичном виде, необходимо проверить упорядочены ли они.

2 Блок-схема алгоритма

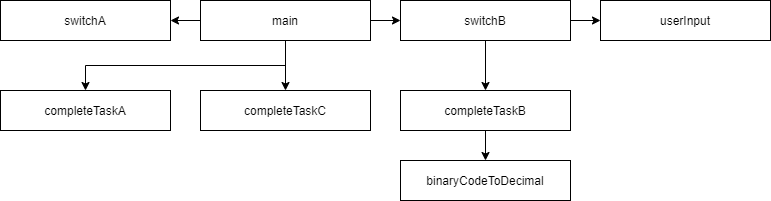


Рисунок 1 – Иерархия методов

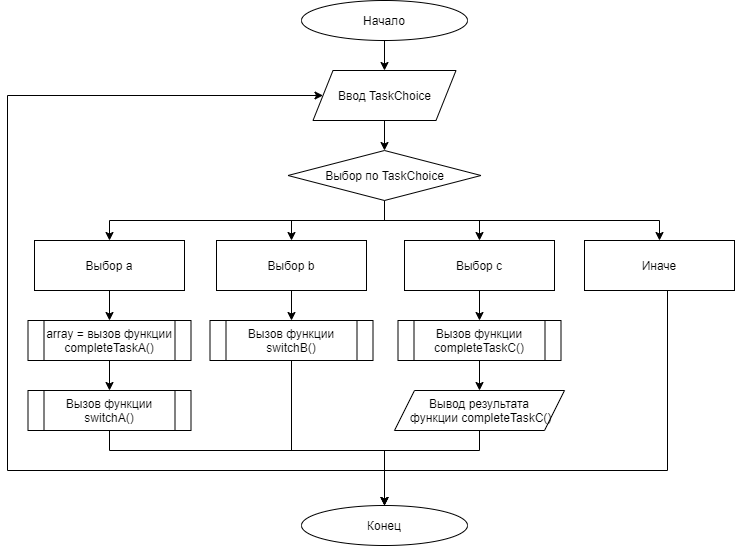


Рисунок 1.1 – Блок-схема главного метода main

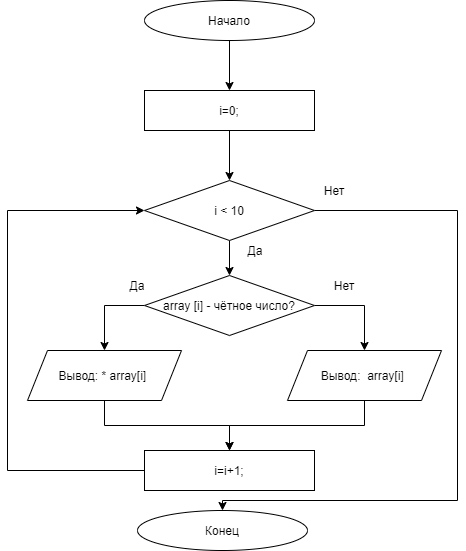


Рисунок 1.2 – Блок-схема вспомогательного метода switchA

