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 №3

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](#_Toc55550816)

[1.1 Математическая модель 3](#_Toc55550817)

[2 Описание структуры пользовательского меню 4](#_Toc55550818)

[3 Проектирование классов 5](#_Toc55550819)

[4 Описание методов классов и спецификации локальных переменных 6](#_Toc55550820)

[5 Таблица тестов 8](#_Toc55550821)

[6 Результаты тестирования 9](#_Toc55550822)

[7 Исходный код 13](#_Toc55550823)

[Список использованных источников 20](#_Toc55550824)

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.

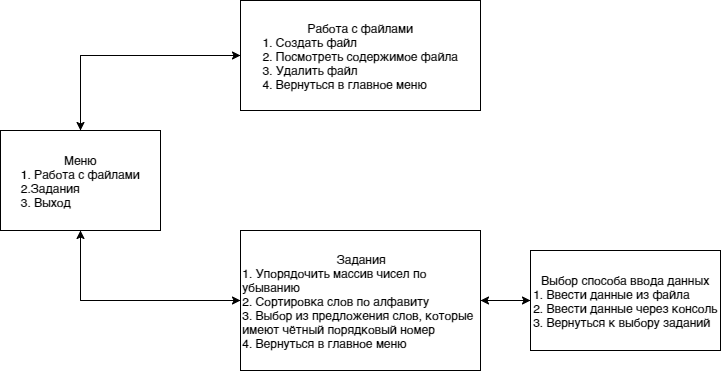
2 Описание структуры пользовательского меню

Рисунок 2.1 – Структура пользовательского меню

3 Проектирование классов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название класса | Описание | Поля |
| 1 | Main | Главный класс | - |
| 2 | File | Класс для работы с файлами | String path |
| 3 | TaskPick | Класс для выбора способа ввода данных для заданий | String path |
| 4 | Tasks | Класс для выбора задания | - |
| 5 | TaskAB | Класс для решения 1 и 2 задания | int[] digit; String[] Words; |
| 6 | TaskC | Класс для решения 3 задания | String[] words; |

4 Описание методов классов и спецификации локальных переменных

Таблица 4.1 – Описание методов классов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название класса | Наименование метода | Описание метода |
| 1 | Main | main | Основной метод, из которого запускаются все остальные классы |
| MainMenu() | Метод главного меню |
| 2 | File | Work() | Метод работы с файлами |
| 3 | TaskPick | ChoiceFile(int choice) | Метод способа ввода данных |
| FileInput() | Метод для ввода данных через файл |
| ConsoleInput() | Метод для ввода данных через консоль |
| 4 | Tasks | TasksFile() | Метод с выбором задания |
| 5 | TaskAB | TaskAB(String string, String choice) | Конструктор строки |
| TaskAB() | Конструктор по-умолчанию |
| Parse(String string) | Метод для перевода строки в массив чисел |
| Sort(String Task) | Метод для сортировки массива в завимости от задания |
| Output(String Task) | Метод для вывода данных в зависимости от задания |
| 6 | TaskC | TaskC(String Words) | Конструктор строки |
| Split(String C) | Метод для разделения строки |
| Output() | Метод для вывода чётных слов |

Таблица 4.2 – Спецификация локальных переменных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название класса | Имя переменной | Тип переменной | Назначение переменной |
| 1 | TaskAB | digit | Массив int | Массив чисел, которые нужно обработать |
| Words | Массив String | Массив слов, которые нужно обработать |
| 2 | TaskC | words | Массив String | Массив слов из разделённой строки |

5 Таблица тестов

|  |  |
| --- | --- |
| № | Описание теста |
| 1 | Подача строки вместо номера операции |
| 2 | Ввести номер не имеющийся в меню |
| 3 | Запись файла и последующее считывание |
| 4 | Попытка считать несуществующий файл |
| 5 | Ввод данных в объект из файла |

