Министерство науки и образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

Высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий и анализа данных

РАЗРАБОТКА КОНСОЛЬНОГО ПРИЛОЖЕНИЯ

Отчёт по лабораторной работе №1

Вариант 21

По дисциплине: «Объектно-ориентированное программирование»

Выполнил студент

группы АСУб-19-1 \_\_\_\_\_\_\_\_\_\_\_ С.Э. Казаков

Принял

Доцент \_\_\_\_\_\_\_\_\_\_\_  Т.В. Маланова

г.Иркутск 2020

Оглавление

[1 Постановка задачи 3](#_Toc50664526)

[1.1 Математическая модель 3](#_Toc50664527)

[2 Блок-схема алгоритма 4](#_Toc50664528)

[2.1 Задача 1 4](#_Toc50664529)

[2.2 Задача 2 5](#_Toc50664530)

[2.3 Задача 3 6](#_Toc50664531)

[3 Таблица спецификации 7](#_Toc50664532)

[4 Таблица тестов 8](#_Toc50664533)

[5 Результаты тестирования 9](#_Toc50664534)

[6 Исходный код 10](#_Toc50664535)

[Список использованных источников 13](#_Toc50664536)

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.

a. Дан массив чисел. Необходимо упорядочить его по убыванию.

b. Отсортируйте слова в массиве по алфавиту.

c. Выберите из предложения все слова, которые имеют четный

порядковый номер.

1.1 Математическая модель

При помощи метода “пузырька” упорядочить числа, пока они все не будут отсортированы

Отсортировать слова по первому символу в порядке возрастания

Разбить строку на слова и вывести каждое второе слово.

