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 |
| Кафедра вычислительной техники |
| наименование кафедры |

|  |
| --- |
| **Отчет** |
| по лабораторной работе №3 по дисциплине «Работа с потоками данных и обработка исключений» |
| наименование темы  Вариант №4 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-19-1 |  |  |  | Аласаев П.М |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | Маланова Т.В. |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Работа защищена с оценкой | | | |  | | |

Иркутск 2020 г.

**Оглавление**

[1 Постановка задачи 3](#_Toc55556938)

[2 Описание структуры пользовательского меню 4](#_Toc55556939)

[3 Проектирование классов 5](#_Toc55556940)

[4 Описание методов классов и спецификации локальных переменных 6](#_Toc55556941)

[5 Таблица тестов 9](#_Toc55556942)

[6 Результаты тестирования 11](#_Toc55556943)

[7 Исходный код 18](#_Toc55556944)

[Список использованных источников 25](#_Toc55556945)

1 Постановка задачи

1. Найдите символ в строке, который встречается наибольшее число раз.
2. Найдите самое длинное слово в предложении.
3. Удалите в предложении каждое третье слово.

Вариант 4

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. Разработать класс, реализующий текстовое меню, позволяющее осуществлять выбор вариантов выполнения индивидуального задания. Создание объекта класса, решающего задание и вызов его методов, должны осуществляться из класса реализующего меню. При выполнении задания предусмотреть методы для организации ввода и выводы данных как на консоль, так и в текстовые файлы. Работа с файлами может быть организована методами класса реализующего пользовательское меню, либо в рамках специально разработанного класса. Для всех проверяемых исключений должны быть реализованы обработчики, в которых выводится сообщение о возникновении исключи- 17 тельной ситуации и предпринимаются меры для продолжения работы программы в штатном режиме.

2 Описание структуры пользовательского меню

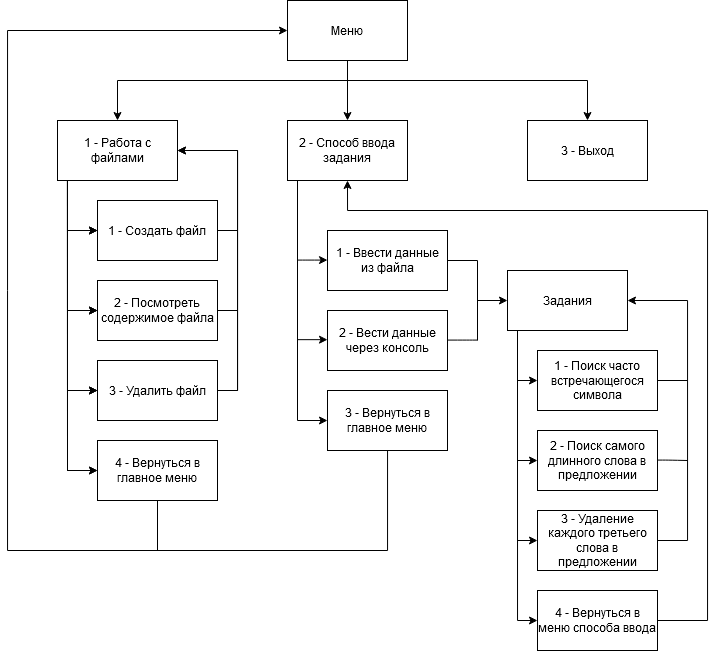


Рис 1 – Структура пользовательского меню

3 Проектирование классов

Лабораторная работа решается через шесть классов: Lab3, WorkFile, Pick, Tasks, Array и Strill.

Класс Lab3 содержит в себе только статистические методы (Работа с файлами, cпособ ввода задания и выход).

Класс WorkFile содержит в себе только статистические методы меню работы с файлами (Создать файл, посмотреть содержимое файла, удалить файл и выход в главное меню).

Класс Pick содержит в себе только статистические методы меню выбора ввода данных (Ввести данные через файл, ввести данные через консоль и выход в главное меню).

Класс Tasks содержит в себе статистические методы меню заданий (Поиск часто встречающегося символа, поиск самого длинного слова в предложении, удаление каждого третьего слова в предложении и вернуться в меню способа ввода данных).

Класс Array содержит задание, связанное с массивом.

