

# 实验八 异常处理

## 一、实验目的

通过编程和上机实验理解Java 语言是如何体现面向对象编程基本思想，掌握面向对象程序设计的方法，掌握异常处理机制和如何自定义异常类型。

## 二、实验器材

装有 JAVA 语言工具软件(Eclipse)的微机若干

## 三、实验内容 练习知识点：

- 自定义异常的步骤：

1. 通过继承 `java.lang.Exception` 类声明自己的异常类； 2. 在方法中生成自定义异常的实例，并用 `throw` 语句抛出；

3. 在方法的声明部分用 `throws` 语句声明该方法可能抛出的异常

1、在程序中要获得一个学生的成绩，此成绩必须在0~100 之间，如果代表成绩的数据小于0 则抛出一个数据太小的异常，如果大于100 则抛出一个数据太大的异常。Java 提供的异常类中不存在描述这些情况的异常，那么只能在程序中自己定义这两个异常。

要求：

(1) 定义异常类 `TooHigh` 和 `TooLow` 分别代表数据太大和数据太小的异常，这两个类继承

`Exception` 类

(2) 定义方法 `public void getScore(int x) throws TooHigh, TooLow` 用来获得成绩

2、在接口实验的基础上增加异常处理：

(1) 对于汽车类的求速度的方法体：`a*b/c`，如果参数 `c` 等于0，抛出一个 `ArithmeticException` 的异常对象，并且传递给该异常对象“除数为0”的信息。如果 `a*b<0`，抛出一个 `Exception` 的异常对象，并且传递给该对象“速度不能为负数”的信息。

(2) 同理，对飞机类的求速度的方法体，也要求对形参做有效性的处理验证，如果 `a+b+c<0`，则抛出一个 `Exception` 的异常对象，并且传递给该对象“速度不能为负数”的信息。

```
interface Common{  
  
    double speed(double a,double b,double c);  
  
}
```

```

class Plane implements Common{

    public double speed(double a,double b,double

        c){ return a+b+c;

    }

}

class Car007 implements Common{

    public double speed(double a,double b,double

        c){ return a*b/c;

    }

}

public class ComputeTime{

    public static void main(String[] args)throws

        Exception{ Car007 car=new Car007();

        double speed=car.speed(30,-12, 0); System.out.println("该交通

        工具的运行速度为"+speed); System.out.println("运行1000 公

        里使用的时间为"+1000/speed);

        }

}

```

#### 四、实验报告要求

1. 整理实验结果。
2. 小结实验心得体会。