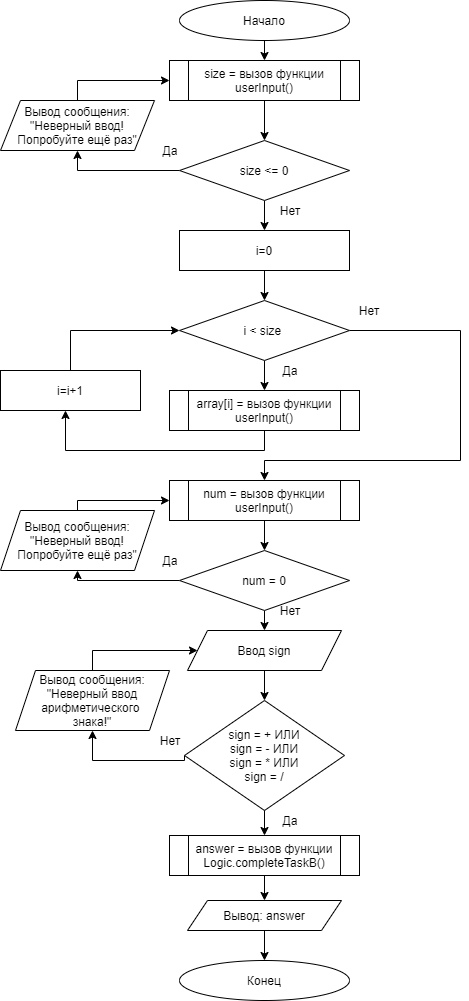


Рисунок 1.3 – Блок-схема вспомогательного метода switch

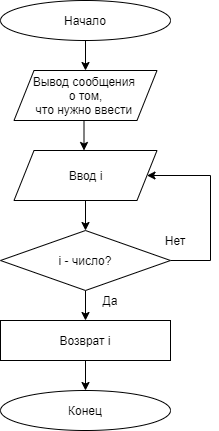


Рисунок 1.4 – Блок-схема вспомогательного метода userInput

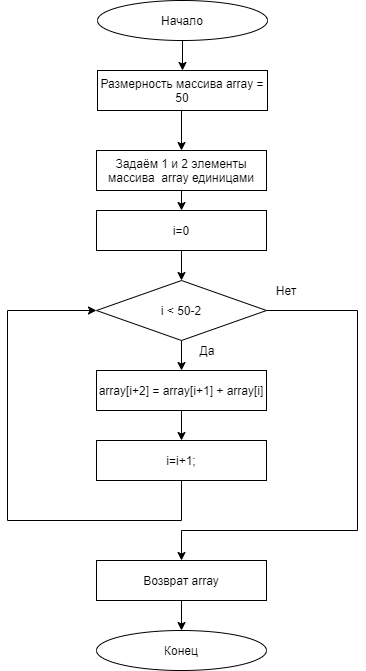


Рисунок 1.5 – Блок-схема метода completeTaskA

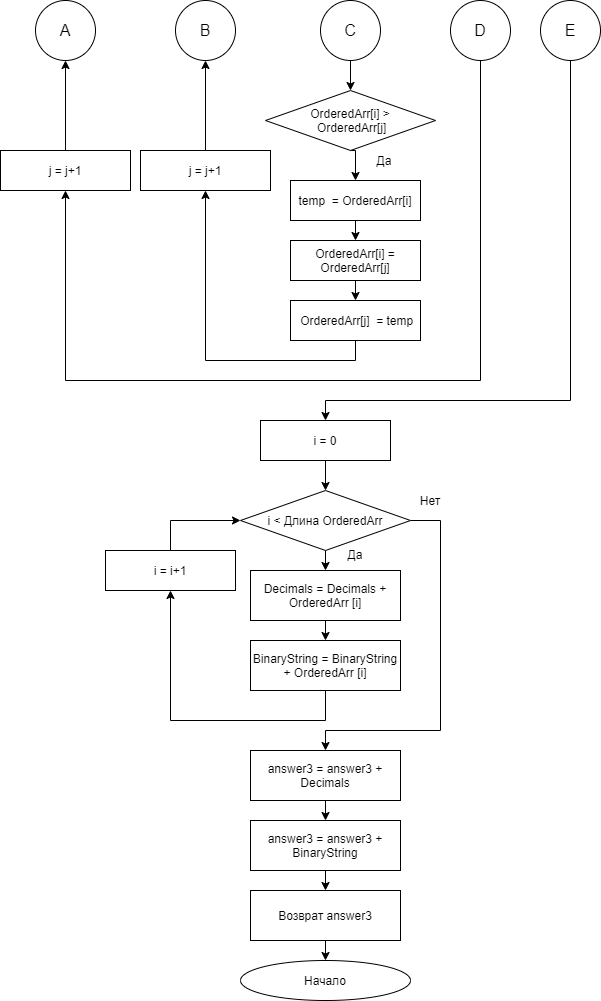
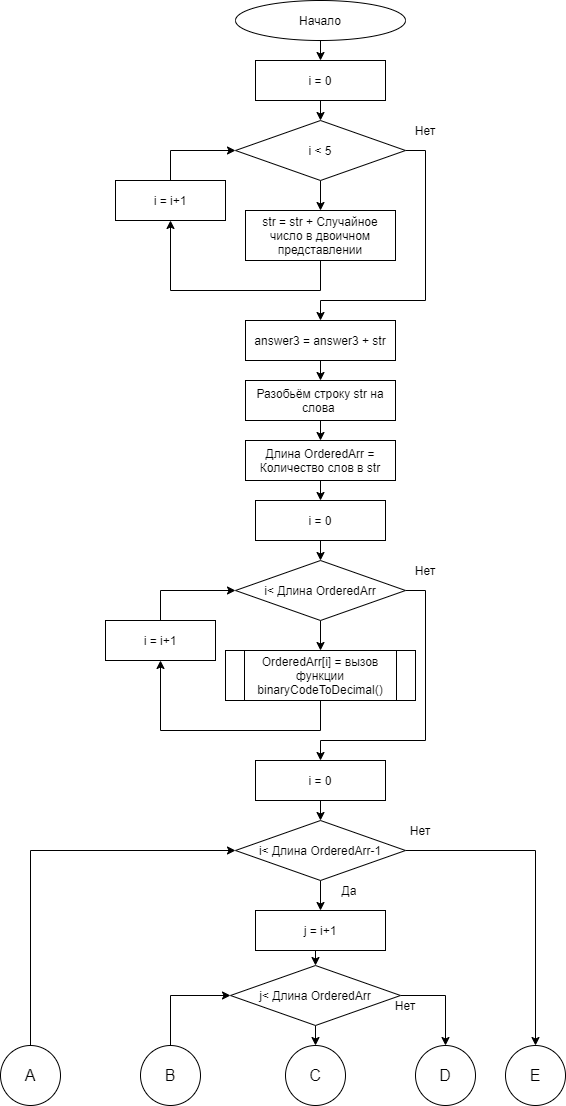


