

《Java面向对象程序设计》 第6章

内部类与异常类

导读

主要内容

- 内部类
- 匿名类
- 异常类
- 断言
- **Class**类

难点

- 异常类

§ 6.1 内部类

Java支持在一个类中声明另一个类，这样的类称作内部类，而包含内部类的类成为内部类的外嵌类。

例6-1

§ 6.2 匿名类

§ 6.2.1 和类有关的匿名类

Java允许我们直接使用一个类的子类的类体创建一个子类对象。

创建子类对象时，除了使用父类的构造方法外还有类体，此类体被认为是一个子类去掉类声明后的类体，称作匿名类。

假设People是类，那么下列代码就是用People的一个子类（匿名类）创建对象：

```
new People () {  
    匿名类的类体  
};
```

例6-2

§ 6.2.2 和接口有关的匿名类

假设**Comparable**是一个接口，那么，Java允许直接用接口名和一个类体创建一个匿名对象，此类体被认为是实现了**Comparable**接口的类去掉类声明后的类体，称作匿名类。

下列代码就是用实现了**Comparable**接口的类（匿名类）创建对象：

```
new Comparable() {  
    实现接口的匿名类的类体  
}
```

例6-3

§ 6.3 异常类

所谓异常就是程序运行时可能出现一些错误，比如试图打开一个根本不存在的文件等，异常处理将会改变程序的控制流程，让程序有机会对错误作出处理。

§ 6.3.1 try~catch语句

Java使用try~catch语句来处理异常，将可能出现的异常操作放在try~catch语句的try部分，将发生异常后的处理放在catch部分。

try~catch语句的格式如下：

```
try {  
    包含可能发生异常的语句  
}  
catch(ExceptionSubClass1 e) {  
    ...  
}  
catch(ExceptionSubClass2 e) {  
    ...  
}
```

例6-4

§ 6.3.2 自定义异常类

我们也可以扩展Exception类定义自己的异常类，然后规定哪些方法产生这样的异常。一个方法在声明时可以使用**throws关键字**声明要产生的若干个异常，并在该方法的方法体中具体给出产生异常的操作，即用相应的异常类创建对象，并使用**throw关键字**抛出该异常对象，导致该方法结束执行。

例6-5

§ 6.4 断言

断言语句用于调试代码阶段。在调试代码阶段让断言语句发挥作用，这样就可以发现一些致命的错误，当程序正式运行时就可以关闭断言语句，但仍把断言语句保留在源代码中，如果以后应用程又需要调试，可以重新启用断言语句。

使用关键字**assert**声明一条断言语句，断言语句有以下两种格式：

assert booleanExpression;

assert booleanExpression:messageException;

例6-6

§ 6.5 Class类

§ 6.5.1 获取类的有关信息

Class是java.lang包中的类，该类的实例用来封装对象运行时的状态。当一个类被加载且创建对象时，和该类相关的一个类型为Class的对象就会自动创建。

任何对象调用`getClass()`方法都可以获取和该对象相关的一个Class对象，这个Class对象调用如下的方法可以获取创建对象的类的有关信息，比如，类的名字、类中的方法名称、成员变量的名称等等。

`String getName()` 返回类的名字

`Constructor[] getDeclaredConstructors()` 返回类的全部构造方法。

`Field[] getDeclaredFields()` 返回类的全部成员变量。

`Method[] getDeclaredMethods()` 返回类的全部方法。

例6-7

§ 6.5.2 使用Class实例化一个对象

Java通过使用Class可以得到一个类的实例。

◆为了使用Class得到一个类的实例，可以先得到一个和该类相关的Class对象，做到这一点并不困难，只要使用Class的类方法

```
public static Class forName(String className) throws ClassNotFoundException
```

就可以返回一个和参数className指定的类相关的Class对象。

◆再让这个Class对象调用

```
public Object newInstance() throws InstantiationException, IllegalAccessException
```

方法就可以得到一个className类的实例。

例6-8

§ 6.6 小结

1. Java支持在一个类中声明另一个类，这样的类称作内部类，而包含内部类的类成为内部类的外嵌类。
2. 和某类有关的匿名类就是该类的一个子类，该子类没有明显的用类声明来定义，所以称做匿名类。
3. 和某接口有关的匿名类就是实现该接口的一个类，该子类没有明显的用类声明来定义，所以称做匿名类。
4. Java的异常可以出现在方法调用过程中，即在方法调用过程中抛出异常对象，导致程序运行出现异常，并等待处理。Java 使用try~catch语句来处理异常，将可能出现的异常操作放在try~catch语句的try部分，当try部分中的某个方法调用发生异常后，try部分将立刻结束执行，而转向执行相应的catch部分。