Example on exception java113

من الممكن يجي سؤال صممي اكسيبشن مثلا اسمه

negativExcption

ارمي منه اكسيبشن اذا قرأنا رقم سالب

راح تصممي كلاس اكسيبشن بسيط

class negativExcption  extends Exception

{

 Public negativExcption ()

{

//=============================print

}

}

 Public class test

{

main()

{

try

{

 Int num = consol.nextInt() ;

if(num < 0 )

throw new negativExcption () ;

}

catch(negativExcption  e )

{

 S.o.p("negativ number found ") ;

 }

 }

هذ الحل في ابسط صورة

 =======================================================================

Trace :

1 import java.util.\* ;   
 2   
 3 class ex  
 4 {  
 5 public static void f()throws Exception   
 6 {  
 7 try  
 8 {  
 9 System.out.println("k");  
10 throw new Exception();  
11 }  
12 catch(InputMismatchException e)  
13 {  
14 System.out.println("A");  
15 }  
16 catch(Exception e)  
17 {  
18 System.out.println("B");  
19 throw e ;   
20 }  
21 finally  
22 {  
23 System.out.println("final method f");  
24 }  
25   
26 }  
27   
28 public static void m()throws Exception   
29 {  
30 f() ;   
31 }  
32 public static void main(String args[])  
33 {  
34   
35 try  
36 {  
37 m() ;  
38 System.out.println("main");  
39   
40 }  
41 catch(Exception e)  
42 {  
43 System.out.println("ex in main");  
44 }  
45 finally  
46 {  
47 System.out.println("final main");  
48   
49 }  
50 }  
51   
52   
53 }

k

B

final method f

ex in main

final main

------------

التالي مهم

=================================

public class ExceptionClass  
{  
  public static void methodC(int a) throws Exception  
  {  
     if ( a == 1)  
            throw new MiniException(a);  
     else  
            throw new Exception(“Exception number” + a);  
  }

  public static void methodB(int a) throws Exception  
  {     methodC(a); }

  public static void methodA(int a) throws Exception  
  {  
     try{   methodB(a);    }  
     catch (Exception e)

     {    System.out.println("Inside methodA Exception");}  
  }

 public static void main ( String[] args ) throws Exception  
  {  
     try  
     {  
         methodA(1);  
         methodB(2);  
         methodC(1);  
      }  
     catch (MiniException me)  
     {   System.out.println("Inside main MiniException"); }  
     finally  
     {   System.out.println("finally block");  }           
     System.out.print("Done");    
  
  }  
}

**output:**

----Hit any key to start.

Inside methodA Exception

finally block

Exception in thread "main" java.lang.Exception: Exception number2

        at ExceptionClass.methodC(ExceptionClass.java:16)

        at ExceptionClass.methodB(ExceptionClass.java:20)

        at ExceptionClass.main(ExceptionClass.java:35)

 ----Hit any key to continue.

====================================================

Suppose you have the following program

1   
 2 class test   
 3 {  
 4 public  static void main(String arg[])  
 5 {  
 6 try{  
 7 Method() ;   
 8 System.out.println("after the method call") ;   
 9 }  
10 catch(Exception ex)  
11 {  
12 System.out.println("exeption in main ") ;   
13 }  
14 System.out.println("End ") ;   
15 }  
16 static void Method() throws Exception   
17 {  
18 try{  
19 String s = "abc" ;   
20 System.out.println(s.charAt(3)) ;   
21 }  
22 catch(Exception  ex)  
23 {  
24 System.out.println("RunTimeException   exeption in method ") ;   
25 }  
26 }  
27 }

1-

Will the statement in line  8   execute ? why ?  yes

Will the statement in line  12   execute ? why ?  no

Will the statement in line  14   execute ? why ?  yes

=================================================================

2.        Trace the following program:

public class MyException extends Exception  
{ public MyException ()  
  {  super ("MyException thrown!");  
      System.out.print("Immediate attention required!");

  }

}//end of class MyException

public class MyExceptionTestProg  
{    
 public static void main(String[] args) throws  MyException  
 {      
  try  
  {  throw new MyException();   }  
  catch(MyException e) { }    
   } //end of main     
} //end of class