Рисунок 1.6 – Блок-схема метода completeTaskC

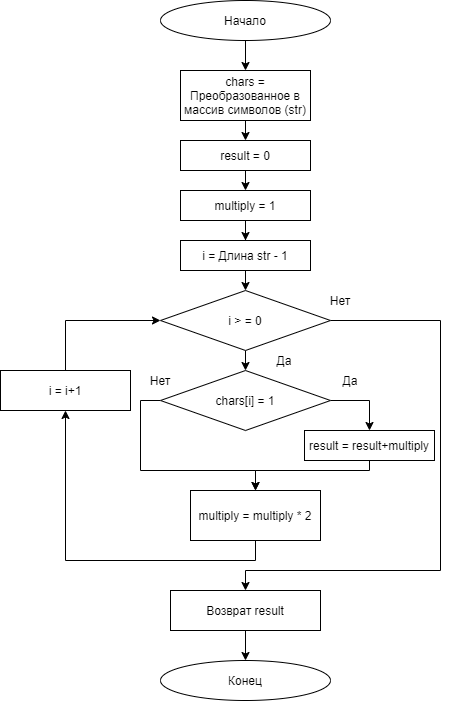


Рисунок 1.7 – Блок-схема вспомогательного метода binaryCodeToDecimal

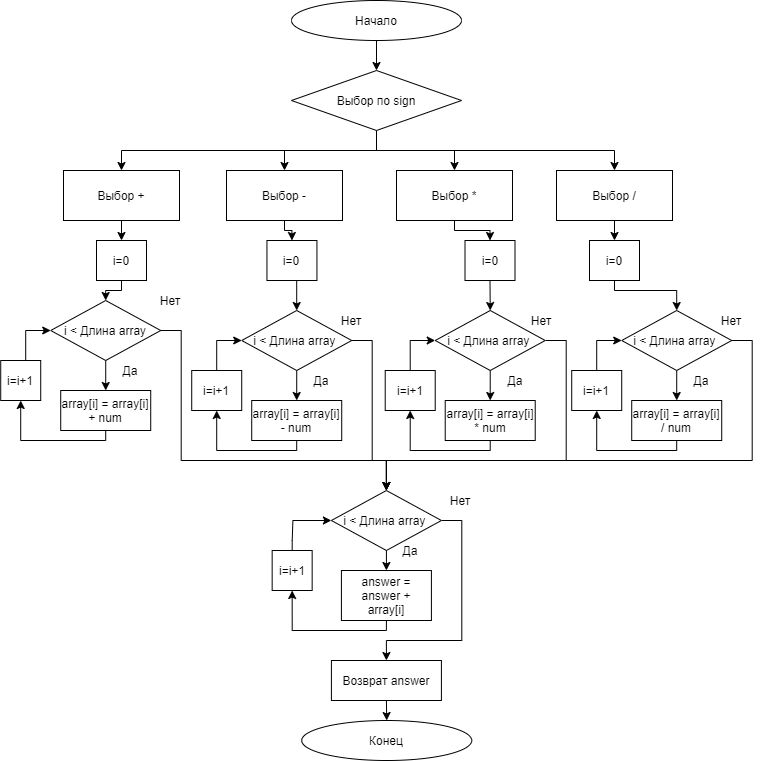


Рисунок 1.8 – Блок-схема вспомогательного метода completeTaskB

3 Таблица спецификации

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменной | Описание переменных | Тип данных |
| **public class L1V9** | | | |
| **int main() – точка входа в программу** | | | |
| 1 | TaskChoice | Входные данные-переменная для выбора задачи для решения. | int |
| 2 | array | Выходные данные-ответ на задание a | int[] |
| **public static void switchB** | | | |
| 1 | size | Результат функции **userInput ()** - переменная, обозначающая размерность массива. | int |
| 2 | num | Результат функции **userInput ()** - переменная, обозначающая операнд, с которым проводятся арифметические действия. | int |
| 3 | sign | Арифметический знак | String |
| 4 | array | Массив чисел, с значениями элементов которого проводятся арифметические действия. Значения элементов массива - результат функции **userInput ().** | int[] |
| 5 | answer | Ответ на задание b | String |
| **public static int userInput ()** | | | |
| 1 | message | Входные данные - сообщение для пользователя, уведомляющая пользователя о том, что нужно ввести. | String |
| 2 | i | Выходные данные - переменная, к которой присваивается входная строка, которая проверяется , является ли числом или нет. | int |
| **public static void switchA** | | | |
| 1 | array | Входные данные – ответ на задание а. | int[] |
|  | | | |
| **public static class Logic** | | | |
| **public static int[] completeTaskA()** | | | |
| 1 | array | Выходные данные – массив чисел Фибоначи | int[] |
| **public static String completeTaskB()** | | | |
| 1 | sign | Входные данные - переменная, обозначающая выбор пользователем арифметического действия для решения задачи. | string |
| 2 | num | Входные данные - переменная, обозначающая операнд, с которым проводятся арифметические действия. | int |
| 3 | size | Входные данные - переменная, обозначающая размерность массива. | int |
| 4 | array | Входные данные - массив чисел, с значениями элементов которого проводятся арифметические действия. | int[] |
| 5 | answer | Выходные данные – ответ на задание b | String |
