Java Essentials Module 4 Assignment

Please use the jar file to test the program.

Command: java -jar Module4Assignment.jar

Notes:

* Out of bounds exercise ran first to demonstrate program can continue
* Package inclusion can be seen at top of Module4Main class
* Inserting and deleting elements in queue interface performed without user interaction
* Second thread output highlighted in output
* As main thread has higher priority than second thread it cannot be guaranteed it will be shown as concurrent output. Please run program again if this occurs

Module4Main Class (contains main()):

**import** maths.functions.MathsFunctions; (shows package inclusion)

**public** **class** Module4Main **extends** Thread{

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

// **TODO** Auto-generated method stub

// Out of Bounds exception exercise

OutOfBounds outOfBounds = **new** OutOfBounds();

outOfBounds.arrayDetails();

outOfBounds.testArray(1);

outOfBounds.testArray(6);

// Maths functions package exercise

MathsFunctions mathsFunctions = **new** MathsFunctions();

mathsFunctions.subtract();

mathsFunctions.multiply();

mathsFunctions.divide();

mathsFunctions.factorial();

System.***out***.println("\nQueue Interface Exercise");

QueueInterface qInterface = **new** QueueInterface();

qInterface.insert(1);

qInterface.insert(2);

qInterface.insert(3);

qInterface.delete();

qInterface.delete();

qInterface.delete();

System.***out***.println("\nMultithread Exercise");

ThreadTimes5 threadtimes5 = **new** ThreadTimes5();

threadtimes5.start();

Thread threadEvens = Thread.*currentThread*();

**for** (**int** i = 2; i <= 40; i = i + 2) {

System.***out***.println(i);

}

}

}

# Maths Functions Class in math.functions Package:

* Implements maths functions subtract, multiply, divide and factorial
* Contained in package math.functions

**package** maths.functions;

**import** java.util.Scanner;

**public** **class** MathsFunctions {

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

**public** **void** subtract() {

System.***out***.println("\nSubtraction: ");

System.***out***.println("Please enter first number: ");

**float** firstNum = mathInput.nextFloat();

mathInput.nextLine();

System.***out***.println("Please enter second number: ");

**float** secondNum = mathInput.nextFloat();

mathInput.nextLine();

**float** result = firstNum - secondNum;

System.***out***.println(firstNum +" - " + secondNum +" = " + result);

}

**public** **void** multiply() {

System.***out***.println("\nMultiplication: ");

System.***out***.println("Please enter first number: ");

**float** firstNum = mathInput.nextFloat();

mathInput.nextLine();

System.***out***.println("Please enter second number: ");

**float** secondNum = mathInput.nextFloat();

mathInput.nextLine();

**float** result = firstNum \* secondNum;

System.***out***.println(firstNum +" x " + secondNum +" = " + result);

}

**public** **void** divide() {

System.***out***.println("\nDivision: ");

System.***out***.println("Please enter first number: ");

**float** firstNum = mathInput.nextFloat();

mathInput.nextLine();

System.***out***.println("Please enter second number: ");

**float** secondNum = mathInput.nextFloat();

mathInput.nextLine();

**float** result;

**try** {

result = firstNum / secondNum;

System.***out***.println(firstNum +" / " + secondNum +" = " + result);

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**public** **void** factorial() {

System.***out***.println("\nFactorial: ");

**int** firstNum;

**do** {

System.***out***.println("Please enter integer number more or equal to 0: ");

firstNum = mathInput.nextInt();

mathInput.nextLine();

} **while** (firstNum < 0);

**int** result = 1;

**if** (firstNum == 0)

result = 1;

**else** **for** (**int** i = 1; i <= firstNum; i++) {

result = result \* i;

}

System.***out***.println(firstNum + "! = " + result);

}

}

# OutofBounds Class:

* To demonstrate OutofBounds exception

**public** **class** OutOfBounds {

**int**[] iArray = {1, 2, 3, 4, 5};

**public** **void** arrayDetails() {

System.***out***.println("Array elements: " + iArray);

System.***out***.println("Array size: " + iArray.length);

}

**public** **void** testArray (**int** i) {

**try** {

System.***out***.println("Array element = " + iArray[i]);

} **catch** (ArrayIndexOutOfBoundsException exception) {

// **TODO** Auto-generated catch block

exception.printStackTrace();

}

}

}

# QueueInterface Class:

* For inserting and deleting to and from queue interface

**import** java.util.LinkedList;

**import** java.util.Queue;

**public** **class** QueueInterface {

Queue<Integer> qExample = **new** LinkedList<>();

**public** **void** insert (**int** i) {

qExample.add(i);

System.***out***.println("\n" + qExample);

**int** size = qExample.size();

System.***out***.println("Size of queue = " + size);

}

**public** **void** delete () {

**int** i = qExample.remove();

System.***out***.println("\nDeleted " + i);

System.***out***.println(qExample);

**int** size = qExample.size();

System.***out***.println("Size of queue = " + size);

}

}

# ThreadTimes5 Class:

* Implements thread for 5 times tables

**class** ThreadTimes5 **extends** Thread

{

**public** **void** run()

{

**for** (**int** i =0; i <= 20; i++) {

System.***out***.println(i + " x 5 = " + (i \* 5));

}

}

}

# Output:

C:\Users\admin\Desktop\Courses\Java\Module4Assignment>java -jar Module4Assignment.jar

Array elements: [I@5cad8086

Array size: 5

Array element = 2

java.lang.ArrayIndexOutOfBoundsException: 6

at OutOfBounds.testArray(OutOfBounds.java:18)

at Module4Main.main(Module4Main.java:14)

at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)

at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)

at java.lang.reflect.Method.invoke(Unknown Source)

at org.eclipse.jdt.internal.jarinjarloader.JarRsrcLoader.main(JarRsrcLoader.java:61)

Subtraction:

Please enter first number:

4

Please enter second number:

8

4.0 - 8.0 = -4.0

Multiplication:

Please enter first number:

3

Please enter second number:

6

3.0 x 6.0 = 18.0

Division:

Please enter first number:

4

Please enter second number:

8

4.0 / 8.0 = 0.5

Factorial:

Please enter integer number more or equal to 0:

5

5! = 120

Queue Interface Exercise

[1]

Size of queue = 1

[1, 2]

Size of queue = 2

[1, 2, 3]

Size of queue = 3

Deleted 1

[2, 3]

Size of queue = 2

Deleted 2

[3]

Size of queue = 1

Deleted 3

[]

Size of queue = 0

Multithread Exercise

2

4

6

8

10

12

14

16

0 x 5 = 0

1 x 5 = 5

2 x 5 = 10

18

20

22

3 x 5 = 15

24

26

28

30

32

34

36

38

40

4 x 5 = 20

5 x 5 = 25

6 x 5 = 30

7 x 5 = 35

8 x 5 = 40

9 x 5 = 45

10 x 5 = 50

11 x 5 = 55

12 x 5 = 60

13 x 5 = 65

14 x 5 = 70

15 x 5 = 75

16 x 5 = 80

17 x 5 = 85

18 x 5 = 90

19 x 5 = 95

20 x 5 = 100