**Recursion class**

import java.io.File;

/\*\*

\*

\* @author Richelin Metellus

\* @version 02/10/2017

\* This class implement the methods to find basic operation, and file finder;

\*/

public class Recursion {

/\*\*

\*

\* @param num the log base2 of the parameter;

\* @return

\*/

public static boolean match = false;

public static int log2(int num)

{

if ( num == 1)

return 0;

else

return 1+ log2(num/2);

}

/\*\*

\*

\* @param num1 multiplicand

\* @param num2 multiplier

\* @return

\*/

public static int prodOf2PositiveInt(int num1, int num2)

{

int temp;

// swap multiplican to multiplier if mulitplier is larger

if (num2 > num1)

{

temp = num1;

num1 = num2;

num2 = temp;

}

if (num2 == 0 || num1== 0)

return 0;

else

return num1 + prodOf2PositiveInt(num1, (num2-1));

}

/\*\*

\*

\* @param A array of variable size

\* @return the sum of integer in the array.

\*/

public static int isabelTech4ArrSum( int[] A)

{

int [] B = new int[A.length/2];

for(int i = 0; i < A.length/2; i++)

{

B[i] = A[2\*i] + A[2\*i + 1];

}

if(B.length == 1)

return B[0];

else

return isabelTech4ArrSum(B);

}

/\*\*

\*

\* @param path directory of the file

\* @param name name of the file searching for

\*/

public static void find(String path,String name)

{

File f = new File(path);

if(f.isFile())

{

if(f.getName().equalsIgnoreCase(name))

{

{

match = true;

System.out.println("File Found at: " + f.getAbsolutePath());

}

}

}

else if(f.isDirectory())

{

File [] folder = f.listFiles();

if(folder != null)

{

for(File item: folder)

{

find(item.getAbsolutePath(), name);

}

}

}

}

}

**Recursion\_Client**

/\*\*

\* @author rich

\* @version 02/10/2017

\* The main class test the recursion methods using JOptionPane box dialog.

\*/

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

import javax.swing.JOptionPane;

public class Recursion\_Client {

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

boolean done = false;

int val = 0, val2 = 0; // value variable to test whether input is valid

while (!done)

{

boolean valid = false;

String choice = JOptionPane.showInputDialog(null,"Please enter an Option by selecting a number 1 to 5\n"

+"1. LogBase2\n"

+"2. Multiply\n"

+"3. Isabel Technique\n"

+"4. Find a file\n"

+"5. Exit");

System.out.println("Menu of choice by selecting a number 1 to 5\n"

+"1. LogBase2\n"

+"2. Multiply\n"

+"3. Isabel Technique\n"

+"4. Find a file\n"

+"5. Exit\n"

+ "your selection:" + choice);

switch (choice) {

case "1":

{

System.out.println("Finding LogBase2");

String argument = "";

while(!valid)

{

argument = JOptionPane.showInputDialog(null,"Please enter an intenger as the argument log base 2");

try

{

val = Integer.parseInt(argument);

valid = true;

}

catch(NumberFormatException nno)

{

JOptionPane.showMessageDialog(null, "Invalid input. Try again");

}

}

System.out.println("Log\_2(" + argument + ") returned " + Recursion.log2(val));

break;

}

case "2":

{

System.out.println("Finding product of two positive integers");

String num1 = "", num2 = "";

while(!valid)

{

num1 = JOptionPane.showInputDialog(null,"Please enter a positive intenger as the multiplicand");

num2 = JOptionPane.showInputDialog(null,"Please enter another positive intenger as the multiplier");

try

{

val = Integer.parseInt(num1);

val2 = Integer.parseInt(num2);

valid = true;

}

catch(NumberFormatException nno)

{

JOptionPane.showMessageDialog(null, "Invalid input. Try again");

}

}

System.out.println(val + " X " + val2 + " = " + Recursion.prodOf2PositiveInt(val, val2));

break;

}

case "3":

{

int[] array;

System.out.println("Testing Isabel's method for summing an array");

File f= new File("C:\\");

while(!valid)

{

String filepath = JOptionPane.showInputDialog(null,"Please enter the path of the File");

String fileName = JOptionPane.showInputDialog(null,"Please enter the name of the File");

f = new File(filepath,fileName);

if(f.exists())

valid = true;

else

{

System.out.println("Not a valid path. Enter a valid path");

}

} Scanner scanFile = new Scanner(f);

scanFile.useDelimiter(",");

int counter = 0;

while(scanFile.hasNext()){

if(scanFile.hasNextInt())

counter++; // only icnrement when an int is found

scanFile.next();

} //Sets scanner back to the beginning of file

scanFile = new Scanner(f);

scanFile.useDelimiter(",");

array = new int[counter]; // create an array with the right size for all the int

for(int i=0; scanFile.hasNext();i++){

while(!scanFile.hasNextInt() && scanFile.hasNext())

scanFile.next();

if(scanFile.hasNextInt()){

array[i]=scanFile.nextInt();

}

} System.out.println("sum of array " + printArray(array) + "is = " + Recursion.isabelTech4ArrSum(array));

break;

}

case "4":

{

System.out.println("Testing File finder");

String filePath = JOptionPane.showInputDialog(null,"Please enter the path of the File");

String fileName = JOptionPane.showInputDialog(null,"Please enter the name of the File");

Recursion.match = false;

Recursion.find(filePath, fileName);

if (!Recursion.match)

System.out.println("File " + fileName + " Not found");

break;

}

case "5":

done = true;

break;

default:

System.out.println("Not a valid choice");

break;

}

}

}

public static String printArray(int[] A)

{

String arrayContent = "[";

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

{

arrayContent += A[i] + " ";

}

return arrayContent + "]";

}

}

**Outpout**

run:

Menu of choice by selecting a number 1 to 5

1. LogBase2

2. Multiply

3. Isabel Technique

4. Find a file

5. Exit

your selection:1

Finding LogBase2

Log\_2(8) returned 3

Menu of choice by selecting a number 1 to 5

1. LogBase2

2. Multiply

3. Isabel Technique

4. Find a file

5. Exit

your selection:2

Finding product of two positive integers

17 X 24 = 408

Menu of choice by selecting a number 1 to 5

1. LogBase2

2. Multiply

3. Isabel Technique

4. Find a file

5. Exit

your selection:3

Testing Isabel's method for summing an array

sum of array is [3 1 5 10] = 19

Menu of choice by selecting a number 1 to 5

1. LogBase2

2. Multiply

3. Isabel Technique

4. Find a file

5. Exit

your selection:4

Testing File finder

File Found at: C:\Users\Rich\Desktop\array.txt

Menu of choice by selecting a number 1 to 5

1. LogBase2

2. Multiply

3. Isabel Technique

4. Find a file

5. Exit

your selection:5

BUILD SUCCESSFUL (total time: 1 minute 51 seconds)