|  |  | **public static String completeTaskC()** |  |
| 1 | str | Строка из случайных чисел в двоичном представлении | String |
| 2 | StrToArr | Преобразованное в массив строк str | String[] |
| 3 | OrderedArr | Упорядоченный список десятичных чисел. | int[] |
| 4 | Decimals | Строка из упорядоченных десятичных чисел. | String |
| 5 | BinaryString | Строка из упорядоченных двоичных чисел. | String |
| 6 | answer | Выходные данные – ответ на задание с. |  |
| **public static int binaryCodeToDecimal()** | | | |
| 1 | str | Входные данные – двоичное число | String |
| 2 | chars | Преобразованное в массив символов str | char[] |
| 3 | multiply | Вспомогательная переменная для перевода из двоичного числа в десятичное. | int |
| 4 | result | Выходные данные – десятичное число | int |

4 Таблица тестов

**main()**

|  |  |  |
| --- | --- | --- |
| № | Входные данные | Данные на выходе |
| 1 | a | Числовой ряд Фибоначи:  1 1 \*2 3 5 \*8 13 21 \*34 55 |
| 2 | b | Введите размерность массива: |
| 3 | c | Сгенерированная строка двоичных чисел:  1001001001,1111101,111111110,101100101,1011001101  Упорядоченная строка двоичных чисел в десятичном представлении : 125,357,510,585,717  Упорядоченная строка двоичных чисел:  1111101,101100101,111111110,1001001001,1011001101 |
| 4 | f | Введите корректные данные |
| 5 | 1 | Введите корректные данные |

**switchB()**

|  |  |  |
| --- | --- | --- |
| № | Входные данные | Данные на выходе |
| 6 | size = 5  array =5,5,5,5,5  num = 5  sign = + | Вывод результата арифметического действия { 10,10,10,10,10,} |
| 7 | size = 5  array =5,5,5,5,5  num = 5  sign = - | Вывод результата арифметического действия { 0,0,0,0,0,} |
| 8 | size = 5  array =5,5,5,5,5  num = 5  sign = \* | Вывод результата арифметического действия { 25,25,25,25,25,} |
| 9 | size = 5  array =5,5,5,5,5  num = 5  sign = / | Вывод результата арифметического действия { 1,1,1,1,1,} |
| 10 | size = 5  array =5,5,5,5,5  num = 5  sign = u | Некорректный ввод арифметического знака! |
| 11 | size = u | Неверный ввод! |
| 12 | size = 5  array =u | Неверный ввод! |
| 13 | size = 5  array =5,5,5,5,5  num = u | Неверный ввод! |
| 14 | size = -1 | Неверный ввод! Попробуйте ещё раз |
| 15 | size = 0 | Неверный ввод! Попробуйте ещё раз |
| 16 | size = 5  array =5,5,5,5,5  num = 0 | Неверный ввод! Попробуйте ещё раз |

