**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА»**

**Факультет информационных систем и технологий**

**Кафедра безопасности информационных систем**

**ОТЧЁТ**

по вводному практическому занятию

по дисциплине Кросс-платформенное программирование

**на тему:**

**«Разработка Java-приложения с использованием простых типов данных и символьных строк**»

Выполнили студенты

группы ИСТ-221:

Танасиенко В.С., \_\_\_\_\_\_\_\_

Кормачев И.Д., \_\_\_\_\_\_\_\_

Проверил:

Бородянский Ю.М.

\_\_\_\_\_\_\_\_

г. Санкт-Петербург, 2024 г.

**Цели:** получение практических навыков создания Java-приложения для получения сведений о файлах и реализации файлового ввода-вывода, совершенствование приёмов программирования на языке Java

**Задание**. Разработать приложение «файловый менеджер». Приложение должно иметь простой интерфейс и осуществлять все основные функции файловых менеджеров:

- постоянное отображение адресной строки;

- выбор нужного каталога в файловом дереве;

- вывод содержимого каталога;

- перемещение по уровням вложенности каталогов;

- вывод свойств выбранного файла;

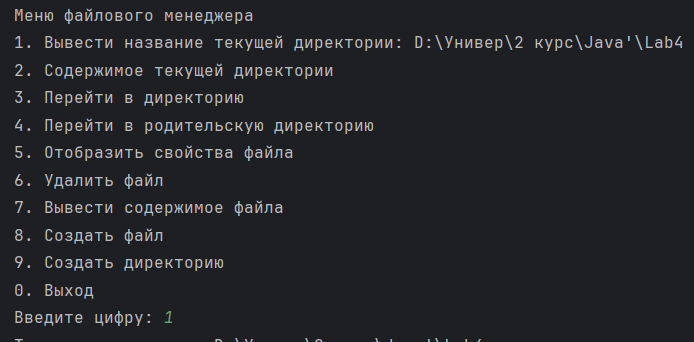
- удаление выбранного файла;

- чтение содержимого файла путем запуска программы notepad.exe;

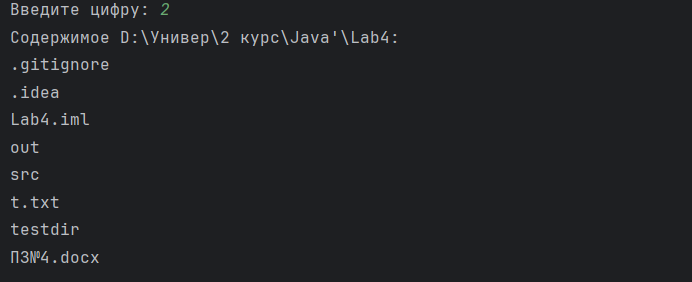
- создание простого текстового файла и запись его на диск;

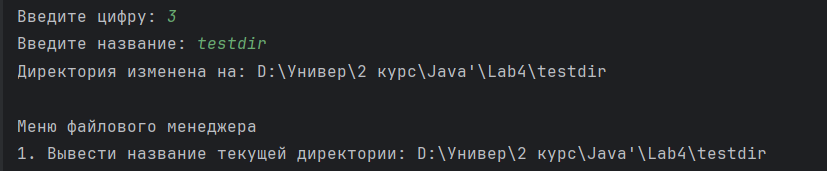
- создание нового каталога.

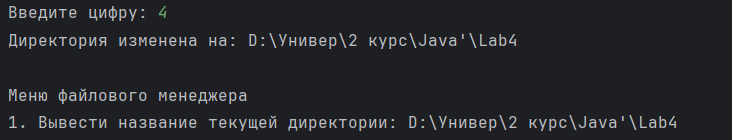
**Ход выполнения работы:**

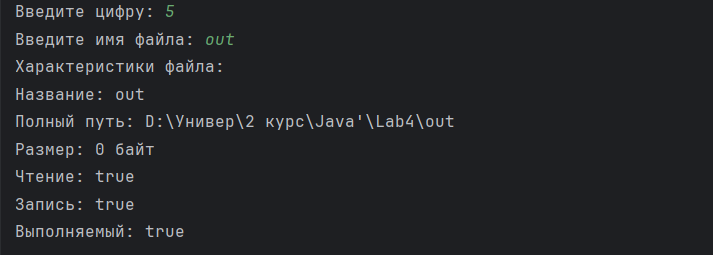
****

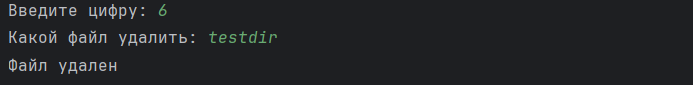
****

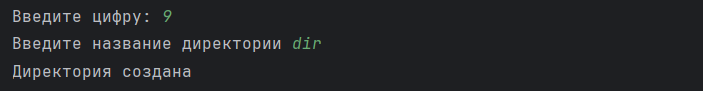
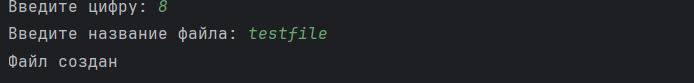
****

****

****

****

****

****

**Выводы:**

В ходе проделанной работы были получены навыки работы с файлами на языке java, методы создания и удаления файлов и директорий, открытие файлов средствами сторонних программ, создания модели файлового менеджера. В дальнейшем эти знания могут быть полезными в создании программ более высокого уровня, в которых будет использоваться взаимодействие программного кода и сторонних приложений, в управлении файловой системой компьютера.