2 Блок-схема алгоритма

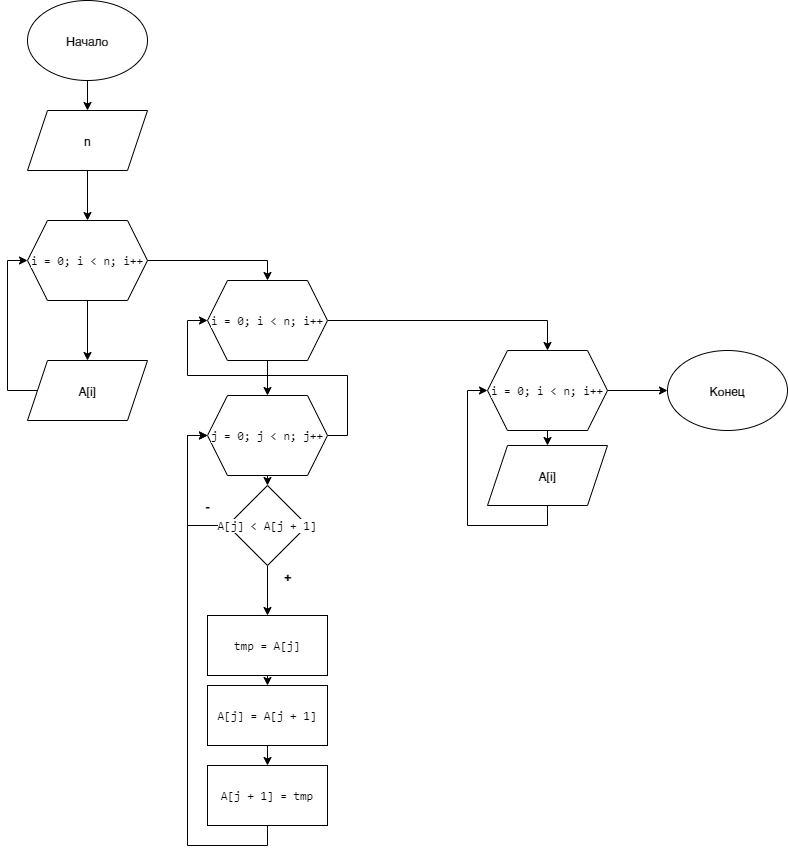
2.1 Задача 1

Рисунок 2.1 – Блок-схема алгоритма задачи 1

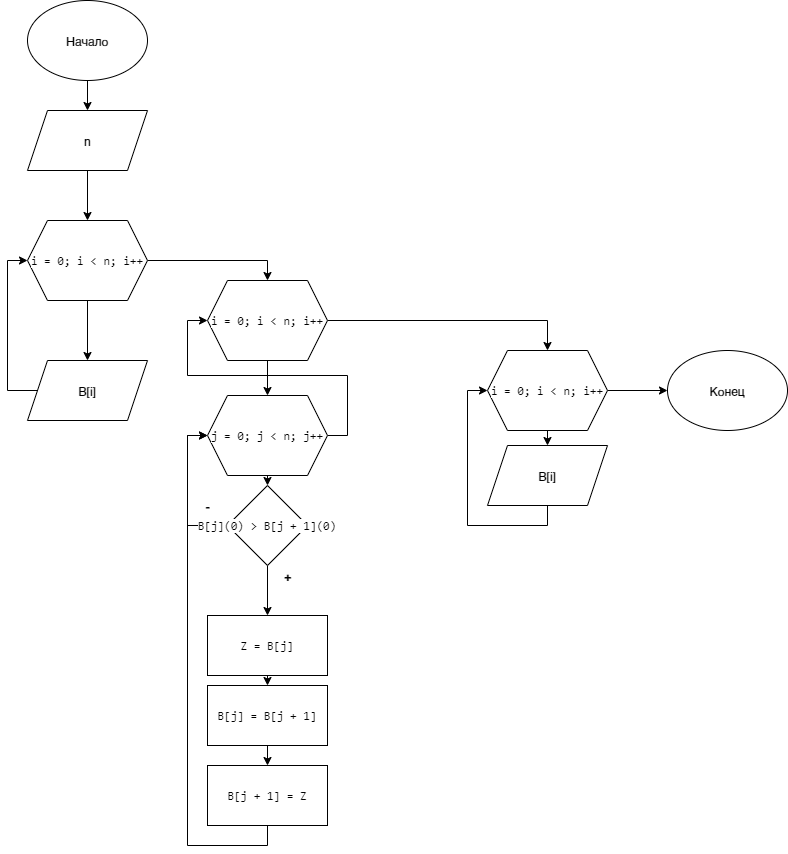
2.2 Задача 2

Рисунок 2.2 – Блок-схема алгоритма задачи 2

2.3 Задача 3

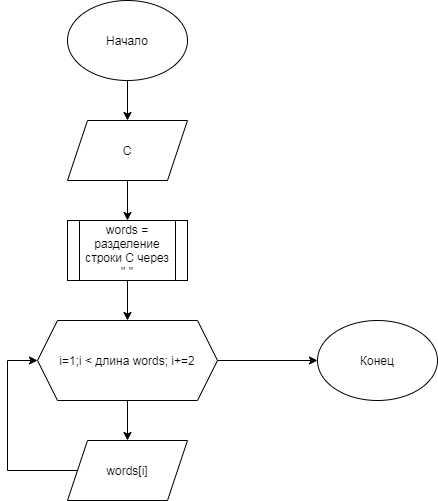


Рисунок 2.3 – Блок-схема алгоритма задачи 3

3 Таблица спецификации

Таблица 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Имя переменной | Описание переменных | Тип данных | ОДЗ |
| 1 | trigger | Переменная для выбора выполняемого задания | String | [A],[B],[C] |
| 2 | n | Переменная, отвечающая за кол-во элементов в массиве | int | [0…1000] |
| 3 | A | Массив чисел задания A | int | [0…1000] |
| 4 | i | Счетчик цикла | int | [0…1000] |
| 5 | j | Второй счетчик цикла | int | [0…1000] |
| 6 | tmp | Переменная для перемешивания чисел | int | [0…1000] |
| 7 | B | Массив строк задания B | String | [A…z] |
| 8 | Z | Переменная для перемешивания строки | String | [A…z] |
| 9 | С | Строка предложения | String | [A…z] |
| 10 | words | Массив слов из разбитого предложения | String | [A…z] |

4 Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание | Входные данные | Выходные данные |
| 1 | Проверка работы первого задания | 5  10; -10; 7; 0; 1; | 10 7 0 1 -10 |
| 2 | Проверка работы второго задания | 4  “Cycle”; “Apple”; ”Bike”; “Dive”; | Apple Bike Cycle Dive |
| 3 | Проверка работы третьего задания | “Today was a good day” | was good |

