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** |
| по лабораторной работе №1 по дисциплине «Объектно-ориентированное программирование»  «Разработка классов и их использование» |
| наименование темы  Вариант №9 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-19-1 |  |  |  | К.У. Ботурхонов |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | Т.В. Маланова |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Работа защищена с оценкой | | | |  | | |

Иркутск 2020 г.

**Содержание**

[1 Постановка задачи 3](#_Toc51240978)

[2 Описание структуры классов 4](#_Toc51240979)

[3 Таблица спецификация переменных 5](#_Toc51240980)

[4 Таблица тестов 6](#_Toc51240981)

[5 Результаты тестирования 7](#_Toc51240982)

[6 Исходный код 8](#_Toc51240983)

[Список использованных источников 12](#_Toc51240984)

1 Постановка задачи

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.

Провести объектную декомпозицию задачи из индивидуального задания  
и разработать класс, содержащий соответствующие поля для хранения необходимых данных и методы, обеспечивающие достаточную для решения задачи  
функциональность класса. Создание объекта класса решающего задание и вызов его методов должны осуществляться из главного класса. Ввод и вывод данных осуществляется как в предыдущей лабораторной работе, для передачи исходных данных экземпляру класса, решающему задачу, должны быть разработаны соответствующие методы.

2 Описание структуры классов

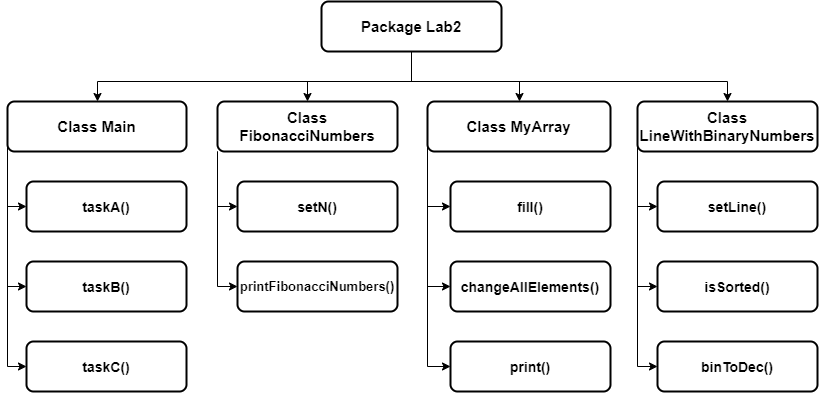
****

Рисунок 1 - Структура классов

3 Таблица спецификация переменных

Таблица 1 – Спецификация локальных переменных

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Назначение | Тип | Диапазон |
| N | Размерность массива | Integer | [1, 1000] |
| size | Размерность массива | Integer | [1, 1000] |
| operationSign | знак операции | Char | ASCII |
| operand | Операнд | Double | [1, 1000] |
| line | Строка | String | - |

Таблица 2 – Спецификация методов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Назначение | Модификатор доступа | Аргументы | Возвращаемый тип |
| **Класс Main** | | | | |
| taskA | Выполнение задания A | static | - | void |
| taskB | Выполнение задания B | static | - | void |
| taskC | Выполнение задания C | static | - | void |
| **Класс FibonacciNumbers** | | | | |
| setN | Установка размера массива | public | int n | void |
| printFibonacciNumbers | Вывод чисел Фибоначчи | public | - | void |
| FibonacciNumbers | Конструктор | public | int n | - |
| **Класс MyArray** | | | | |
| setSize | Установка размера массива | public | int size | void |
| fill | Заполнение массива | public | - | void |
| changeAllElements | Выполнение операций над массивом | public | char operationSign, double operand | void |
| print | Вывод массива | public | - | void |
| **Класс LineWithBinaryNumbers** | | | | |
| setLine | Установление значения line | public | String line | void |
| Продолжение таблицы 2 | | | | |
| Название | Назначение | Модификатор доступа | Аргументы | Возвращаемый тип |
| isSorted | Проверка сортирована ли строка | public | - | boolean |
| binToDec | Перевод чисел из 2-го СС на 10ый СС | public | String binaryNumber | int |

4 Таблица тестов

Таблица 3 - Тесты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Задание | Вход | Выход | Примечания |
| 1 | a | - | 1 1 2% 3 5 8% 13 21 34% 55 | Первые 10 чисел Фибоначчи |
| 2 | b | {5 8 4 6 9}  ch = ‘+’  operand = 5 | 10.0 13.0 9.0 11.0 14.0 | Элементы массива увеличены на 5 |
| 3 | с | “10 100 11 1000” | “Ряд не упорядочен” | Второе число больше |
| 4 | b | {88 46 9 11 5 10}  ch = ‘/’  operand = 2 | 44.0 23.0 4.5 5.5 2.5 5.0 | Элементы массива разделены на 2 |
| 5 | с | “1 10 11 100 101 110 111” | “Ряд упорядочен” | - |
| 6 | b | {4 9 6 0}  ch = ‘\*’  operand = 3 | 12.0 27.0 18.0 0.0 | Элементы массива умножены на 3 |
| 7 | c | “10 100 1000 1001” | “Ряд упорядочен” |  |