6 Результаты тестирования

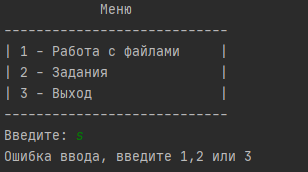


Рисунок 6.1 – Вывод NetBeans после первого теста

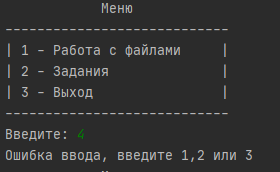


Рисунок 6.2 – Вывод NetBeans после второго теста

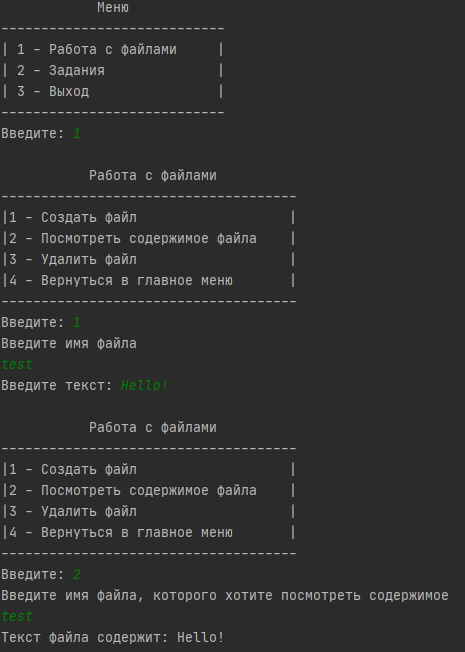
****

Рисунок 6.3 – Вывод NetBeans после сохранения

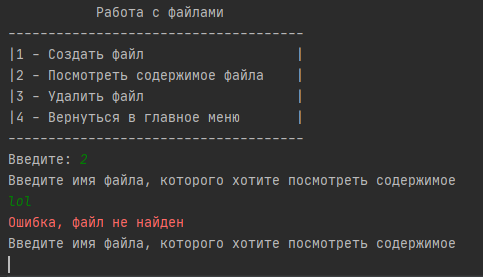


Рисунок 6.4 – Попытка считать несуществующий файл

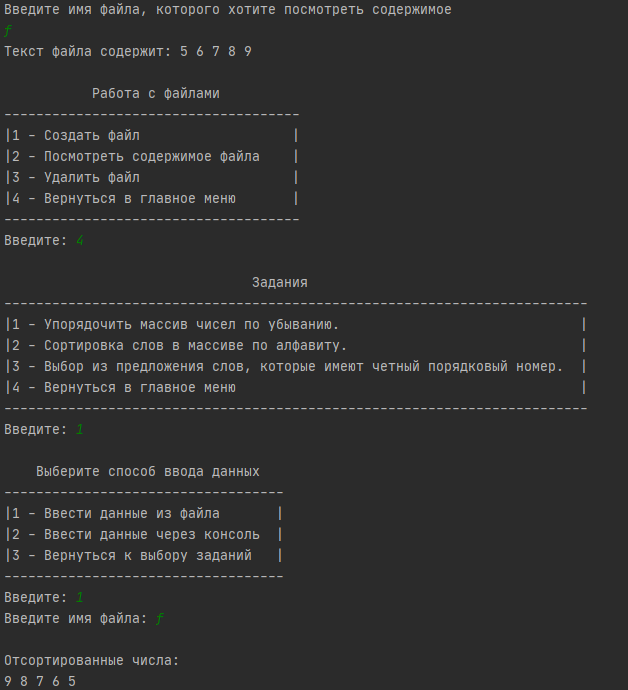


Рисунок 6.5 – Выводы NetBeans при подачи данных в объекты из файла

7 Исходный код

Main

package com.company;  
import java.util.Scanner;  
  
public class Main {  
 private static Scanner in = new Scanner(System.in);  
 public static void main(String[] args) {  
 MainMenu();  
 }  
 public static void MainMenu(){  
 int menu = 0;  
 do {  
 System.out.println(" Меню ");  
 System.out.println("----------------------------");  
 System.out.println("| 1 - Работа с файлами |");  
 System.out.println("| 2 - Задания |");  
 System.out.println("| 3 - Выход |");  
 System.out.println("----------------------------");  
 System.out.print("Введите: ");  
  
 try{  
 menu = Integer.parseInt(in.nextLine());  
 if (menu < 1 || menu > 3) throw new IllegalArgumentException();  
 }catch (IllegalArgumentException e){  
 System.out.println("Ошибка ввода, введите 1,2 или 3");  
 }  
  
 switch (menu) {  
 case 1: File.work();  
 case 2: Tasks.tasksFile();  
 }  
 } while(menu != 3);  
 System.out.println("Конец программы");  
 }  
}