Класс Strill содержит задания, связанные со строкой.

4 Описание методов классов и спецификации локальных переменных

Таблица 1 – Описание методов класса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Класс | Сигнатура метода | Возвра-щаемый тип | Назначение метода | Модификатор |
| Lab3 | main | void | Главное меню | public |
| WorkFaile | work | void | Работа с файлами | public |
| Pick | choiceFile | void | Выбор ввода данных | public |
| FileInput | Ввод данных через файл | private |
| ConsoleInput | Ввод данных через консоль | private |
| Tasks | tasksFile | void | Меню заданий | public |
| Array | Array | Array | Перегрузка конструктора, принимает аргументы для заполнения полей объекта переданными значениями | public |
| Array | Array | Конструктор по умолчанию. Устанавливает поля объекта значениями по умолчанию |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Класс | Сигнатура метода | Возвра-щаемый тип | Назначение метода | Модификатор |
| Array | Array | Array | Конструктор копирования. Возвращает копию объекта переданного в аргументе | public |
| setWords | Void | Метод для установки свойства |
| getWords | String[] | Возвращает массив |
| delete | void | Удаляет каждое третье слово в предложении |
| words | String[] | Хранит строку над которой будут производиться различные действия | private |
| Strill | Strill | Strill | Конструктор по умолчанию. Устанавливает поля объекта значениями по умолчанию | public |
| Strill | Strill | Перегрузка конструктора, принимает аргументы для заполнения полей объекта переданными значениями |
| Strill | Strill | Конструктор копирования. Создает копию объекта переданного в аргументе |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Класс | Сигнатура метода | Возвра-щаемый тип | Назначение метода | Модификатор |
| Strill | setString | void | Метод для установки значения | Public |
| getString | String | Метод для получения поля |
| mostLengthWord | Метод для получения самого длинного слова |
| maxConstainsChar | Метод для получения часто встречающего символа |

Таблица 2 – Спецификация локальных переменных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип | Одз |
| 1 | menu | Счетчик | int | [0; 2\*109] |
| 2 | choice | Счетчик | int | [0; 2\*109] |
| 3 | FileName | Запоминает имя файла | String | [0; 2\*109] |
| 4 | File1 | Используется для записи потока символов | String | [0; 2\*109] |
| 5 | File2 | Используется для чтения символов в файле | String | [0; 2\*109] |
| 6 | File3 | Используется для удаления файла | String | [0; 2\*109] |
| 7 | selection | Счетчик | int | [0; 2\*109] |
| 8 | option | Счетчик | int | [0; 2\*109] |
| 9 | answer | Строка вывода предложения без третьего слова | String | [0;2\* 109] |
| 10 | iChar | Строка | char | [0;2\* 109] |
| 11 | isFirst | Проверка | Boolean | [0;2\* 109] |
| 12 | max | Вывод символа, который встречается больше всего раз | string | [0;2\* 109] |
| 13 | max | Вывод самого длинного слова в предложении | String | [0;2\* 109] |
| 14 | mas | Строка | String | [0;2\* 109] |
| 15 | works | Строка | String | [0;2\* 109] |

5 Таблица тестов

Таблица 3 - Тесты

|  |  |  |
| --- | --- | --- |
| № | Описание теста | Результат работы программы |
| 1 | Ввод в меню числа, не соответствующего пунктам меню | Сообщение об ошибке ввода, повторение данного пункта меню |
| 2 | Ввод в меню некорректных символов | Сообщение об ошибке ввода, повторение данного пункта меню |
| 3 | Ввод в главном меню «1» | Переход в меню «работа с файлами» |
| 4 | Ввод в главном меню «2» | Переход в меню для выбора способа ввода текста |
| 5 | Ввод в главном меню «3» | Завершение работы программы |
| 6 | Ввод в меню работы с файлами «1»(создание файла) и корректного имени файла | Предлагает ввести имя файла и текст-содержание создаваемого файла, создает файл, выдает сообщение об успешном создании файла, переходит снова в меню работы с файлами |
| 7 | Ввод в меню работы с файлами «2»(посмотреть содержимого файла) и имени существующего файла | Показывает содержимое файла, переходит снова в меню работы с файлами |
| 8 | Ввод в меню работы с файлами «2»(посмотреть содержимого файла) и имени несуществующего файла | Сообщение об ошибке, переход снова в меню работы с файлами |
| 9 | Ввод в меню работы с файлами «3»(удалить файл) и имени существующего файла | Удаление файла, сообщение о том, что файл удален, переход снова в меню работы с файлами |
| 10 | Ввод в меню работы с файлами «3»(удалить файл) и имени несуществующего файла | Сообщение о том, что файл не обнаружен, переход снова в меню работы с файлами |
| 11 | Ввод в меню работы с файлами «4»(вернуться в главное меню | Переход в главное меню |
| 12 | Ввод в меню выбора способа ввода текста «1»(из файла) и ввода имени существующего файла | Переход в меню заданий |

