Java自定义异常

如果 [Java](http://c.biancheng.net/java/) 提供的内置异常类型不能满足程序设计的需求，可以设计自己的异常类型。自定义异常类必须继承现有的 Exception 类或 Exception 的子类来创建，其语法形式为：

<class><自定义异常名><extends><Exception>

在编码规范上，一般将自定义异常类的类名命名为 XXXException，其中 XXX 用来代表该异常的作用。  
  
自定义异常类一般包含两个构造方法：一个是无参的默认构造方法，另一个构造方法以字符串的形式接收一个定制的异常消息，并将该消息传递给超类的构造方法。  
  
例如，以下代码创建一个名称为 IntegerRangeException 的自定义异常类：

1. **class** IntegerRangeException extends Exception
2. {
3. **public** IntegerRangeException()
4. {
5. super();
6. }
7. **public** IntegerRangeException(String s)
8. {
9. super(s);
10. }
11. }

以上代码创建的自定义异常类 IntegerRangeException 类继承自 Exception 类，在该类中包含两个构造方法。

#### 例 1

编写一个程序，对会员注册时的年龄进行验证，即检测是否在 0~100 岁。  
  
(1) 这里创建了一个异常类 MyException，并提供两个构造方法。MyException 类的实现代码如下：

1. **public** **class** MyException extends Exception
2. {
3. **public** MyException()
4. {
5. super();
6. }
7. **public** MyException(String str)
8. {
9. super(str);
10. }
11. }

(2) 接着创建测试类，调用自定义异常类。代码实现如下：

1. import java.util.InputMismatchException;
2. import java.util.Scanner;
3. **public** **class** Test07
4. {
5. **public** **static** void main(String[] args)
6. {
7. int age;
8. Scanner input=**new** Scanner(System.in);
9. System.out.println("请输入您的年龄：");
10. **try**
11. {
12. age=input.nextInt(); //获取年龄
13. **if**(age<0)
14. {
15. **throw** **new** MyException("您输入的年龄为负数！输入有误！");
16. }
17. **else** **if**(age>100)
18. {
19. **throw** **new** MyException("您输入的年龄大于100！输入有误！");
20. }
21. **else**
22. {
23. System.out.println("您的年龄为："+age);
24. }
25. }
26. **catch**(InputMismatchException e1)
27. {
28. System.out.println("输入的年龄不是数字！");
29. }
30. **catch**(MyException e2)
31. {
32. System.out.println(e2.getMessage());
33. }
34. }
35. }

(3) 运行该程序，当用户输入的年龄为负数时，则拋出 MyException 自定义异常，执行第二个 catch 语句块中的代码，打印出异常信息。程序的运行结果如下所示。

请输入您的年龄：

-2

您输入的年龄为负数！输入有误！

当用户输入的年龄大于 100 时，也会拋出 MyException 自定义异常，同样会执行第二个 catch 语句块中的代码，打印出异常信息，如下所示。

请输入您的年龄：

110

您输入的年龄大于100！输入有误！

在该程序的主方法中，使用了 if…else if…else 语句结构判断用户输入的年龄是否为负数和大于 100 的数，如果是，则拋出自定义异常 MyException，调用自定义异常类 MyException 中的含有一个 String 类型的构造方法。在 catch 语句块中捕获该异常，并调用 getMessage() 方法输出异常信息。  
  
提示：因为自定义异常继承自 Exception 类，因此自定义异常类中包含父类所有的属性和方法。