**File**

package com.company;  
  
import java.io.\*;  
import java.io.FileNotFoundException;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.util.Arrays;  
import java.util.Scanner;  
  
public class File {  
 private static Scanner in = new Scanner(System.in);  
 private static String path = "D:\\untitled\\";  
 public static void work() {  
 int choice = 0;  
 do {  
 System.out.println();  
 System.out.println(" Работа с файлами ");  
 System.out.println("-------------------------------------");  
 System.out.println("|1 - Создать файл |");  
 System.out.println("|2 - Посмотреть содержимое файла |");  
 System.out.println("|3 - Удалить файл |");  
 System.out.println("|4 - Вернуться в главное меню |");  
 System.out.println("-------------------------------------");  
 System.out.print("Введите: ");  
  
  
 try {  
 choice = Integer.parseInt(in.nextLine());  
 if (choice < 1 || choice > 4) throw new IllegalArgumentException();  
 } catch (IllegalArgumentException e) {  
 System.out.println("Ошибка ввода, введите 1,2,3 или 4");  
 }  
  
 String fileName;  
 switch (choice) {  
 case 1:  
 System.out.println("Введите имя файла ");  
 fileName = in.nextLine() + ".txt";  
 FileWriter file1 = null;  
 try {  
 file1 = new FileWriter(fileName);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 System.out.print("Введите текст: ");  
 String text = in.nextLine() + ' ';  
 try {  
 file1.write(text);  
 } catch (IOException e) {  
 System.err.println("Ошибка при записи текста");  
 }  
 try {  
 file1.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 break;  
  
 case 2:  
 FileReader file2 = null;  
 do {  
 System.out.println("Введите имя файла, которого хотите посмотреть содержимое ");  
 fileName = in.nextLine() + ".txt";  
 try {  
 file2 = new FileReader(path + fileName);  
 } catch (FileNotFoundException e) {  
 System.err.println("Ошибка, файл не найден");  
 continue;  
 }  
 break;  
 } while (true);  
  
  
 char[] a = new char[255];  
  
 int length = 0;  
 try {  
 length = file2.read(a);  
 } catch (IOException e) {  
 System.err.println("Ошибка чтения");  
 }  
  
 try {  
 a = Arrays.copyOf(a, length);  
 } catch (NegativeArraySizeException e) {  
 System.out.println("Фаил пустой ");  
 break;  
 }  
 System.out.print("Текст файла содержит: ");  
 for (char b : a) {  
 System.out.print(b);  
 }  
 System.out.println();  
 try {  
 file2.close();  
 } catch (IOException e) {  
 System.out.println("Ошибка!");  
 }  
 break;  
 case 3:  
 System.out.print("Введите фаил который хотите удалить: ");  
  
 fileName = in.nextLine() + ".txt";  
  
  
 java.io.File File3 = new java.io.File(path + fileName);  
  
 if (File3.delete()) {  
 System.out.println("Файл " + fileName + " удален");  
 } else {  
 System.out.println("Файла " + fileName + " не существует ");  
 }  
 break;  
 }  
 } while (choice!=4);  
 }  
}

**TaskPick**

package com.company;  
  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.Scanner;  
  
public class TaskPick {  
 private static Scanner in = new Scanner(System.in);  
 private static String path = "D:\\untitled\\";  
 public static void choiceFile(int choice){  
 int selection = 0;  
 do{  
 System.out.println();  
 System.out.println(" Выберите способ ввода данных ");  
 System.out.println("-----------------------------------");  
 System.out.println("|1 - Ввести данные из файла |");  
 System.out.println("|2 - Ввести данные через консоль |");  
 System.out.println("|3 - Вернуться к выбору заданий |");  
 System.out.println("-----------------------------------");  
 System.out.print("Введите: ");  
 try {  
 selection = Integer.parseInt(in.nextLine());  
 if (selection < 1 || selection > 3) throw new IllegalArgumentException();  
 } catch (IllegalArgumentException e) {  
 System.out.println("Ошибка ввода, введите 1,2 или 3");  
 }  
 switch (selection) {  
 case 1:  
 try {  
 String Temp = FileInput();  
 switch (choice) {  
 case 1:  
 new TaskAB(Temp, "A");  
 break;  
 case 2:  
 new TaskAB(Temp, "B");  
 break;  
 case 3:  
 new TaskC(Temp);  
 break;  
 }  
  
 } catch (IOException e) {  
 System.err.println("Файл не обнаружен");  
 }  
 break;  
 case 2:  
 String Inp = ConsoleInput();  
 switch (choice) {  
 case 1:  
 new TaskAB(Inp, "A");  
 break;  
 case 2:  
 new TaskAB(Inp, "B");  
 break;  
 case 3:  
 new TaskC(Inp);  
 break;  
 }  
 break;  
 }  
 }while (selection!=3);  
 }  
  
  
 private static String FileInput() throws IOException {  
 StringBuilder stroka = new StringBuilder();  
 System.out.print("Введите имя файла: ");  
 String filename = in.nextLine()+".txt";  
 FileReader fr = new FileReader(path +filename);  
  
 char [] a = new char[255];  
  
 int count = 0;  
  
 try {  
 count = fr.read(a);  
 } catch (IOException ex) {  
 System.err.println("Ошибка ввода ");  
 }  
  
 for(int i = 0; i < count; i++)  
 stroka.append(a[i]);  
 fr.close();  
  