**Приложение:**

**Код программы:**

import java.io.\*;

import java.util.\*;

public class Main {

public static Scanner scanner = new Scanner(System.in);

public static File currentDir;

public static void main(String[] args) {

currentDir = new File(System.getProperty("user.dir"));

while (true) {

displayMenu();

int choice = getUserChoice();

switch (choice) {

case 1:

displayCurrentDirectory();

break;

case 2:

listContents();

break;

case 3:

navigateDirectory();

break;

case 4:

navigateToParentDirectory();

break;

case 5:

displayFileProperties();

break;

case 6:

deleteFile();

break;

case 7:

readFileContents();

break;

case 8:

createTextFile();

break;

case 9:

createDirectory();

break;

case 0:

System.out.println("Выход");

System.exit(0);

break;

default:

System.out.println("Неверный ввод");

}

}

}

public static void displayMenu() {

System.out.println("\nМеню файлового менеджера");

System.out.println("1. Вывести название текущей директории: " + currentDir.getAbsolutePath());

System.out.println("2. Содержимое текущей директории");

System.out.println("3. Перейти в директорию");

System.out.println("4. Перейти в родительскую директорию");

System.out.println("5. Отобразить свойства файла");

System.out.println("6. Удалить файл");

System.out.println("7. Вывести содержимое файла");

System.out.println("8. Создать файл");

System.out.println("9. Создать директорию");

System.out.println("0. Выход");

System.out.print("Введите цифру: ");

}

public static int getUserChoice() {

int choice = scanner.nextInt();

scanner.nextLine();

return choice;

}

public static void displayCurrentDirectory() {

System.out.println("Текущая директория: " + currentDir.getAbsolutePath());

}

public static void listContents() {

System.out.println("Содержимое " + currentDir.getAbsolutePath() + ":");

File[] files = currentDir.listFiles();

if (files != null) {

for (File file : files) {

System.out.println(file.getName());

}

}

}

public static void navigateDirectory() {

System.out.print("Введите название: ");

String directoryName = scanner.nextLine();

File newDir = new File(currentDir, directoryName);

if (newDir.isDirectory()) {

currentDir = newDir;

System.out.println("Директория изменена на: " + currentDir.getAbsolutePath());

} else {

System.out.println("Не является директорией");

}

}

public static void displayFileProperties() {

System.out.print("Введите имя файла: ");

String fileName = scanner.nextLine();

File file = new File(currentDir, fileName);

if (file.exists()) {

System.out.println("Характеристики файла:");

System.out.println("Название: " + file.getName());

System.out.println("Полный путь: " + file.getAbsolutePath());

System.out.println("Размер: " + file.length() + " байт");

System.out.println("Чтение: " + file.canRead());

System.out.println("Запись: " + file.canWrite());

System.out.println("Выполняемый: " + file.canExecute());

} else {

System.out.println("Файл не найден");

}

}

public static void deleteFile() {

System.out.print("Какой файл удалить: ");

String fileName = scanner.nextLine();

File file = new File(currentDir, fileName);

if (file.exists()) {

if (file.delete()) {

System.out.println("Файл удален");

} else {

System.out.println("Не удалось удалить");

}

} else {

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

}

}

public static void readFileContents() {

System.out.print("Введите название файла: ");

String fileName = scanner.nextLine();

try {

Process process = Runtime.getRuntime().exec("notepad.exe " + fileName, null, currentDir);

} catch (IOException e) {

e.printStackTrace();

}

}

private static void createTextFile() {

try {

System.out.print("Введите название файла: ");

String fileName = scanner.nextLine();

File file = new File(currentDir, fileName);

if (file.createNewFile()) {

System.out.println("Файл создан");

} else {

System.out.println("Не удалось создать");

}

} catch (IOException e) {

e.printStackTrace();

}

}

private static void createDirectory() {

System.out.print("Введите название директории ");

String directoryName = scanner.nextLine();

File newDir = new File(currentDir, directoryName);

if (newDir.mkdir()) {

System.out.println("Директория создана");

} else {

System.out.println("Не удалось создать");

}

}

private static void navigateToParentDirectory() {

File parentDir = currentDir.getParentFile();

if (parentDir != null) {

currentDir = parentDir;

System.out.println("Директория изменена на: " + currentDir.getAbsolutePath());

} else {

System.out.println("Уже в корневой директории");

}

}

}