Министерство образования и науки РФ

ФГБОУ ВО «Омский государственный технический университет»

Кафедра «Автоматизированные системы обработки информации и управления»

**ОТЧЕТ ПО ЛАБОРАТОРНЫМ РАБОТАМ**

по дисциплине «Объектно-ориентированное программирование»

студента Поцукова Максима Николаевича группы ПЭ-171

Проверил,

старший преподаватель

\_\_\_\_\_\_\_\_\_\_\_Д.Н. Шлома

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2018 г.

Выполнил студент

\_\_\_\_\_\_\_\_\_\_\_М.Н. Поцуков

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2018 г.

Омск 2018

АННОТАЦИЯ

Для решения задач, поставленных в данных лабораторных работах, использовался язык программирования Java, версии 1.8.

Набор исходного кода осуществлялся в интегрированной среде разработки Intellij IDEA Community Edition 2018.

Запуск откомпилированной программы, требует наличия на целевой ПЭВМ среды выполнения для Java, т.е. Java Runtime Environment(JRE).

В целях упрощения процедуры запуска, в директории c программой, соответствующей каждой лабораторной работе, находится скриптовый файл – для ОС, основанных на ядре Linux это \_start.sh, а для ОС семейства Microsoft Windows – \_start.bat. Запуск скриптового файла инициирует запуск программы.

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
| 1. Лабораторная работа №3……………………………….…………………..    1. Описание структуры исходного кода программы…………….…..………   1.2 Скриншоты запущенной программы…………………………….………   * 1. Ввод………………………………………………………………………   Список использованных источников………………………………………..  Приложение А1. Лабораторная работа №3. Исходный код..…………………  Приложение А2. Лабораторная работа №3. Исходный код..…………………  Приложение А3. Лабораторная работа №3. Исходный код..…………………  Приложение А4. Лабораторная работа №3. Исходный код..…………………. | 4  4  4  5  6  7  8  9  10 |

1. **Лабораторная работа №3**

Спроектировать архитектуру ПО для просмотра текстовых, мультимедийных файлов и файлов произвольного формата.

* 1. **Описание структуры программы**

В общем виде структура исходного кода программы для каждой лабораторной работы, представляет собой совокупность классов, основным из которых является класс Main.

Класс Main, содержит точку входа в Java программу в виде статического метода main, осуществляет запросы к пользователю, организует взаимодействие между классами, содержащими алгоритмы, и выводит результат работы программы.

Метод System.out.println позволяет выводить текст в консоль.

Класс Scanner используется для получения (считывания) данных введенных пользователем в виде [String](https://javadevblog.com/rekomendatsii-po-ispol-zovaniyu-string-stringbuffer-i-stringbuilder-v-java.html" \t "_blank), byte, short, int, long, float, double.

Метод nextLine в классе Scanner позволяет читать до конца текущей строки.

* 1. **Скриншоты запущенной программы**

После запуска программы, на консоль выводится номер лабораторной работы, вариант, условие задачи и запрос на ввод значений. Общий вид окна запущенной программы, приведен на рисунке 1.1.