5 Результаты тестирования

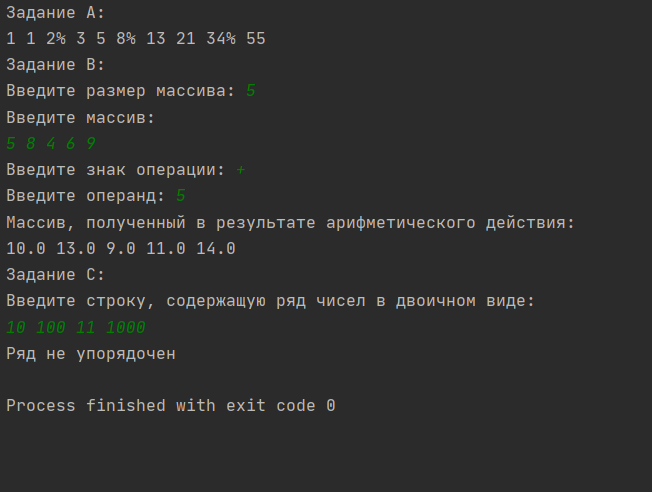


Рисунок 2 – Результат тестов 1

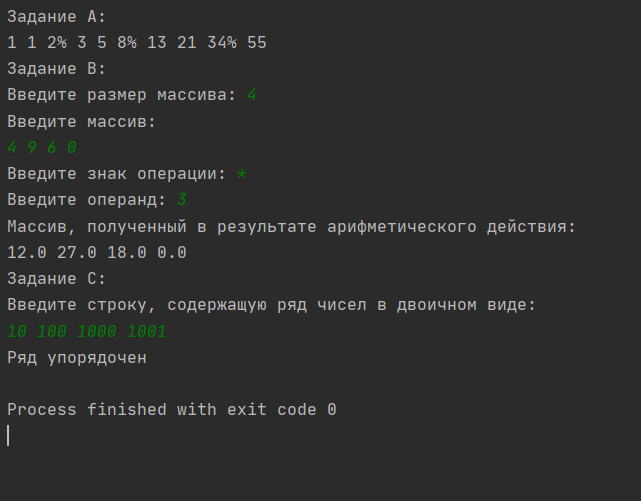


Рисунок 3 – Результат тестов 2

6 Исходный код

Исходный код класса Main

import java.util.Scanner;  
public class Main {  
  
 static Scanner *in* = new Scanner(System.*in*);  
  
 public static void main(String[] args) {  
 System.*out*.println("Задание A: ");  
 *taskA*();  
 System.*out*.println("Задание B: ");  
 *taskB*();  
 System.*out*.println("Задание C: ");  
 *taskC*();  
 }  
  
 static void taskA() {  
 final int N = 10;  
 FibonacciNumbers fibonacciNumbers = new FibonacciNumbers(N);  
 fibonacciNumbers.printFibonacciNumbers();  
 }  
  
 static void taskB() {  
 *//-----------------------------Ввод массива-------------------------------* int size;  
 System.*out*.print("Введите размер массива: ");  
 size = *in*.nextInt();  
 System.*out*.println("Введите массив: ");  
 MyArray myArray = new MyArray(size);  
 myArray.fill();  
 *//--------------------------Операция с массивом---------------------------* System.*out*.print("Введите знак операции: ");  
 String operationSign = *in*.next();  
 System.*out*.print("Введите операнд: ");  
 double operand = *in*.nextInt();  
 myArray.changeAllElements(operationSign.charAt(0), operand);  
 *//----------------------------Вывод массива-------------------------------* System.*out*.println("Массив, полученный в результате арифметического действия: ");  
 myArray.print();  
  
 }  
  
 static void taskC() {  
 System.*out*.println("Введите строку, содержащую ряд чисел в двоичном виде: ");  
 *in*.nextLine();  
 String line = *in*.nextLine();  
  
 LineWithBinaryNumbers lineWithBinaryNumbers = new LineWithBinaryNumbers(line);  
  
 if (lineWithBinaryNumbers.isSorted()) {  
 System.*out*.println("Ряд упорядочен");  
 } else {  
 System.*out*.println("Ряд не упорядочен");  
 }  
 }  
}