Продолжение таблицы 3

|  |  |  |
| --- | --- | --- |
| № | Описание теста | Результат работы программы |
| 13 | Ввод в меню выбора способа ввода текста «1»(из файла) и ввода имени несуществующего файла | Сообщение о том, что файл не обнаружен, переход в меню выбора способа ввода |
| 14 | Ввод в меню выбора способа ввода текста «2»(из консоли) и ввода текста | Переход в меню заданий |
| 15 | Ввод в меню выбора способа ввода текста «3»(в главное меню) | Переход в главное меню |

6 Результаты тестирования

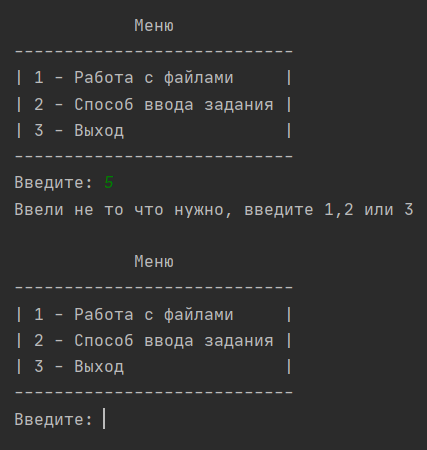


Рис 2 – Тест №1

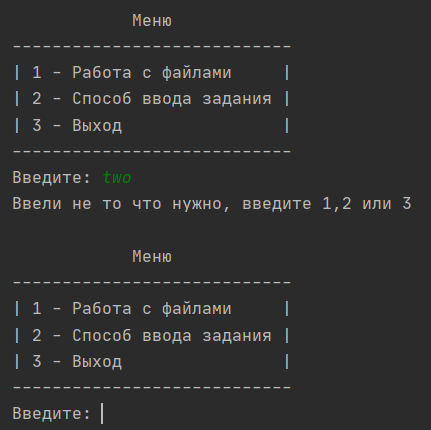


Рис 3 – Тест №2

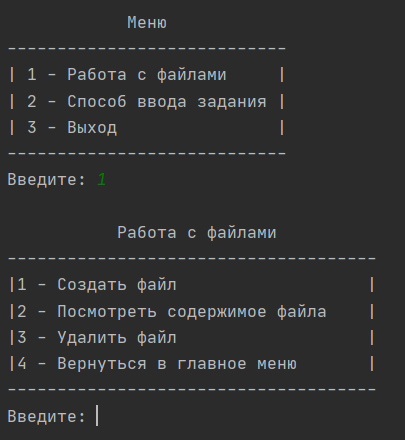


Рис 4 – Тест №3

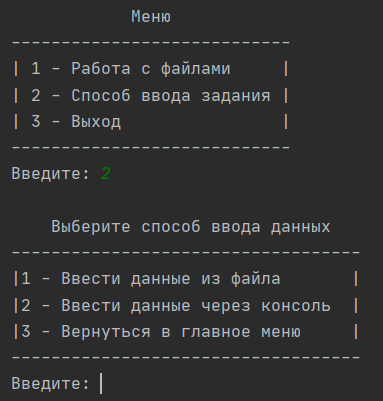


Рис 5 – Тест №4

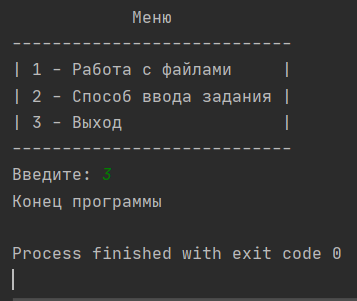


Рис 6 – Тест №5

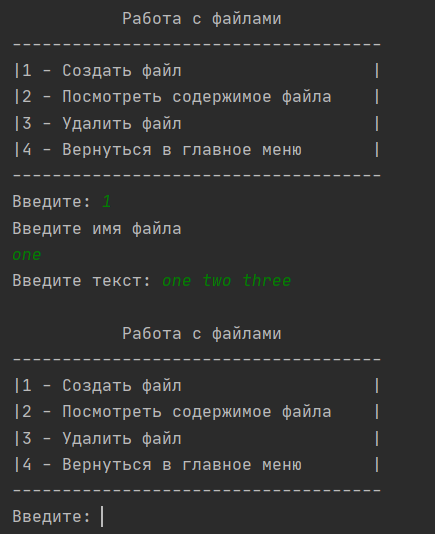


Рис 7 – Тест №6

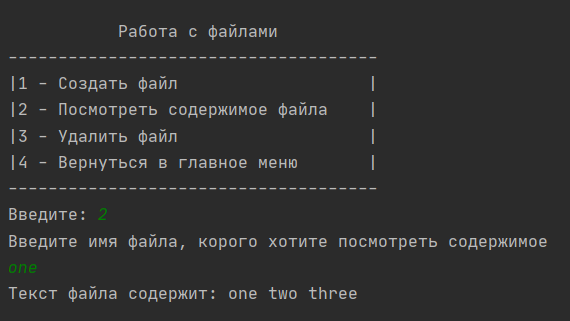


Рис 8 – Тест №7

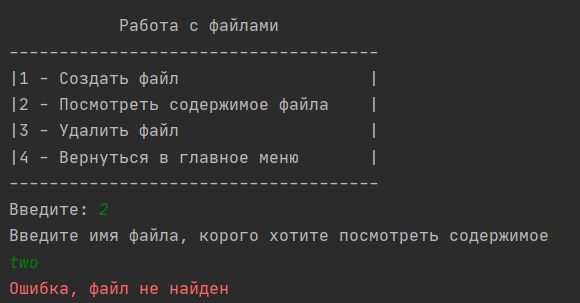


Рис 9 – Тест №8

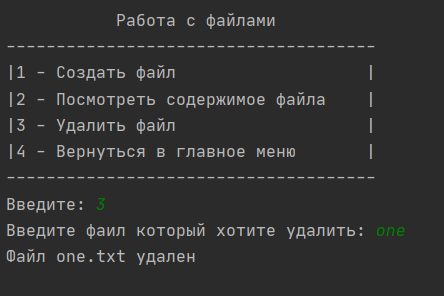


