ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования
«Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

Кафедра Безопасности информационных систем

Отчет по лабораторной работе №4

«Использование операторов и функций. Приложение, реализующее файловый ввод – вывод.»

По дисциплине: «Кроссплатформенное программирование»

Цель работы:

1. Изучение общих принципов программирования в среде java.
2. Написание приложения, осуществляющего взаимодействия со средствами дискового ввода – вывода.

Теоретическая часть:

Поскольку «интерфейс» данной программы представляет собой терминал с конечным количеством команд, реализован он был посредствам условной конструкции if – else if – else. Каждая из команд «терминала» реализована в отдельной функции.

Так как реализовывался терминал, необходим был доступ к файловой системе, он в свою очередь был обеспечен через класс File. Он располагается в java.io.File. У данного класса много методов, приведу в пример лишь те, которые использовал, непосредственно, в программе:

· listFiles – возвращает список файлов в данной директории.

· mkDir – создает новый каталог в данной директории.

· createNewFile – создаёт новый файл в данной директории.

· delete – удаляет элемент и возврщает результат удаления (boolean).

Одна из функций данной программы получает аттрибуты файла (размер, дата создания, время последнего доступа и т. д.). Это было выполнено с помощью класса BasicFileAttributes, который находится в java.nio.file.attribute.BasicFileAttributes. Для получения данных, были применены соответствующие методы данного класса creationTime(), lastAccessTime(), size() и т. д.

Также одна из функций открывает файл в notepad. Это было реализовано с помощью класса Process. Объекты у этого метода получаются вызовом метода exec() у объекта Runtime, который в свою очередь запускает отдельный процесс. Объект класса Process может использоваться для управления процессом и получения информации о нём.

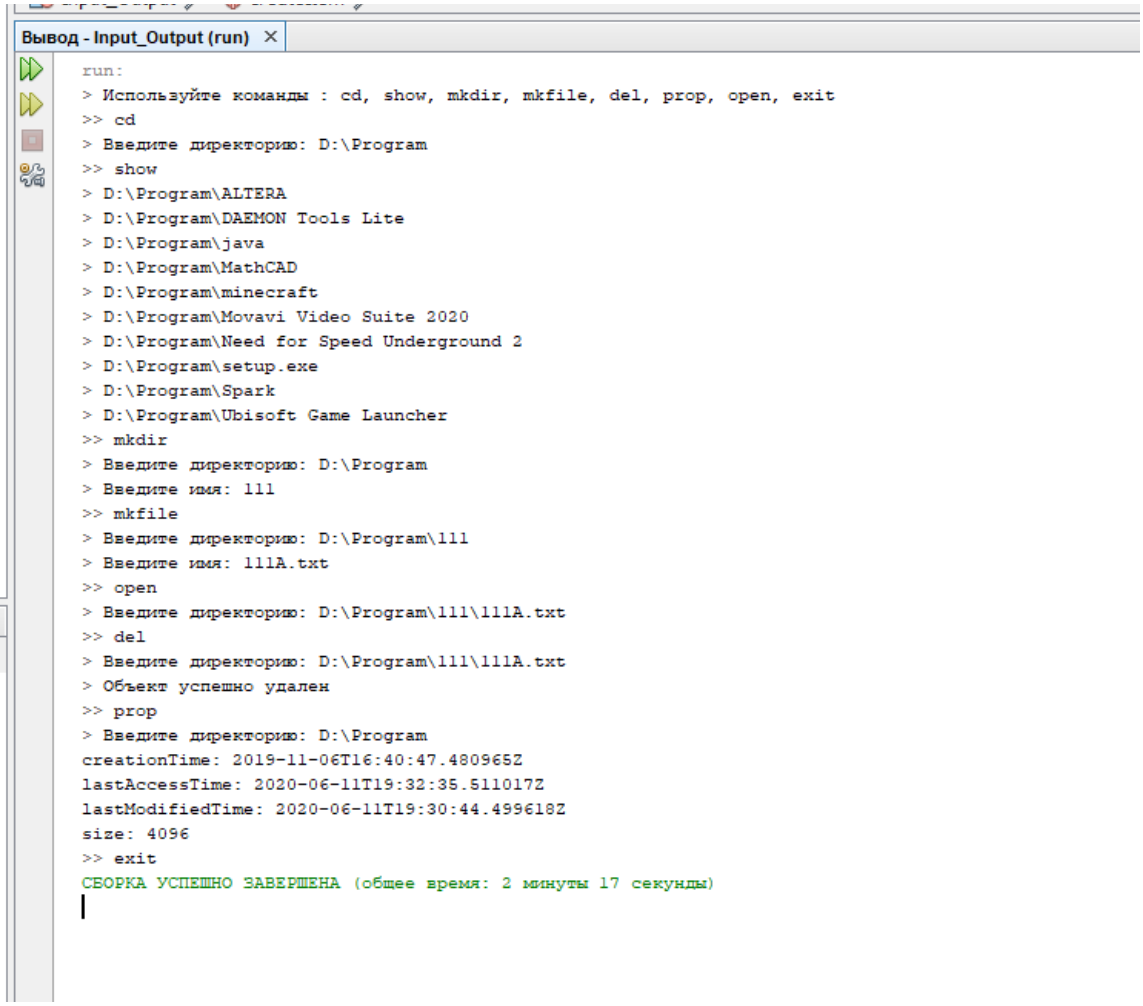
Так как некоторые методы данных классов могут выбрасывать исключения, они были обработаны в блоках try – catch. В качестве «обработки» исключений был применеён метод printStackTrace, который возвращал место ошибки.