5 Результаты тестирования

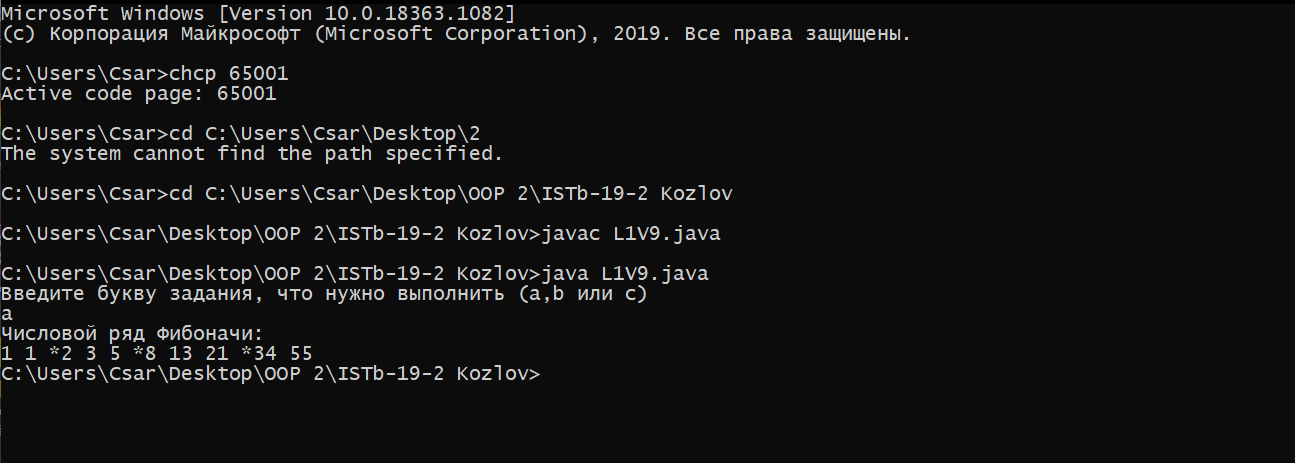


Рисунок 3.1 – Тест 1

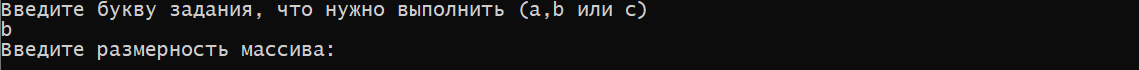


Рисунок 3.2 – Тест 2

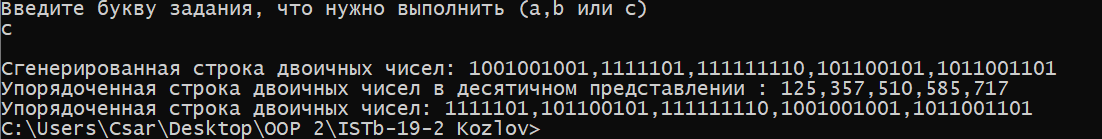


Рисунок 3.3 – Тест 3

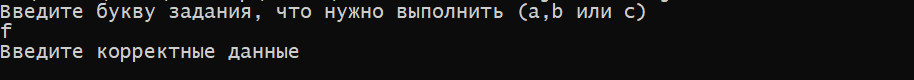


Рисунок 3.4 – Тест 4

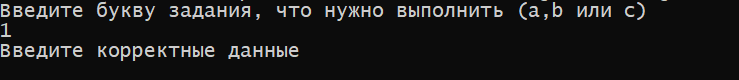


Рисунок 3.5 – Тест 5

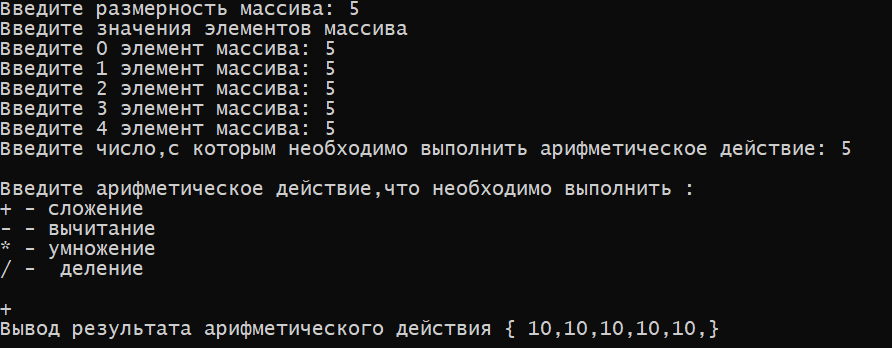


Рисунок 3.6 – Тест 6

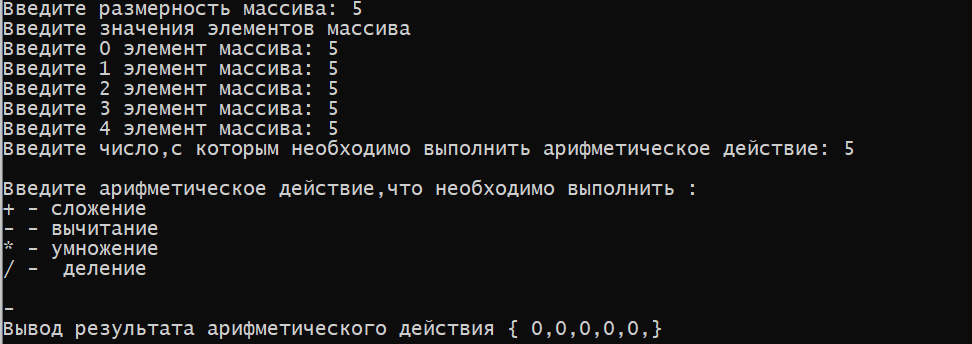


Рисунок 3.7 – Тест 7

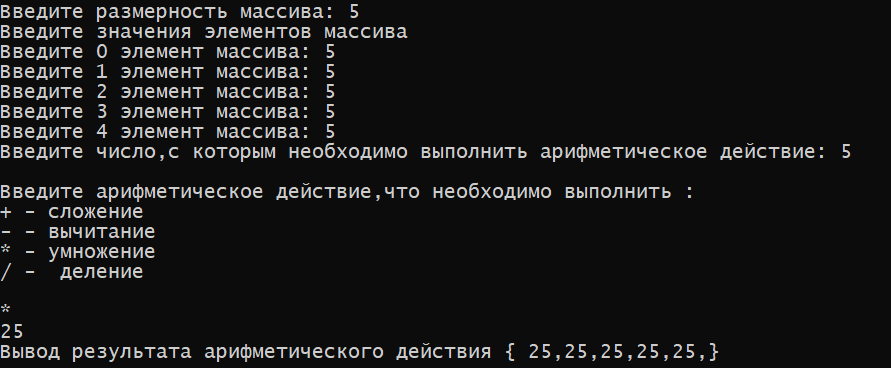


Рисунок 3.8 – Тест 8

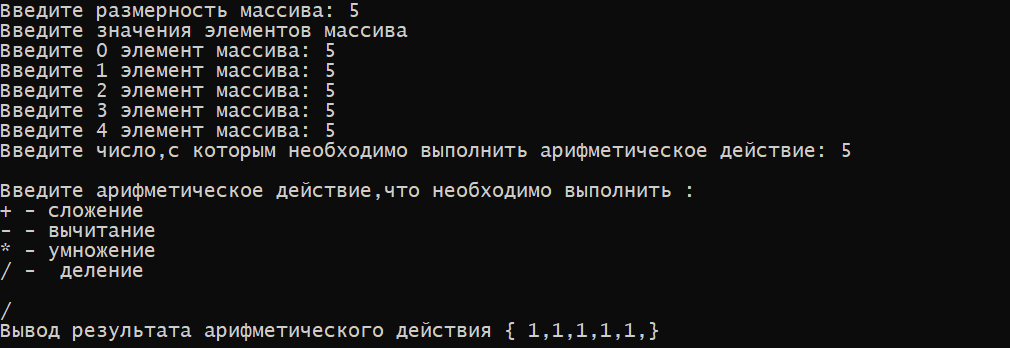


Рисунок 3.9 – Тест 9

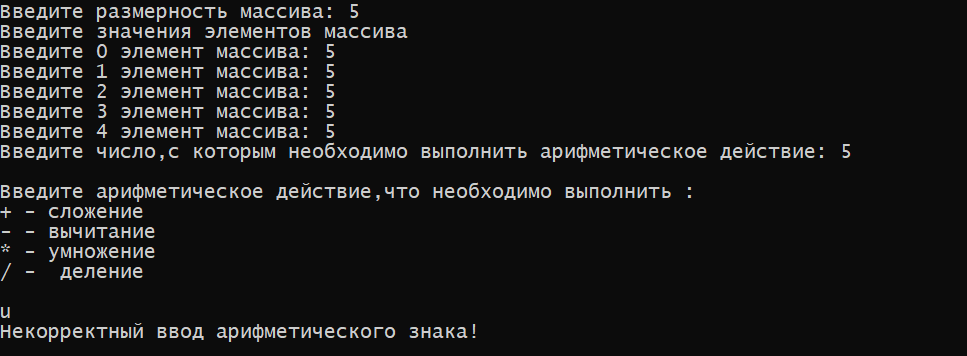


Рисунок 4.0 – Тест 10

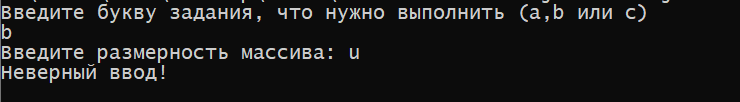


Рисунок 4.1 – Тест 11

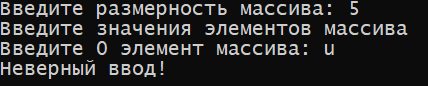


Рисунок 4.2 – Тест 12

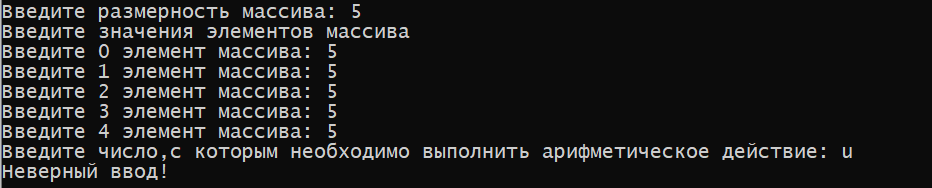


Рисунок 4.3 – Тест 13



Рисунок 4.4 – Тест 14



Рисунок 4.5 – Тест 15

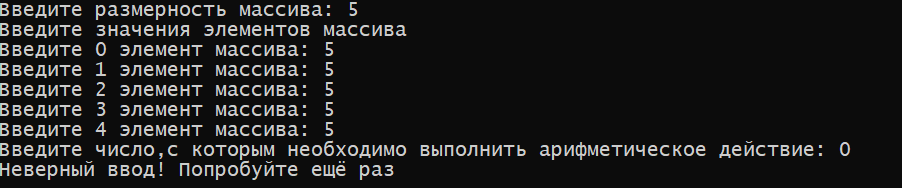


Рисунок 4.6 – Тест 16

6 Исходный код

package com.company;  
  