Рис 10 – Тест №9

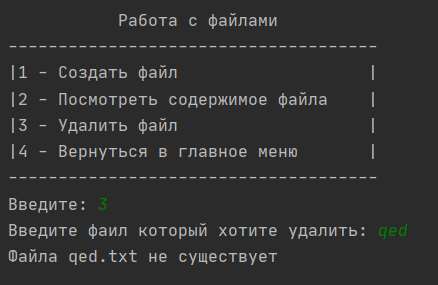


Рис 11 – Тест №10

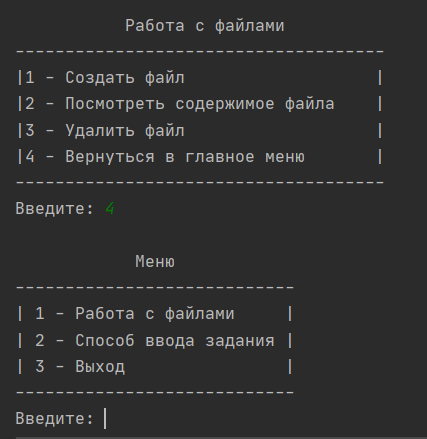


Рис 12 – Тест №11

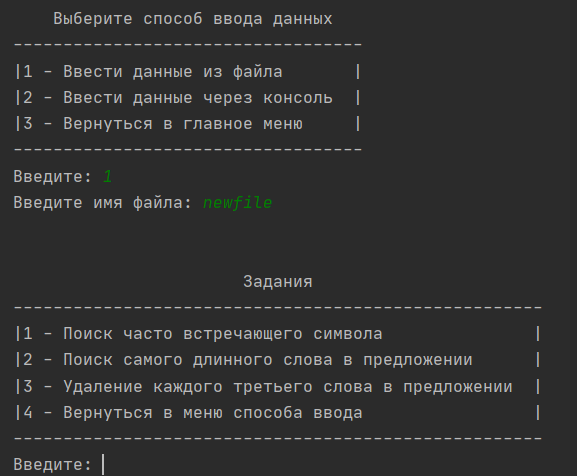


Рис 13 – Тест №12

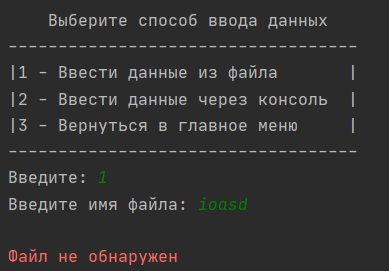


Рис 14 – Тест №13

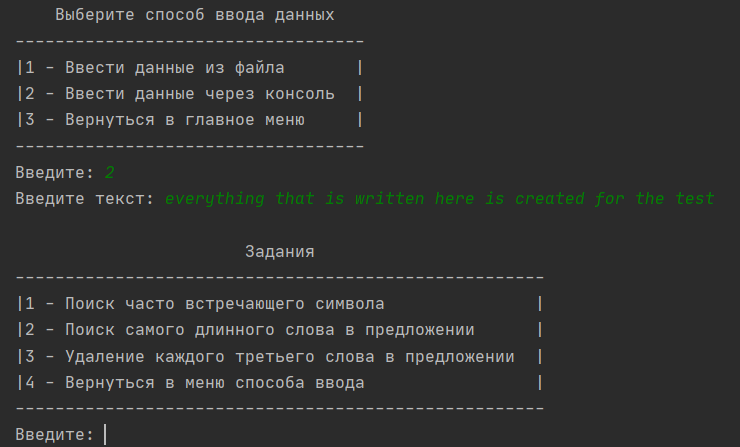


Рис 15 – Тест №14

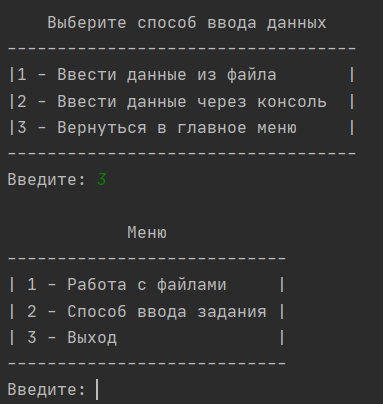


Рис 16 – Тест №15

7 Исходный код

1. Класс Lab3

package Lab;

import java.util.Scanner;

public class Lab3{

private static Scanner in = new Scanner(System.in);

public static void main(String[] args){

MenuMain();

}

private static void MenuMain(){

int menu = 0;

do {

System.out.println();

System.out.println(" Меню ");

System.out.println("----------------------------");

System.out.println("| 1 - Работа с файлами |");

System.out.println("| 2 - Способ ввода 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("Ввели не то что нужно, введите 1,2 или 3");

}

switch (menu) {

case 1 -> WorkFile.work();

case 2 -> Pick.choiceFile();

}

} while(menu != 3);

System.out.println("Конец программы");

}

}

1. Класс WorkFile

package Lab;

import java.io.\*;

import java.util.Arrays;

import java.util.Scanner;

public class WorkFile {

private static Scanner in = new Scanner(System.in);

private static String path = "C:\\Users\\alsae\\File\\";

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("Ввели не то что нужно, введите 1,2,3 или 4");

}

String fileName = null;

switch (choice){

case 1:

System.out.println("Введите имя файла ");

fileName = in.nextLine() + ".txt";

FileWriter file1 = null;

try {

file1 = new FileWriter(path + 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, корого 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");

}

a = Arrays.copyOf(a, length);

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";

File File3 = new File(path + fileName);

if(File3.delete()){

System.out.println("Файл " + fileName + " удален");

} else{

System.out.println("Файла " + fileName + " не существует ");

}

break;

}

}while(choice != 4);