Исходный код класса FibonacciNumbers

public class FibonacciNumbers implements Cloneable{  
  
 private int N;  
  
 public void setN(int n) {  
 if (n > 0) {  
 N = n;  
 } else {  
 System.*err*.println("Ошибка: N < 0");  
 }  
 }  
  
 public FibonacciNumbers() {  
 setN(10);  
 }  
  
 public FibonacciNumbers(int n) {  
 setN(n);  
 }  
  
 public void printFibonacciNumbers () {  
 int[] fibonacciNumbers = new int[N];  
 fibonacciNumbers[0] = 1;  
 fibonacciNumbers[1] = 1;  
 for (int i = 2; i < N; i++) {  
 fibonacciNumbers[i] = fibonacciNumbers[i-2] + fibonacciNumbers[i-1];  
 }  
 for (int i = 0; i < N; i++) {  
 System.*out*.print(fibonacciNumbers[i]);  
 if (fibonacciNumbers[i] % 2 == 0) {  
 char ch = '%';  
 System.*out*.print(ch);  
 }  
 System.*out*.print(" ");  
 }  
 System.*out*.println();  
 }  
  
 @Override  
 public Object clone() throws CloneNotSupportedException {  
 return (FibonacciNumbers) super.clone();  
 }  
}

Исходный код класса MyArray

import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
public class MyArray implements Cloneable{  
 private ArrayList<Double> array = new ArrayList<>();  
 private final Scanner in = new Scanner(System.*in*);  
 private int size;  
 private final int DEFAULT\_SIZE = 10;  
  
 public int getSize() {  
 return size;  
 }  
  
 public void setSize(int size) {  
 if (size > 0) {  
 this.size = size;  
 } else {  
 System.*err*.println("Размер массива должен быть строго положителным!");  
 }  
 }  
  
 public MyArray() {  
 setSize(DEFAULT\_SIZE);  
 }  
  
 public MyArray(int size) {  
 setSize(size);  
 }  
  
 public void fill() {  
 array.clear();  
 for (int i = 0; i < size; i++) {  
 array.add(in.nextDouble());  
 }  
 }  
  
 public void changeAllElements(final char operationSign, final double operand) {  
 switch (operationSign) {  
 case '+' -> {  
 for (int i = 0; i < array.size(); i++) {  
 array.set(i, array.get(i) + operand);  
 }  
 }  
 case '-' -> {  
 for (int i = 0; i < array.size(); i++) {  
 array.set(i, array.get(i) - operand);  
 }  
 }  
 case '\*' -> {  
 for (int i = 0; i < array.size(); i++) {  
 array.set(i, array.get(i) \* operand);  
 }  
 }  
 case '/' -> {  
 for (int i = 0; i < array.size(); i++) {  
 array.set(i, array.get(i) / operand);  
 }  
 }  
 }  
 }  
  
 public void print() {  
 for (Double elem: array) {  
 System.*out*.print(elem + " ");  
 }  
 System.*out*.println();  
 }  
  
 @Override  
 public Object clone() throws CloneNotSupportedException {  
 MyArray clone = (MyArray) super.clone();  
 clone.array = (ArrayList<Double>) array.clone();  
 return clone;  
 }  
}

Исходный код класса LineWithBinaryNumbers

import java.util.Scanner;  
  
public class LineWithBinaryNumbers implements Cloneable{  
  
 private Scanner in = new Scanner(System.*in*);  
 private String line;  
 private String[] binaryNumbers;  
  
 public void setLine(String line) {  
 this.line = line;  
 binaryNumbers = line.split(" ");  
 }  
  
 public LineWithBinaryNumbers(String line) {  
 setLine(line);  
 }  
  
 public boolean isSorted () {  
  
 boolean sorted = true;  
  
 for (int i = 0; i < binaryNumbers.length - 1; i++) {  
 if (binToDec(binaryNumbers[i]) > binToDec(binaryNumbers[i+1])) {  
 sorted = false;  
 break;  
 }  
 }  
 return sorted;  
 }  
  
 private int binToDec (String binaryNumber) {  
 int n = 0;  
 try {  
 n = Integer.*parseInt*(binaryNumber, 2);  
 } catch (NumberFormatException error) {  
 System.*err*.println("Ошибка: Неверный ввод данных");  
 System.*exit*(0);  
 }  
 return n;  
 }  
  
 @Override  
 public Object clone() throws CloneNotSupportedException {  
 LineWithBinaryNumbers clone = (LineWithBinaryNumbers) super.clone();  
  
 clone.binaryNumbers = (String[]) this.binaryNumbers.clone();  
  
 return clone;  
 }  
}

Список использованных источников

1. API Specification for the Java 7 SE. [официальный сайт]

URL: <http://docs.oracle.com/javase/7/docs/api/> (дата обращения:05.09.2020)

1. The Java Tutorials. SE [электронный ресурс],

URL:<http://docs.oracle.com/javase/tutorial/index.html>

(дата обращения:05.09.2020)