import java.util.Random;  
import java.util.Scanner;  
  
public class Main {  
  
public static void switchB() {  
 Scanner input = new Scanner(System.*in*);  
 int size = *userInput*("Введите размерность массива: ");  
 while (size <= 0) {  
 size = *userInput*("Неверный ввод! Попробуйте ещё раз\n");  
 }  
  
 int array[] = new int[size];  
 System.*out*.println("Введите значения элементов массива");  
 for (int i = 0; i < size; i++) {  
 array[i] = *userInput*("Введите " + (i) + " элемент массива: ");  
 }  
  
 int num = *userInput*("Введите число,с которым необходимо выполнить арифметическое действие: ");  
  
 while (num == 0) {  
 num = *userInput*("Неверный ввод! Попробуйте ещё раз\n");  
 }  
  
 System.*out*.println("\nВведите арифметическое действие,что необходимо выполнить : " + "\n" +  
 "+ - сложение" + "\n" +  
 "- - вычитание" + "\n" +  
 "\* - умножение" + "\n" +  
 "/ - деление" + "\n");  
 String sign;  
  
 sign = input.nextLine();  
  
 while (!sign.equals("+") && !sign.equals("-") && !sign.equals("\*") && !sign.equals("/")) {  
 System.*out*.println("Неверный ввод арифметического знака!");  
 sign = input.nextLine();  
 }  
  
 String answer = Logic.*completeTaskB*(sign, array, num);  
  
 System.*out*.print("Вывод результата арифметического действия ");  
  
 System.*out*.println("{" + " " + answer + "}");  
}  
  
public static void switchA(int []array)  
{  
 int i = 0;  
  
 System.*out*.print("Числовой ряд Фибоначи: \n");  
 while (i < 10) {  
 if (array[i] % 2 == 0) {  
 System.*out*.print("\*" + array[i] + " ");  
 } else {  
 System.*out*.print(array[i] + " ");  
 }  
 i++;  
 }  
}  
  
 public static int userInput(String message) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.print(message);  
 int i;  
 while (true) {  
 try {  
 i = Integer.*parseInt*(sc.next());  
 break;  
 } catch (NumberFormatException e) {  
 System.*out*.println("Неверный ввод!");  
 }  
 }  
 return i;  
 }  
  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.println("Введите букву задания, что нужно выполнить (a,b или c)");  
 String TaskChoice;  
  
