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 №2

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

[Оглавление 2](#_Toc51011166)

[1 Постановка задачи 3](#_Toc51011167)

[1.1Математическаямодель 3](#_Toc51011168)

[2 Описание структуры классов 4](#_Toc51011169)

[3 Описание методов классов и спецификация локальных переменных 5](#_Toc51011170)

[4 Таблица тестов 6](#_Toc51011171)

[5 Результаты тестирования 7](#_Toc51011172)

[6 Исходный код 9](#_Toc51011173)

[Список использованных источников 12](#_Toc51011174)

1 Постановка задачи

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 в экземпляр класса, решающего задачу должны быть разработаны соответствующие методы.

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 Описание структуры классов

Программа состоит из 3 классов: Main, TaskA, TaskB.

1. Main – основной класс.
2. TaskA – класс для выполнения первой задачи
3. TaskB – класс для выполнения второй задачи.

В свою очередь, в каждом дополнительном классе имеется конструкторы копирования, ввода и методы вывода данных.

Так же для каждого класса выделен уникальный, направленный на выполнение поставленной задачи метод.

Для TaskA это Input, Sort, Output, Copy.

Для TaskB это Input, Sort, Output.

3 Описание методов классов и спецификация локальных переменных

Таблица 3.1 – Описание методов классов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название класса | Наименование метода | Описание метода |
| 1 | Main | main | Основной метод, из которого запускаются все остальные классы |
| 2 | TaskA | Copy(TaskA other) | Конструктор копирования |
| TaskA() | Конструктор по умолчанию |
| Input(int n) | Ввод переменных |
| Sort(int n) | Метод для циклической сортировки массива digit для дальнейшего вывода |
| Output(int n) | Вывод переменных |
| 3 | TaskB | Input(int n) | Ввод переменных |
| Sort(int n) | Метод для циклической сортировки массива слов B для дальнейшего вывода |
| Output(int n) | Вывод переменных |

Таблица 3.2 – Спецификация локальных переменных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название класса | Имя переменной | Тип переменной | Назначение переменной |
| 1 | TaskA | digit | Массив int | Массив чисел, которые нужно сортировать |
| 2 | TaskB | B | Массив String | Массив слов, которые нужно сортировать |
| Z | String | Строка для перемешиваня |

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 |
| 4 | Проверка работы конструктора копирования | Скопировать файл в новый, изменить изначальный, вывести оба | Изначальный изменен, но второй остался таким же |

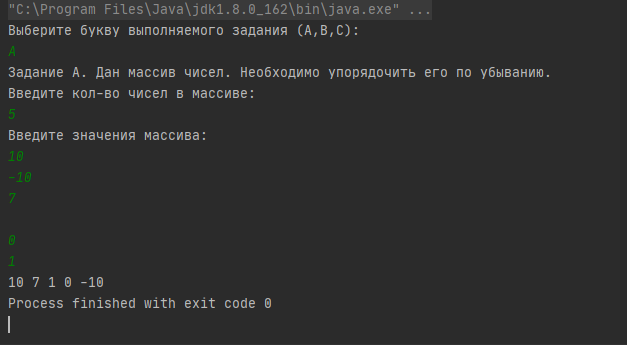
5 Результаты тестирования

Рисунок 5.1 – Вывод IDEA после первого теста.

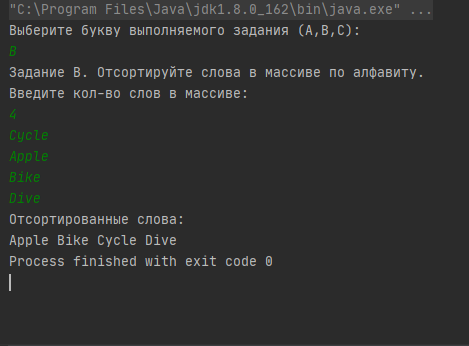


Рисунок 5.2 – Вывод IDEA после второго теста.

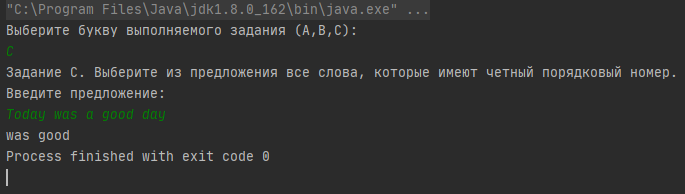


Рисунок 5.3 – Вывод IDEA после третьего теста.

6 Исходный код

Main:

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 значений массива ");

n = in.nextInt();

TaskA A = new TaskA();

A.Input(n);

A.Sort(n);

A.Output(n);

break;

}

//b

case "B": {

System.out.println("Задание B. Отсортируйте слова в массиве по алфавиту.\n" + "Введите кол-во слов в массиве: ");

n = in.nextInt();

TaskB B = new TaskB();

B.Input(n);

B.Sort(n);

B.Output(n);

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;

}

}

}

}

TaskA:

package com.company;

import java.util.Scanner;

public class TaskA {

private int[] digit;

public TaskA(TaskA p) {

}

public TaskA() { //пустой конструктор

}

public void Input(int n) {

digit = new int[n];

Scanner in = new Scanner(System.in);

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

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

digit[i] = in.nextInt();

}

}

public void Sort(int n) {

int tmp;

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

for (int j = 0; j < n - 1; j++) {

if (digit[j] < digit[j + 1]) {

tmp = digit[j];

digit[j] = digit[j + 1];

digit[j + 1] = tmp;

}

}

}

}

public void Copy(TaskA other){ //конструктор копирования

TaskA p = new TaskA();

TaskA c = new TaskA(p);

}

public void Output(int n) {

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

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

}

}

}

TaskB:

package com.company;

import java.util.Scanner;

public class TaskB {

private String[] B;

public void Input(int n) {

B = new String[n];

Scanner in = new Scanner(System.in);

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

B[i] = in.next();

}

}

public void Sort(int n) {

String Z;

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;

}

}

}

}

public void Output(int n) {

System.out.println("Отсортированные слова:");

for (int i = 0; i < n; i++) System.out.print(B[i] + " ");

}

}

Список использованных источников

1. Объектно-ориентированное программирование. Метод. указания по

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

ИРНИТУ, 2017. – 24 c.