**Recursion Class**

import java.io.File;

import java.io.FileNotFoundException;

/\*\*

\*

\* @author Marie Larson

\* @version 2/12/2018

\*/

public class Recursion {

public static double var(int i){

if(i==1){

return 1;

}

else{

return (double)(1.0/i)+var(i-1);

}

}

public static int IsabelTechnique(int[] Arr) throws Array2{

if(!Base2(Arr.length)){

throw new Array2("The Array is not base 2.");

}

int[] Brr = new int[Arr.length/2];

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

Brr[i] = (Arr[2\*i]+Arr[2\*i+1]);

}

if(Brr.length==1){

return Brr[0];

}

return IsabelTechnique(Brr);

}

public static void List(File path){

System.out.println("\nStart of Path Recursion File Folder" + path.getAbsoluteFile());

if(path.isDirectory()){

String[] contents = path.list();

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

File test = new File(path + File.separator + contents[i]);

if(test.isDirectory()){

List(test);

}

else{

System.out.println(test.getAbsolutePath());

}

}

}

if(path.isFile()){

System.out.println(path.getAbsolutePath());

}

}

private static boolean Base2(int length){

int temp = 2;

while(temp <= length){

if(temp == length){

return true;

}

temp \*= 2;

}

return false;

}

}

**Recursion\_Client Class**

import java.util.Scanner;

import javax.swing.JOptionPane;

//import java.util.logging.Logger;

import java.io.File;

//import java.util.logging.Level;

import java.io.FileNotFoundException;

/\*\*

\*

\* @author Marie Larson

\* @version 2/12/2018

\*/

public class Recursion\_Client {

public static void main(String() args){

if(showconfirmDialog("Starting lab104")==0){

boolean run = true;

while(run){

run = options();

}

}

System.out.println("End");

}

private static void callVar(){

String response = JOptionPane.showInputDialog( null, "Enter an integer", "Var Method", 0);

System.out.println("run var method.user input = " + response);

double result = Recursion.var(Integer.parseInt(response));

showMessageDialog("result of var method " + result);

}

private static void callIsabel(){

String response = JOptionPane.showInputDialog( null, "Enter path to file of integers", "Isabel's technique", 0);

int[] temp = getArrayForFile(response);

if(!(temp==null)){

System.out.println("Run Isabel's technique. user input = " +response);

try {

int result = Recursion.IsabelTechnique(temp);

showMessageDialog("Result of var method " +result);

}

catch (Array2 e){

System.out.println("Array is not of base 2");

showMessageDialog("The data in the file is not base 2.\n"

+ "Check file");

}

}

}

public static void showMessageDialog(String response){

JOptionPane.showMessageDialog(null, response, response, 0);

}

public static int showconfirmDialog(String response){

System.out.println(response);

return JOptionPane.showConfirmDialog(null, response, response, 0);

}

private static boolean options(){

String response;

String optionString = "Arr) run var method\n"

+ "Brr) to run Isabel's method\n"

+ "Crr) to run the list method\n"

+ "Drr) to exit program";

response = JOptionPane.showInputDialog(null, optionString, "Recursive Class", 0);

switch (response){

case "arr":

case "Arr":

System.out.println("Calling var method");

callVar();

break;

case "brr":

case "Brr":

System.out.println("Calling Isabel's Technique\n");

callIsabel();

break;

case "crr":

case "Crr":

System.out.println("Call method 3");

callList();

break;

case "drr":

case "Drr":

System.out.println("Setting flag to exit loop");

if(showconfirmDialog("Would you like to exit the program?")==0){

return false;

}

break;

default :

showMessageDialog("Enter one of the following options: 'Arr', 'Brr', 'Crr', or 'Drr'\n");

System.out.println("Enter one of the following options: 'Arr', 'Brr', 'Crr', or 'Drr'\n");

break;

}

return true;

}

private static int[] getArrayForFile(String response){

File file = new File(response);

Scanner in;

try{

in = new Scanner(file);

}

catch(FileNotFoundException e){

System.out.println("File not found");

showMessageDialog("ERROR: File not found.\n");

return null;

}

ArrayBag Bag = new ArrayBag();

while(in.hasNext()){

if(in.hasNextInt()){

Bag.add(in.nextInt());

}

else{

in.next();

}

}

int[] result = new int[Bag.getSize()];

for(int i=0;i<Bag.getSize();i++){

result[i] = (int)Bag.get(i);

}

return result;

}

private static void callList(){

String response = JOptionPane.showInputDialog(null, "Enter the path to the file folder", "List method", 0);

File file = new File(response);

System.out.println("Run list method. user input = " + response);

Recursion.List(file);

//Sysem.out.println("")

}

}

**Output:**

run:

Begin

Starting lab104

Calling var method

run var method.user input = 5

Calling Isabel's Technique

File not found

Call method 3

Run list method. user input = C:\Windows\DigitalLocker

Start of Path Recursion File FolderC:\Windows\DigitalLocker

Start of Path Recursion File FolderC:\Windows\DigitalLocker\en-US

Setting flag to exit loop

Would you like to exit the program?

End

BUILD SUCCESSFUL (total time: 1 minute 35 seconds)