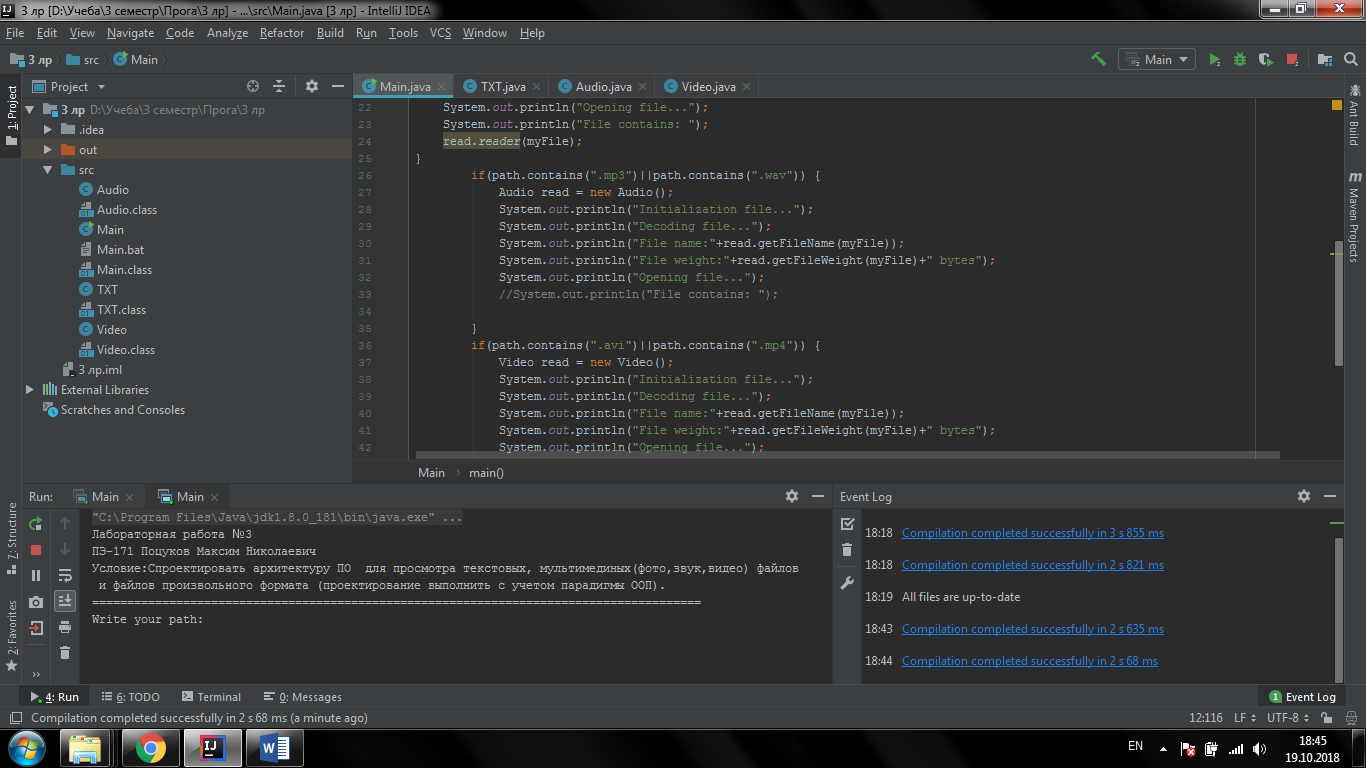


Рисунок 1.1 – Запрос на ввод пути

Результат работы программы приведен на рисунке 1.2.

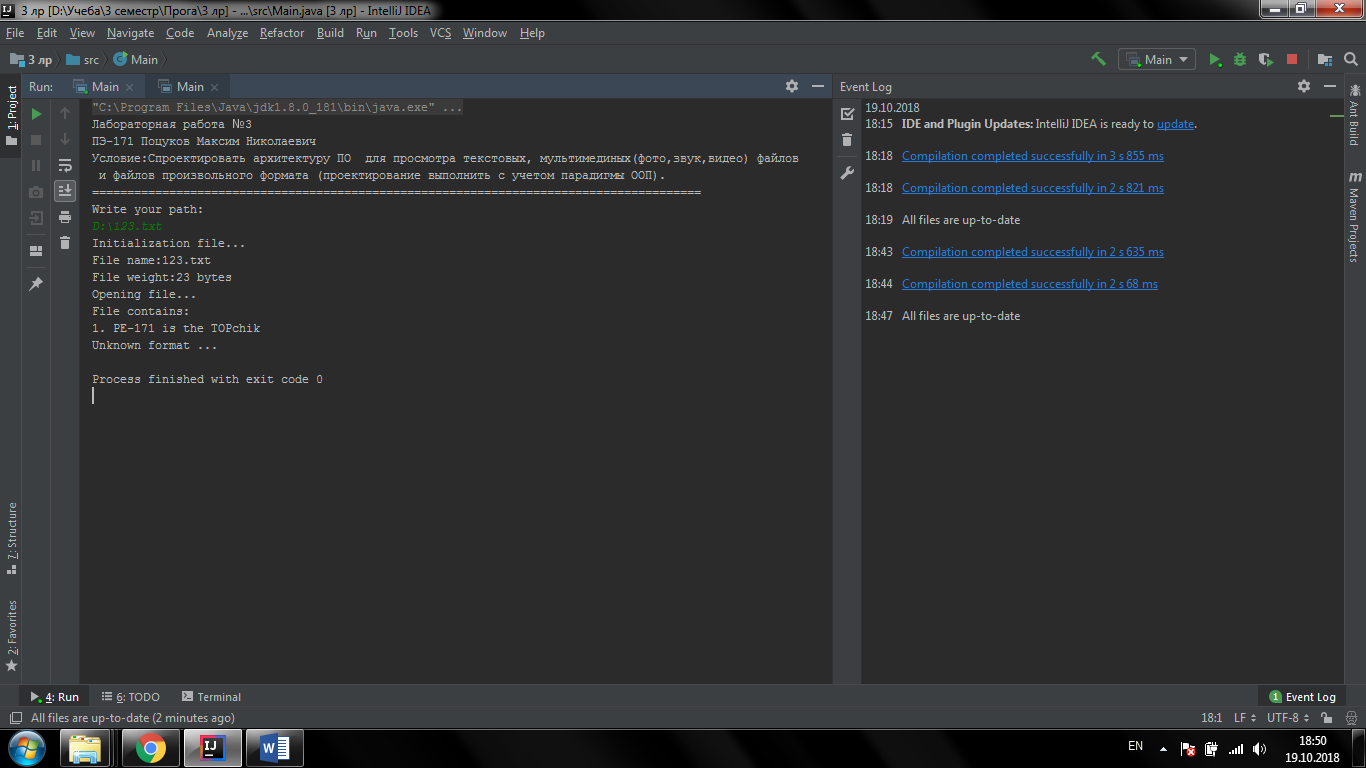


Рисунок 1.2 – Результат работы программы

По окончании работы программы, при открытии текстовых файлов отображается что записано в них, выводится наименование файла и его размер, при открытии мультимедийных файлов выводится наименование файла и его размер.