*Answer*

*Tutorial 2 .*

*Immediate attention required!*

==============================================

**Sheet**

**Chapter 12: Exceptions**

**Tutorial**

**1.** Trace the following programs:

a**. public class UsingExceptions{  
 3 public static void main(String args[]) {  
 4 try {  
 5 throwException() : }  
 6 catch ( Exception exception) {  
 7 System.out.println( “ exception handled in main” ):  
 8 }  
 9 doesNotThrowException();  
10 }  
11 Public static void throwException() throws Exception {  
12 try {  
13 System.out.println( “ method throwException”);  
14 throw new Exception(); }  
15 catch (Exception exception ) {  
16 System.out.println( “ Exception handled in mthd “ +   
17 “throwException” );  
18 Throw exception; }  
19 finally {  
20 System.out.println( “ finally executed in throwException”); } }  
21   
22 public static void doesNotThrow Exception() {  
23 try {   
24 System.out.println(“mthd doesNotThrowException”); }  
25 Catch ( Exception exception) {  
26 System.out.println( exception) ; }  
27 finally {  
28 System.out.println( “ finally executed in” +   
29 “doesNotThrowException” ) ; }  
30 System.out.println ( “ End of method doesNotThrowException”); }}  
31   
32 =====================================================================================  
33 b. public class X{  
34 public sstatic void main ( String args[] ) {  
35 try { throwException(); }  
36 catch ( Exception exception) {   
37 System.out.println( “ main exception handler”); }  
38 public static void throwException() throws Exception {  
39 try {   
40 System.out.println(“method throwException”);  
41 throw new Exception(); }  
42 catch (RuntimeException r ) {  
43 System.out.println( “ Exception handler in throwException”);}  
44 finally {  
45 System.out.println( “ Finally is always executed”);}}}  
46**

**==================================================================================  
47 c. public class C{  
48 public static void main( String args[]) {  
49 try { method1(); }  
50 catch (Exception exception ) { exception.printStackTrace(); }}  
51 public static void method1() throws Exception {  
52 try { method2(); }  
53 catch (Exception exception ) {   
54 throw new Exception(“Exception thrown in method1”); }}  
55 public static void method2() throws Exception {  
56 try { method3(); }  
57 catch ( Exception exception) {  
58 throw new Exception(“Exception thrown in method2”); }}  
59 public static void method3() throws Exception {  
60 throw new Exception(“Exception thrown in method3”);}}  
61  ==================================================================================================**

**Programming assignment**

 Write a program that includes an exception superclass SuperExc and a subclass Subexc.  Include a client program that has a main that calls methodA, which in turn calls methodB.  The latter throws a Subex exception that is handled in methodA which rethrows  a SuperExc that is handled in main.

--------------------------------

الحلول للشيت السابق :

**Sheet**

**Chapter 12: Exceptions**

**Tutorial**

**1.** Trace the following programs:

*Output*:

A -

method throwException  
 Exception handled in mthd throwException  
 finally executed in throwException  
 exception handled in main  
mthd doesNotThrowException  
 finally executed indoesNotThrowException  
 End of method doesNotThrowException  
  
-------------

b-

method throwException  
 Finally is always executed  
 main exception handler  
-------------

c-

java.lang.Exception: Exception thrown in method1  
    at C.method1(C.java:8)  
    at C.main(C.java:3)

---------------------

Program :

class superexe extends Exception  
{  
public superexe()   
{  
super(" in superexe " ) ;   
}  
public superexe( String msg)  
{  
super(msg) ;   
}  
}  
  
class subExc extends superexe   
{  
public subExc()  
{  
super("in subExc") ;   
}  
public subExc(String msg)  
{  
super(msg) ;   
}  
  
}

----------

public class clint\_test  
{  
public static void main(String arg[])  
{  
try{  methodA() ; }  
catch(superexe e) { e.toString() ;  
e.printStackTrace(); }  
  
}//end main   
  
static void methodA() throws superexe   
{  
try  
{  
methodB() ;   
}  catch(subExc  ss)  
{  
throw ss ;   
}  
}  
static void methodB()throws subExc   
{  
throw new subExc(" in method b " ) ;   
}  
}// end class

output

superexe:  in method b   
    at clint\_test.methodA(clint\_test.java:18)  
    at clint\_test.main(clint\_test.java:5)

============================

**a**Trace the following program:

public class Excp{

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

     String s ="abc";

   int k= 0; int j;

try {

if (k == 0) j = 10/k;

    } catch (ArithmeticException ex) {

System.out.println("arithmetic error");

      throw ex;

    } catch (Exception ex) {

System.out.println("Error");

    } finally {

System.out.println("in finally clause");

    }

System.out.println("after try block");

System.out.println(s);

}}

|  |
| --- |
| arithmetic error  in finally clause  Exception in thread "main" java.lang.ArithmeticException: / by zero          at Excp.main(Excp.java:7) |