 TaskChoice = input.nextLine();  
  
 while (!TaskChoice.equals("a") && !TaskChoice.equals("b") && !TaskChoice.equals("c")) {  
 System.*out*.println("Введите корректные данные");  
 TaskChoice = input.nextLine();  
 }  
  
 switch (TaskChoice) {  
 case "a": {  
 int[] array=Logic.*completeTaskA*();  
  
 L1V9.*switchA*(array);  
 break;  
 }  
 case "b": {  
 L1V9.*switchB*();  
 break;  
 }  
 case "c": {  
 System.*out*.print(Logic.*completeTaskC*());  
 break;  
 }  
 }  
  
 }  
   
 public static class Logic {  
 public static int[] completeTaskA() {  
 int[] array = new int[50];  
  
 array[0] = array[1] = 1;  
  
 int i = 0;  
  
 while (i < 50 - 2) {  
 array[i + 2] = array[i + 1] + array[i];  
  
 i++;  
 }  
  
 return array;  
 }  
  
 public static String completeTaskB(String sign, int[] array, int num) {  
  
 String answer = "";  
 switch (sign) {  
 case "+":  
 for (int i = 0; i < array.length; i++) {  
 array[i] = array[i] + num;  
 }  
 break;  
 case "-":  
 for (int i = 0; i < array.length; i++) {  
 array[i] = array[i] - num;  
 }  
 break;  
 case "\*":  
 for (int i = 0; i < array.length; i++) {  
 array[i] = array[i] \* num;  
 }  
 break;  
 case "/":  
 for (int i = 0; i < array.length; i++) {  
 array[i] = array[i] / num;  
 }  
 break;  
 }  
 for (int i = 0; i < array.length; i++) {  
  
 answer += array[i] + ",";  
 }  
 return answer;  
  
 }  
  
 private static int binaryCodeToDecimal(String str) {  
 char[] chars = str.toCharArray();  
 int result = 0;  
 int multiply = 1;  
 for (int i = str.length() - 1; i >= 0; i--) {  
 if (chars[i] == '1') {  
 result += multiply;  
 }  
 multiply \*= 2;  
 }  
 return result;  
 }  
  
 public static String completeTaskC() {  
  
 String answer3 ="";  
 String str = "";  
  
 for (int i = 0; i < 5; i++) {  
 if (i != 4) {  
 str += Integer.*toBinaryString*(1 + (int) (Math.*random*() \* 1000)) + ",";  
 } else {  
 str += Integer.*toBinaryString*(1 + (int) (Math.*random*() \* 1000));  
 }  
 }  
 answer3+="\n" + "Сгенерированная строка двоичных чисел: " + str;  
  
  
 String[] StrToArr = str.split(",");  
  
 int[] OrderedArr = new int[StrToArr.length];  
  
 for (int i = 0; i < OrderedArr.length; i++) {  
 OrderedArr[i] = *binaryCodeToDecimal*(StrToArr[i]);  
 }  
  
 for (int i = 0; i < StrToArr.length - 1; i++) {  
 for (int j = i + 1; j < OrderedArr.length; j++) {  
 if (OrderedArr[i] > OrderedArr[j]) {  
 int temp = OrderedArr[i];  
 OrderedArr[i] = OrderedArr[j];  
 OrderedArr[j] = temp;  
 }  
 }  
 }  
 String BinaryString = "";  
 String Decimals = "";  
 for (int i = 0; i < OrderedArr.length; i++) {  
 if (i != OrderedArr.length - 1) {  
 Decimals += OrderedArr[i] + ",";  
 BinaryString += Integer.*toBinaryString*(OrderedArr[i]) + ",";  
 } else {  
 Decimals += OrderedArr[i];  
 BinaryString += Integer.*toBinaryString*(OrderedArr[i]);  
 }  
 }  
 answer3+="\n" + "Упорядоченная строка двоичных чисел в десятичном представлении : " + Decimals;  
 answer3 +="\n" + "Упорядоченная строка двоичных чисел: " + BinaryString;  
 return answer3;  
 }  
 }  
}

7 Список использованных источников

1. Объектно-ориентированное программирование. Метод. указания по

выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во

ИРНИТУ, 2017. – 24 c.