5 Результаты тестирования

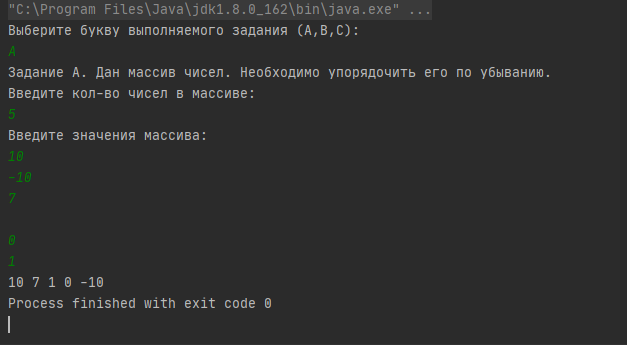


Рисунок 5.1 – Вывод IDEA после первого теста.

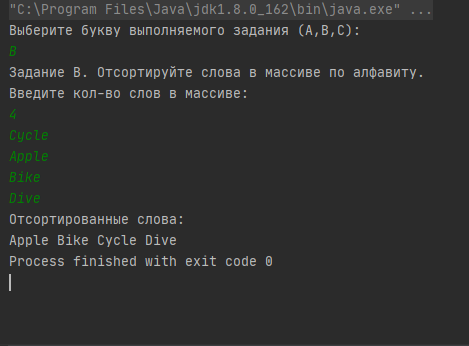


Рисунок 5.2 – Вывод IDEA после второго теста.

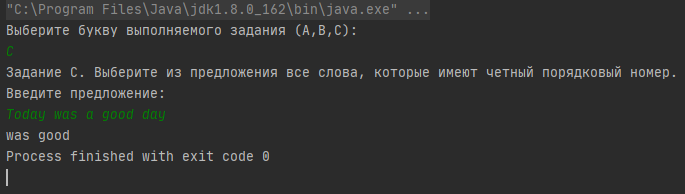


Рисунок 5.3 – Вывод IDEA после третьего теста.

6 Исходный код

package com.company;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.in);  
 System.out.println("Выберите букву выполняемого задания (A,B,C): ");  
 String trigger = in.next();  
 int n;  
 switch (trigger) {  
 //a  
 case "A": {  
 System.out.println("Задание А. Дан массив чисел. Необходимо упорядочить его по убыванию.\n" + "Введите кол-во чисел в массиве: ");  
 int tmp = 0;  
 n = in.nextInt();  
 int[] A = new int[n];  
 System.out.println("Введите значения массива:");  
 for (int i = 0; i < n; i++) {  
 A[i] = in.nextInt();  
 }  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n - 1; j++) {  
 if (A[j] < A[j + 1]) {  
 tmp = A[j];  
 A[j] = A[j + 1];  
 A[j + 1] = tmp;  
 }  
 }  
 }  
 for (int i = 0; i < n; i++) {  
 System.out.print(A[i] + " ");  
 }  
 break;  
 }  
  
 //b  
 case "B": {  
 String Z;  
 System.out.println("Задание B. Отсортируйте слова в массиве по алфавиту.\n" + "Введите кол-во слов в массиве: ");  
 n = in.nextInt();  
 String[] B = new String[n];  
 for (int i = 0; i < n; i++) {  
 B[i] = in.next();  
 }  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n - 1; j++) {  
 if (B[j].charAt(0) > B[j + 1].charAt(0)) {  
 Z = B[j];  
 B[j] = B[j + 1];  
 B[j + 1] = Z;  
 }  
 }  
 }  
 System.out.println("Отсортированные слова:");  
 for (int i = 0; i < n; i++) System.out.print(B[i] + " ");  
 break;  
 }  
  
 //c  
 case "C": {  
 System.out.println("Задание C. Выберите из предложения все слова, которые имеют четный порядковый номер. \nВведите предложение: ");  
 String C;  
 C = in.nextLine(); //для работы следующего nextLine  
 C = in.nextLine();  
 String[] words;  
 words = C.split(" "); //Разделение строки в массив слов через пробел  
 for (int i = 1; i < words.length; i += 2) System.out.print(words[i] + " ");  
 break;  
 }  
 }  
 }  
}

Список использованных источников

1. Объектно-ориентированное программирование. Метод. указания по

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

ИРНИТУ, 2017. – 24 c.