* 1. **Вывод**

В результате выполнения лабораторной работы разработана программа для открытия текстовых файлов, а точнее отображения что записано в них в консоли, вывода наименования файла и его размера, открытия мультимедийных файлов и вывод наименования файла и его размера. Исходный код программы приведен в приложениях А1, А2, А3, А4.

**Список использованных источников**

1. Шилд Г. Java 8. Полное руководство, 9-е изд. Пер. с англ. – М.: ООО «И.Д. Вильямс», 2015. – 1376 с.
2. Давыдов С. IntelliJ IDEA. Профессиональное программирование на Java / Станислав Давыдов, Алексей Ефимов. – М.: БХВ-Петербург, 2015. – 800 c.;
3. Руководство по языку программирования Java: [Сайт] https://metanit.com/java/tutorial/ дата обращения: (03.09.2018).

**Приложение А1**

**Лабораторная работа №3. Исходный код класса Main**

import java.util.Scanner;

import java.io.File;

public class Main {

public static void main(String[] args) throws Exception {

System.out.println("Лабораторная работа №3");

System.out.println("ПЭ-171 Поцуков Максим Николаевич");

System.out.println("Условие:Спроектировать архитектуру ПО для просмотра текстовых, мультимединых(фото,звук,видео) файлов\n" +

" и файлов произвольного формата (проектирование выполнить с учетом парадигмы ООП).");

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

System.out.println("Write your path: ");

Scanner scan=new Scanner(System.in);

String path=scan.next();

File myFile= new File(path);

if(path.contains(".txt")) {

TXT read = new TXT();

System.out.println("Initialization file...");

System.out.println("File name:"+read.getFileName(myFile));

System.out.println("File weight:"+read.getFileWeight(myFile)+" bytes");

System.out.println("Opening file...");

System.out.println("File contains: ");

read.reader(myFile);

}

if(path.contains(".mp3")||path.contains(".wav")) {

Audio read = new Audio();

System.out.println("Initialization file...");

System.out.println("Decoding file...");

System.out.println("File name:"+read.getFileName(myFile));

System.out.println("File weight:"+read.getFileWeight(myFile)+" bytes");

System.out.println("Opening file...");

//System.out.println("File contains: ");

}

if(path.contains(".avi")||path.contains(".mp4")) {

Video read = new Video();

System.out.println("Initialization file...");

System.out.println("Decoding file...");

System.out.println("File name:"+read.getFileName(myFile));

System.out.println("File weight:"+read.getFileWeight(myFile)+" bytes");

System.out.println("Opening file...");

//System.out.println("File contains: ");

}

else

System.out.println("Unknown format ...");

}

}

**Приложение А2**

**Лабораторная работа №3. Исходный код класса TXT**

import java.util.Scanner;

import java.io.File;

import java.io.FileReader;

public class TXT {

public String getFileName(File file){

String name=file.getName();

return name;

}

public long getFileWeight(File file){

long weight=file.length();

return weight;

}

public static void reader(File file) throws Exception{

FileReader fr=new FileReader(file);

Scanner scan=new Scanner((fr));

int i=1;

while(scan.hasNextLine()){

System.out.println(i+". "+scan.nextLine());

i++;

}

fr.close();

}

}

**Приложение А3**

**Лабораторная работа №3. Исходный код класса Audio**

import java.util.Scanner;

import java.io.File;

import java.io.FileReader;

public class Audio {

public String getFileName(File file){

String name=file.getName();

return name;

}

public long getFileWeight(File file){

long weight=file.length();

return weight;

}

// public static void reader(File file) throws Exception{

// FileReader fr=new FileReader(file);

// Scanner scan=new Scanner((fr));

// int i=1;

// while(scan.hasNextLine()){

// System.out.println(i+". "+scan.nextLine());

// i++;

// }

// fr.close();

//

// }

}

**Приложение А4**

**Лабораторная работа №3. Исходный код класса Video**

import java.util.Scanner;

import java.io.File;

import java.io.FileReader;

public class Video {

public String getFileName(File file){

String name=file.getName();

return name;

}

public long getFileWeight(File file){

long weight=file.length();

return weight;

}

// public static void reader(File file) throws Exception{

// FileReader fr=new FileReader(file);

// Scanner scan=new Scanner((fr));

// int i=1;

// while(scan.hasNextLine()){

// System.out.println(i+". "+scan.nextLine());

// i++;

// }

// fr.close();

//

// }

}