Exceptions

- تسمح لغة جافا بمعالجة الأخطاء والاستثناءات وذلك بالتقاط الخطأ الذي سببته إحدى التعليمات فيتم رمي غرض من نوع هذا الخطأ ومعالجتة بالشكل المناسب ومتابعة البرنامج بشكل طبيعي
- الية المعالجة تتم من خلال استخدام التعليمات try {} catch(){} finally{}
- حيث نضع ضمن كتلة try التعليمات التي نتوقع أن ترمي استثناء ويتم التقاط الخطأ من خلال العبارة catch حيث يحدد وسيطها نوع الخطأ الذي تم رميه وضمن كتلتها يتم كتابة ألية المعالجة التي نريدها لمعالجة الخطأ
- يجب أن يكون نوع الاستثناء الذي يتم رميه من كتلة try هو نفس نوع الوسيط الموجود ضمن عبارة catch او من صف ابن له

مثال:

اكتب برنامج يقوم بالقسمة على صفر.

```
public class Test
{

public static void main(String args[])
{
int d = 0;
int a = 10 / d;
}
}
```

بعد تنفيذ البرنامج السابق نلاحظ أنه ستظهر رسالة الخطأ التالية:

```
java.lang.ArithmeticException: / by zero

at Test. main(Test.java:6
```

بينما عند كتابة كتلةcatch, try معالجة مناسبة كما في الكود التالي:

```
public class Test2
{
public static void main(String args[]) {
int d, a;
try // monitor a block of code.
{
d = 0;
a = 42 / d;
System.out.println("This will not be printed.");
}
catch (ArithmeticException  e) {         // catch divide-by-zero error
System.out.println("Division by zero.");
}
System.out.println("After catch statement.");
}
}
```

النتيجة تكون:

```
Division by zero.
After catch statement.
```

عبارة Finally

وهي تعليمة اختيارية تعرف بعد كتلة معالجة الاستثناء Try {} ctach(){}

حيث انها اذا كتبت تنفذ دوما" أي أننا نستخدمها عندما نريد تنفيذ تعليمات معينة سواء رمي الاستثناء أو لا

```
public class HelloWorld
{
public static void main ( String[] args)
{
Int i=0 ;
String greeting[] = { "Hello World!" ," No, I mean it!" , "HELLO
WORLD!!"};
while (i<4)
{
try {
System.out.println(greeting[i]);
i++;
```

```
} catch (ArrayIndexOutOfBoundsException   e) {
System.out.println("Re-setting Index Value");
i= 0;
} finally {
System.out.println("This always printed");
}
}}
```

الخرج:

Hello World!
This always printed
No, I mean it!
This always printed
HELLO WORLD!!
This always printed
Re-setting Index Value
This always printed

الواجهات المرئية:تستخدم لاظهار نوافذ مرئية نستخدمها لإدخال قيم ولأخراج قيم

فمثلا في المثال التالي نستخدم من الصف JOptionPane swingالموجود في الحزمةjavax الدالة showInput Dialog()_ وهي دالة تظهر صندوق حوارلإدخال قيمة محرفية أما الدالة showMessageDialog()_ فهي دالة تظهر رسالة

مثال: برنامج لإدخال عددين صحيحين وحساب مجموعهما وطباعته.

```
import javax.swing.JOptionPane;
public class Test3
{
public static void main( String args[] )
{
  String s1; String s2;
int n1; int n2;
s1=JOptionPane.showInputDialog("enter a first number");
s2=JOptionPane.showInputDialog("enter a second
number");
n1=Integer.parseInt(s1);
n2=Integer.parseInt(s2);
```

```
int n=n1+n2;
JOptionPane.showMessageDialog(null,"the sum="+n,"the
result",JOptionpane.INFORMATION_MESSAGE);
} //  end main
}
```

انتهت المحاضرة