--------------------------------------------------------------

Trace

16 public class midExceptionTrace  
17 {  
18 public static void main(String args[])throws Exception3  
19 {  
20 try  
21 {  
22 m1() ;   
23 m2() ;   
24 m3() ;   
25 }  
26 catch(Exception2 ex)  
27 {  
28 System.out.println("inSide Exception2 in main " ) ;  
29 }  
30 catch(Exception3 ex)  
31 {  
32 System.out.println("inSide Exception3 in main " ) ;  
33 throw ex ;   
34 }  
35 catch(Exception ex)  
36 {  
37 System.out.println("inSide Exception in main " ) ;   
38 }  
39 finally  
40 {  
41 System.out.println("finally") ;   
42 }  
43 System.out.println("done") ;   
44 }  
45 public static void m1()throws Exception3 , Exception2  
46 {  
47 try  
48 {  
49 m2() ;   
50 throw new Exception1() ;   
51 }  
52 catch(Exception1 ex1)  
53 {  
54 System.out.println("catch Exception1 in m1") ;   
55 }  
56 /\* catch(Exception2 ex2)  
57 {  
58 System.out.println("catch Exception2 in m1") ;   
59 }\*/  
60 }  
61   
62 public static void m2()throws Exception3 , Exception2  
63 {  
64 try  
65 {  
66 m3() ;   
67 throw new Exception2() ;   
68 }  
69 catch(Exception3 ex1)  
70 {  
71 System.out.println("catch Exception3 in m2") ;   
72 }  
73 }  
74   
75 public static void m3() throws Exception3  
76 {  
77 throw new Exception3() ;   
78 }  
79   
80 }// end class   
81

Output :

catch Exception3 in m2

catch Exception1 in m1

catch Exception3 in m2

inSide Exception3 in main

finally

Exception in thread "main" Exception3

=======================================================

**Question 1: [3 marks] What is the output of the following Java code?**

|  |
| --- |
| **Output**  **1 ½ pt**  **2 ½ pt**  **4 ½ pt**  **5 ½ pt**  **Exception in method f1() caught in main 1 pt** |

**public class Quiz2 {  
public static void f1() throws Exception {  
int a=100, b=200;  
System.out.println("1");  
try {  
 System.out.println("2");  
 f2(a,b);  
 System.out.println("3");  
}  
catch (Exception e) {   
 System.out.println("4");   
 throw e;   
}  
finally {   
 System.out.println("5");   
}  
  
System.out.println("6");  
}  
  
//-------------------------------------------------------  
public static void f2 (int x, int y) throws Exception {  
 if (x<y) throw new Exception();  
}  
//-------------------------------------------------------  
public static void main(String s[]) {   
try {  
 f1();  
}  
catch (Exception e){**

**System.out.println ("Exception in method f1()caught in main");}  
}}  
Question 2: [7 marks]** 10 students in a class deserve a bonus grade. You are given a text file **“Student\_Data.txt”**, where each line in the file includes the student’s first name, last name, current grade and bonus grade. For example the first line of the file will look like this:

**Sara Ali 73.5 2**

Assume you have the following class declaration

**public class Student implements Serializable {**

**public String firstName, lastName;**

**public double grade; }**

**Complete the program below such that it will do the following:**

1. Using a **Scanner** object, read the data of each student from the file **“Student\_Data.txt”**, and store it in an object of type **Student**. The **grade** of the Student that you will store in the object should be the new **grade** (after adding the bonus).
2. Store each **Student** object in a binary object file called **“Student\_File.obj”**.
3. Declare an array of 10 students called **studArray**. Read the data of the students from the object file **“Student\_File.obj”**, and store it in the array.

|  |
| --- |
| import java.util.\*; import java.io.\*;  class Student implements Serializable{ public String firstName, lastName; public double grade;  } public class Quiz2FileTest{ public static void main(String [] args) throws IOException, **¼ pt** ClassNotFoundException **½ pt** {  Scanner scanner= new Scanner (new File("Student\_Data.txt")); **½ pt**  File studFile= new File("Student\_File.obj"); **½ pt**  FileOutputStream outFileStream = new FileOutputStream (studFile);**½ pt**  ObjectOutputStream outObjStream= new ObjectOutputStream (outFileStream); **½ pt**    for (int i=0; i<10; i++){ **¼ pt**  Student obj= new Student(); **¼ pt**  obj.firstName= scanner.next(); **¼ pt**  obj.lastName= scanner.next(); **¼ pt**  obj.grade= scanner.nextDouble()+scanner.nextDouble(); **½ pt**  outObjStream.writeObject(obj); **½ pt** }  Student [] studArray= new Student[3];  FileInputStream inFileStream = new FileInputStream (studFile); **½ pt**  ObjectInputStream inObjStream= new ObjectInputStream (inFileStream); **½ pt**  for (int i=0; i<10; i++){ **¼ pt**  studArray[i]= (Student) inObjStream.readObject(); **1 pt**   }  }  } |