 System.out.println();  
 return stroka.toString();  
 }  
  
  
 private static String ConsoleInput(){  
 System.out.print("Введите данные через пробел: ");  
 return in.nextLine();  
 }  
}

**Tasks**

package com.company;  
  
import java.util.Scanner;  
  
public class Tasks {  
 private static Scanner in = new Scanner(System.in);  
 public static void tasksFile(){  
 int option = 0;  
 do {  
 System.out.println();  
 System.out.println(" Задания ");  
 System.out.println("-------------------------------------------------------------------------");  
 System.out.println("|1 - Упорядочить массив чисел по убыванию. |");  
 System.out.println("|2 - Сортировка слов в массиве по алфавиту. |");  
 System.out.println("|3 - Выбор из предложения слов, которые имеют четный порядковый номер. |");  
 System.out.println("|4 - Вернуться в главное меню |");  
 System.out.println("-------------------------------------------------------------------------");  
 System.out.print("Введите: ");  
  
 try{  
 option = Integer.parseInt(in.nextLine());  
 if(option < 1 || option > 4) throw new IllegalArgumentException();  
 }catch (IllegalArgumentException e){  
 System.out.println("Ошибка ввода, введите 1,2,3 или 4");  
 }  
  
 switch (option) {  
 case 1: TaskPick.choiceFile(option);  
 case 2: TaskPick.choiceFile(option);  
 case 3: TaskPick.choiceFile(option);  
 }  
 } while (option != 4);  
 }  
}

**TaskAB**

package com.company;  
  
public class TaskAB {  
 private int[] digit;  
 private String[] Words;  
 public TaskAB(String string, String choice) {  
 String[] str = string.split(" ");  
 Words = new String[str.length];  
 Words = str;  
 switch(choice){  
 case "B":  
 Sort(choice);  
 Output(choice);  
 break;  
 case "A":  
 try {  
 this.digit = Parse(string);  
 Sort(choice);  
 Output(choice);  
 } catch (NumberFormatException e) {  
 System.err.println("Ошибка ввода - ожидалось число");  
 }  
 break;  
 }  
 }  
 public TaskAB(){  
 }  
  
 private static int[] Parse(String string) {  
 String[] str = string.split(" ");  
 int size = str.length;  
 int[] arr = new int[size];  
 for (int i = 0; i < size; i++) {  
 arr[i] = Integer.parseInt(str[i]);  
 }  
 return arr;  
 }  
  
 public void Sort(String Task) {  
 switch (Task) {  
 case "A":  
 int tmp;  
 for (int i = 0; i < digit.length; i++) {  
 for (int j = 0; j < digit.length - 1; j++) {  
 if (digit[j] < digit[j + 1]) {  
 tmp = digit[j];  
 digit[j] = digit[j + 1];  
 digit[j + 1] = tmp;  
 }  
 }  
 }  
 break;  
 case "B":  
 String Z;  
 for (int i = 0; i < Words.length; i++) {  
 for (int j = 0; j < Words.length - 1; j++) {  
 if (Words[j].charAt(0) > Words[j + 1].charAt(0)) {  
 Z = Words[j];  
 Words[j] = Words[j + 1];  
 Words[j + 1] = Z;  
 }  
 }  
 }  
 break;  
 }  
 }  
  
 public TaskAB(TaskAB other){ //конструктор копирования  
 System.out.println("Скопированный массив:");  
 this.digit = other.digit;  
 }  
  
 public void Output(String Task) {  
 switch(Task) {  
 case "A":  
 System.out.println("Отсортированные числа:");  
 for (int i = 0; i < digit.length; i++) System.out.print(digit[i] + " ");  
 break;  
 case "B":  
 System.out.println("Отсортированные слова:");  
 for (int i = 0; i < Words.length; i++) System.out.print(Words[i] + " ");  
 break;  
 }  
 }  
}

**TaskC**

package com.company;  
  
public class TaskC {  
 private static String[] words;  
 public TaskC(String Words){  
 Split(Words);  
 Output();  
 }  
 public static void Split(String C) {  
 words = C.split(" "); //Разделение строки в массив слов через пробел  
 }  
 public static void Output() {  
 for (int i = 1; i < words.length; i += 2) System.out.print(words[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.