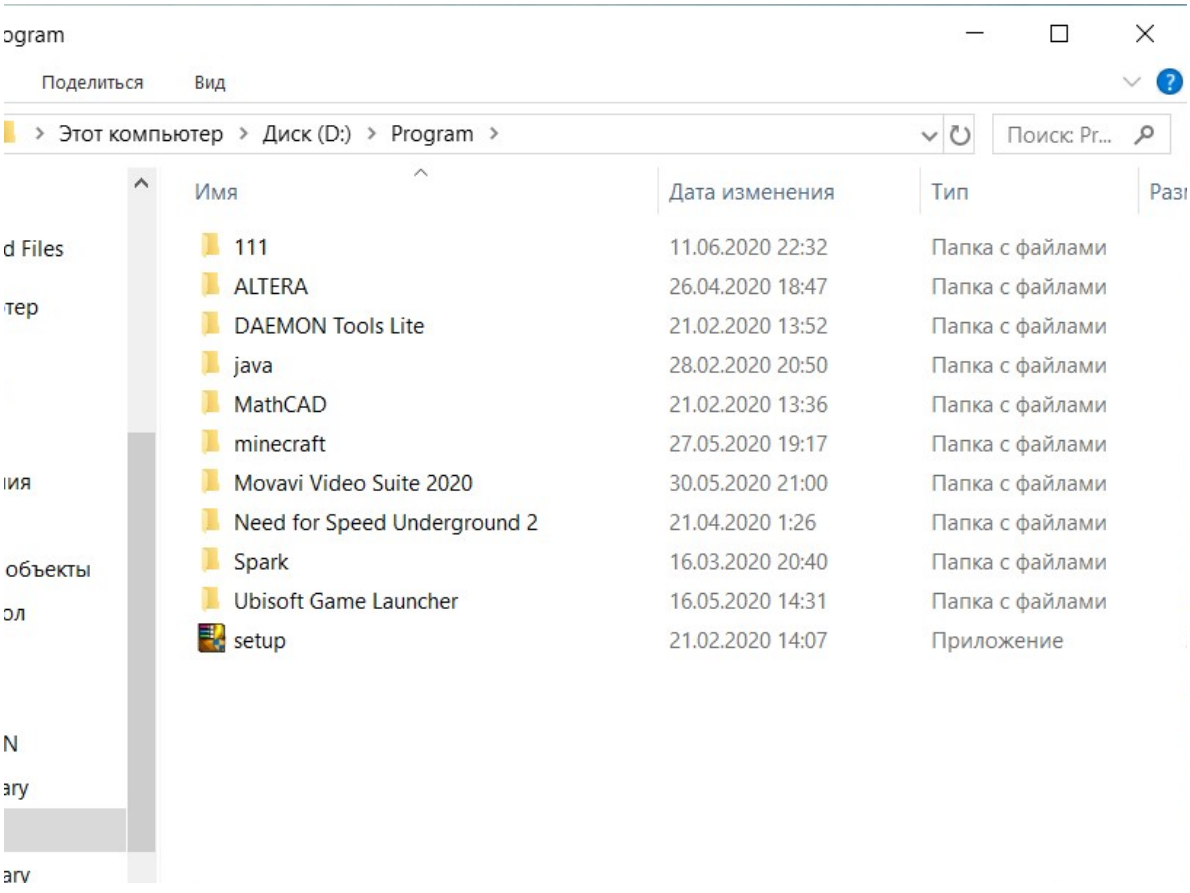
Ход работы:

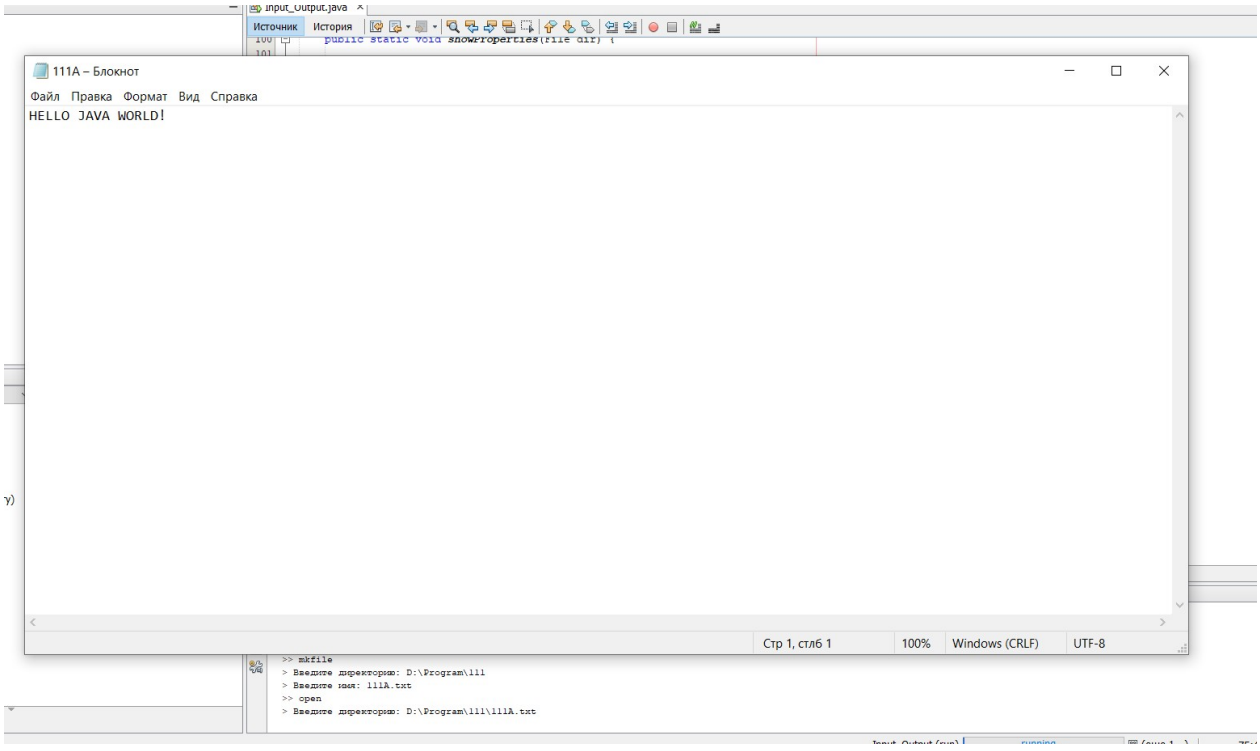
```
import java.io.File;
import java.io.IOException;
import java.util.*;
import java.nio.file.Files;
import java.nio.file.attribute.BasicFileAttributes;

public class Input_Output {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);

        String answer = "";
        String currentDirectory = "";

        System.out.println("> Используйте команды : cd, show, mkdir, mkfile,
del, prop, open, exit");

        while (!answer.equals("exit")) {
            System.out.print(">> ");
            answer = scn.next().toLowerCase();

            if (answer.equals("cd")) {
                currentDirectory = changeDirectory();
            }

            else if (answer.equals("show")) {
                if (!currentDirectory.isEmpty())
                    showItemsFromDirectory(new File(currentDirectory));
                else
                    System.out.print("> Сначала нужно войти в директорию ");
            }

            else if (answer.equals("mkdir") | answer.equals("mkfile")) {
                System.out.print("> Введите директорию: ");
                String dir = scn.next();

                createItem(new File(dir), answer);
            }

            else if (answer.equals("del")) {
                System.out.print("> Введите директорию: ");
                String dir = scn.next();

                removeItem(new File(dir));
            }

            else if (answer.equals("prop")) {
                System.out.print("> Введите директорию: ");
                String dir = scn.next();

                showProperties(new File(dir));
            }

            else if (answer.equals("open")) {
                System.out.print("> Введите директорию: ");
                String dir = scn.next();

                openFile(dir);
            }

            else if (!answer.equals("exit"))
                System.out.println("> Неизвестная команда");
```

```
        }
    }

    public static void showItemsFromDirectory(File directory) {
        if (directory.isDirectory()) {
            for (File item : directory.listFiles()) {
                System.out.println("> " + item);
            }
        }
    }

    public static void createItem(File dir, String com) {
        Scanner scn = new Scanner(System.in);
        System.out.print("> Введите имя: ");
        String ans = scn.next();
        File newItem = new File(dir.toString() + "//" + ans);

        if (com.equals("mkdir")) {
            newItem.mkdir();
        }
        else {
            try {
                newItem.createNewFile();
            }
            catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    public static void removeItem(File dir) {
        boolean deleted = dir.delete();
        if(deleted)
            System.out.println("> Объект успешно удален ");
        else
            System.out.println("> Невозможно удалить данный объект ");
    }

    public static void showProperties(File dir) {

        try {
            BasicFileAttributes attr = Files.readAttributes(dir.toPath(),
BasicFileAttributes.class);

            System.out.println("creationTime: " + attr.creationTime());
            System.out.println("lastAccessTime: " + attr.lastAccessTime());
            System.out.println("lastModifiedTime: " +
attr.lastModifiedTime());
            System.out.println("size: " + attr.size());
        }

        catch(IOException e) {
            e.printStackTrace();
        }
    }

    public static String changeDirectory() {
        Scanner scn = new Scanner(System.in);
        System.out.print("> Введите директорию: ");
        String dir = scn.next();

        return dir;
    }
```

```
        public static void openFile(String path) {
            try {
                Process process = Runtime.getRuntime().exec("cmd /c notepad.exe "
    + path);
                process.waitFor();
            }

            catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    }
```







Выводы:

- В ходе выполнения лабораторной работы произошло знакомство с операциями над файлами.