**Name: Sreelakshmi M Nair**

**Roll No: 40**

**Batch: RMCA**

**Date:31-05-2022**

**Object Oriented Programming LAB**

**Experiment No.: 17**

**Aim**

Program to list the sub directories and files in a given directory and also search for a file name.

**Procedures**

**Source Code**

import **java.io.File**;

import **java.io.\***;

import **java.util.\***;

**public** **class** p1 {

**public** **static** **final** **String** RESET = "\033[0m";

**public** **static** **final** **String** RED = "\033[0;31m";

**public** **static** **final** **String** TEXT\_RESET = "\u001B[0m";

**public** **static** **final** **String** TEXT\_BLACK = "\u001B[30m";

**public** **static** **final** **String** TEXT\_RED = "\u001B[31m";

**static** **void** RecursivePrint(**File**[] arr, **int** index, **int** level, **String** searchfor)

 {

    if (index == arr.length)

    return;

    for (**int** i = 0; i < level; i++)

    System.out.print("\t");

    if (arr[index].getName().toLowerCase().contains(searchfor))

    System.out.print(TEXT\_RED);

    else

    System.out.print(RESET);

    if (arr[index].isFile())

    System.out.println(arr[index].getName());

    else if (arr[index].isDirectory()) {

        System.out.println("[" + arr[index].getName() + "]");

        RecursivePrint(arr[index].listFiles(), 0, level + 1, searchfor);

        }

        RecursivePrint(arr, ++index, level, searchfor);

        }

**public** **static** **void** main(**String**[] args) {

**Scanner** scan = new Scanner(System.in);

            System.out.println("Enter the directory path");

**String** maindirpath = scan.nextLine();

            System.out.println("Enter the file/directory name to search");

**String** searchfor = scan.nextLine();

**File** maindir = new File(maindirpath);

            if (maindir.exists() && maindir.isDirectory())

            {

**File** arr[] = maindir.listFiles();

                System.out.println("Files from main directory" + maindir);

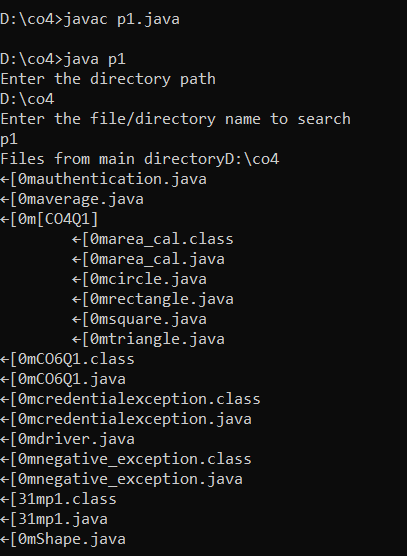
                RecursivePrint(arr, 0, 0, searchfor.toLowerCase());

                }

}

}

**Output**



**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 18**

**Aim**

Write a program to write to a file, then read from the file and display the contents on the console.

**Procedure**

**Source Code**

import **java.io.BufferedReader**;

import **java.io.FileReader**;

import **java.io.FileWriter**;

import **java.io.IOException**;

**public** **class** Files {

**public** **static** **void** main(**String**[] args) {

        try {

**FileWriter** writer = new FileWriter("Files.txt",true);

            writer.write("new file is created");

            writer.close();

**FileReader** reader = new FileReader("Files.txt");

**BufferedReader** br= new BufferedReader(reader);

**String** line;

            System.out.println("Data read from the file");

            while ((line = br.readLine()) != null) {

                System.out.println(line);

            }

            reader.close();

        } catch (**IOException** e) {

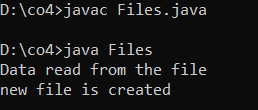
            System.out.println("-----Error-----");

        }

    }

}

**Output Screenshot**



**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 19**

**Aim**

Write a program to copy one file to another.

**Procedure**

**Source Code**

import **java.io.FileInputStream**;

import **java.io.FileOutputStream**;

import **java.io.IOException**;

**public** **class** Copy {

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

**FileInputStream** fileinput = new FileInputStream("1.txt");

**FileOutputStream** fileoutput = new FileOutputStream("2.txt");

**int** i;

    while ((i = fileinput.read()) != -1) {

        fileoutput.write(i);

        }

        System.out.println("Successfully copied one file to another");

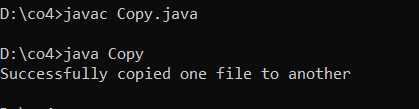
        fileinput.close();

        fileoutput.close();

    }

}

**Output Screenshot**



**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 20**

**Aim**

Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

**Procedure**

**Source Code**

import **java.io.FileInputStream**;

import **java.io.FileOutputStream**;

import **java.io.IOException**;

**public** **class** Integers

{

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

**FileInputStream** source = new FileInputStream ("source.txt");

**FileOutputStream** destination\_odd = new FileOutputStream ("odd.txt");

**FileOutputStream** destination\_even = new FileOutputStream ("even.txt");

**int** i;

        while((i = source.read()) != -1){

            if(i%2==0) {

                destination\_even.write(i);

            }

            else {

                destination\_odd.write(i);

            }

        }

        System.out.println("copied");

        source.close();

        destination\_even.close();

        destination\_odd.close();

    }

}

**Output Screenshot**

