File handeling in java

import java.io.File;

class Abh {

public static void main(String[] args)

{

// File name specified

File obj = new File("myfile.txt");

System.out.println("File Created!");

}

}

// Import the FileWriter class

import java.io.FileWriter;

// Import the IOException class for handling errors

import java.io.IOException;

public class Abh {

public static void main(String[] args)

{

try {

FileWriter Writer

= new FileWriter("myfile.txt");

Writer.write(

"Files in Java are seriously good!!");

Writer.close();

System.out.println("Successfully written.");

}

catch (IOException e) {

System.out.println("An error has occurred.");

e.printStackTrace();

}

}

}

// Import the File class

import java.io.File;

public class Abh {

public static void main(String[] args)

{

File Obj = new File("myfile.txt");

if (Obj.delete()) {

System.out.println("The deleted file is : "

+ myObj.getName());

}

else {

System.out.println(

"Failed in deleting the file.");

}

}

}

LIS

LIS

class LIS {

/\* lis() returns the length of the longest

increasing subsequence in arr[] of size n \*/

static int lis(int arr[], int n)

{

int lis[] = new int[n];

int i, j, max = 0;

/\* Initialize LIS values for all indexes \*/

for (i = 0; i < n; i++)

lis[i] = 1;

/\* Compute optimized LIS values in

bottom up manner \*/

for (i = 1; i < n; i++)

for (j = 0; j < i; j++)

if (arr[i] > arr[j] && lis[i] < lis[j] + 1)

lis[i] = lis[j] + 1;

/\* Pick maximum of all LIS values \*/

for (i = 0; i < n; i++)

if (max < lis[i])

max = lis[i];

return max;

}

public static void main(String args[])

{

int arr[] = { 10, 22, 9, 33, 21, 50, 41, 60 };

int n = arr.length;

System.out.println("Length of lis is " + lis(arr, n)

+ "\n");

}

}

Diamond

interface DemoInterface1

{

public default void display()

{

System.out.println("the display() method of DemoInterface1 invoked");

}

}

interface DemoInterface2

{

public default void display()

{

System.out.println("the display() method of DemoInterface2 invoked");

}

}

public class DemoClass implements DemoInterface1, DemoInterface2

{

public void display()

{

DemoInterface1.super.display();

DemoInterface2.super.display();

}

public static void main(String args[])

{

DemoClass obj = new DemoClass();

obj.display();

}

}

Throws

public class Main {

static void checkAge(int age) throws ArithmeticException {

if (age < 18) {

throw new ArithmeticException("Access denied - You must be at least 18 years old.");

}

else {

System.out.println("Access granted - You are old enough!");

}

}

public static void main(String[] args) {

checkAge(15); // Set age to 15 (which is below 18...)

}

}

// Java program to demonstrate the difference

// between wait and sleep

class Abh{

private static Object LOCK = new Object();

public static void main(String[] args)

throws InterruptedException {

Thread.sleep(1000);

System.out.println("Thread '" + Thread.currentThread().getName() +

"' is woken after sleeping for 1 second");

synchronized (LOCK)

{

LOCK.wait(1000);

System.out.println("Object '" + LOCK + "' is woken after" +

" waiting for 1 second");

}

}

}