}

}

1. Класс Pick

package Lab;

import java.io.\*;

import java.util.Scanner;

public class Pick {

private static Scanner in = new Scanner(System.in);

private static String path = "C:\\Users\\alsae\\File\\";

public static void choiceFile(){

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 в главное меню |");

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("Ввели не то что нужно, введите 1,2 или 3");

}

switch (selection) {

case 1:

try{

Tasks.tasksFile(FileInput());

}catch (IOException e){

System.err.println("Файл не обнаружен");

}

break;

case 2:

Tasks.tasksFile(ConsoleInput());

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 текст: ");

InputStream inputStream = System.in;

Reader inputStreamReader = new InputStreamReader(inputStream);

BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

String stroka = null;

try {

stroka = bufferedReader.readLine();

} catch (IOException ex) {

System.err.println("Ошибка ввода ");

}

return stroka;

}

}

1. Класс Tasks

package Lab;

import java.io.\*;

import java.util.Scanner;

public class Tasks{

private static Scanner in = new Scanner(System.in);

private static String path = "C:\\Users\\alsae\\File\\";

public static void tasksFile(String data){

int option = 0;

do {

System.out.println();

System.out.println(" Задания ");

System.out.println("-----------------------------------------------------");

System.out.println("|1 - Поиск часто встречающего символа |");

System.out.println("|2 - Поиск самого длинного слова в предложении |");

System.out.println("|3 - Удаление каждого третьего слова в предложении |");

System.out.println("|4 - Вернуться в меню способа ввода |");

System.out.println("-----------------------------------------------------");

System.out.print("Введите: ");

try{

option = Integer.parseInt(in.nextLine());

if(option < 1 || option > 4) throw new IOException();

}catch (IOException e){

System.out.println("Ввели не то что нужно, введите 1,2,3 или 4");

}

System.out.println("Исходные данные: " + data);

switch (option) {

case 1 -> System.out.println(new Strill(data).maxContainsChar());

case 2 -> System.out.println(new Strill(data).mostLengthWord());

case 3 -> System.out.println(new Array(data.split(" ")).delete());

}

} while (option != 4);

}

}

1. Класс Array

package Lab;

public class Array {

private String[] words;

public Array() {

this.words = "defaunt text".split(" ");

}

public Array(String[] words) {

this.words = words;

}

public Array(Array clone) {

String[] originWords = clone.getWords();

String[] newWords = new String[originWords.length];

for (int i = 0; i < originWords.length; i++) {

newWords[i] = originWords[i];

}

this.words = newWords;

}

public void setWords(String[] words) {

this.words = words;

}

public String[] getWords() {

return words;

}

public String delete() {

StringBuilder answer = new StringBuilder(words[0]).append(" ");

for (int i = 1; i < words.length; i++) {

if ((i + 1) % 3 == 0) continue;

answer.append(words[i]).append(" ");

}

return answer.toString();

}

}

1. Класс Strill

package Lab;

import java.util.HashMap;

import java.util.Map;

public class Strill {

private String string;

public Strill() {

this.string = "default text";

}

public Strill(String string) {

this.string = string;

}

public Strill(Strill clone){

this.string = clone.getString();

}

public void setString(String string) {

this.string = string;

}

public String getString() {

return string;

}

public String mostLengthWord(){

String[] mas = string.split(" ");

String max = mas[0];

for (String temp : mas) {

if (temp.length() >= max.length()) max = temp;

}

return max;

}

public char maxContainsChar() {

Map<Character, Integer> map = new HashMap<>();

for (int i = 0; i < string.length(); i++) {

char iChar = string.charAt(i);

if (iChar == ' ') continue;

// получаем число одинаовых букв, если буква встречается первый раз возращаем 0

int count = map.getOrDefault(iChar, 0);

map.put(iChar, ++count);

}

boolean isFirst = true;

Map.Entry<Character, Integer> max = null;

for (Map.Entry<Character, Integer> count : map.entrySet()) {

if (isFirst) {

max = count;

isFirst = false;

}

if (max.getValue() < count.getValue()) max = count;

}

assert max != null;

return max.getKey();

}

}

Список использованных источников

1. Методические указания по выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во ИРНИТУ, 2017. – 24 c.
2. Файн Я. – Программирование на Java для детей, родителей, дедушек и бабушек, 2011