

一、枚举类

1、枚举类的使用

2、如何定义枚举类

3、Enum类中的常用方法

4、使用Enum关键字定义的枚举类实现接口的情况

一、枚举类

1、枚举类的使用

1. 枚举类的理解：类的对象只有有限个，确定的。我们称此类为枚举类
2. 当需要定义一组常量时，强烈建议使用枚举类
3. 如果枚举类中只有一个对象，则可以作为单例模式的实现方式。

2、如何定义枚举类

方式一：jdk5.0之前，自定义枚举类

1、枚举类的属性

1.1、枚举类对象的属性不应允许被改动，所以应该使用`private final`修饰

1.2、枚举类的使用`private final`修饰的属性应该在构造器中为其赋值

1.3、若枚举类显式的定义了带参数的构造器，则在列出枚举值时也必须对应的传入参数

2、在类的内部创建枚举类的实例，声明为`public static final`

3、对象如果有实例变量，应该声明为`private final`，并在构造器中初始化

方式二：jdk5.0，可以使用`enum`关键字定义枚举类

1、使用说明

1.1、使用`enum`定义的枚举类默认继承了`java.lang.Enum`类，因此不能再继承其他类。

1.2、枚举类的构造器只能使用`private`权限修饰符

1.3、枚举类的所有实例必须在枚举类中显式列出(, 分隔;结尾), 列出的实例系统会自动添加public static final修饰

1.4、必须在枚举类的第一行声明枚举类对象

2、JDK1.5 中可以在switch表达式中使用Enum定义的枚举类的对象作为表达式, case子句可以直接使用枚举值的名字, 无需添加枚举类作为限定

3、Enum类中的常用方法

1、values()方法: 返回枚举类型的对象数组。该方法可以很方便地遍历所有的枚举值。

2、valueOf(String str): 可以把一个字符串转为对应的枚举类对象。要求字符串必须是枚举类对象的“名字”。如不是, 会有运行时异常: IllegalArgumentException。

3、toString(): 返回当前枚举类对象常量的名称

4、

方法名	详细描述
valueOf	传递枚举类型的 Class 对象和枚举常量名称给静态方法 valueOf, 会得到与参数匹配的枚举常量。
toString	得到当前枚举常量的名称。你可以通过重写这个方法来使得到的结果更易读。
equals	在枚举类型中可以直接使用“==”来比较两个枚举常量是否相等。Enum 提供的这个 equals()方法, 也是直接使用“==”实现的。它的存在是为了在 Set、List 和 Map 中使用。注意, equals()是不可变的。
hashCode	Enum 实现了 hashCode()来和 equals()保持一致。它也是不可变的。
getDeclaringClass	得到枚举常量所属枚举类型的 Class 对象。可以用它来判断两个枚举常量是否属于同一个枚举类型。
name	得到当前枚举常量的名称。建议优先使用 toString()。
ordinal	得到当前枚举常量的次序。
compareTo	枚举类型实现了 Comparable 接口, 这样可以比较两个枚举常量的大小(按照声明的顺序排列)
clone	枚举类型不能被 Clone。为了防止子类实现克隆方法, Enum 实现了一个仅抛出 CloneNotSupportedException 异常的不变 clone()。

4、使用Enum关键字定义的枚举类实现接口的情况

情况一: 实现接口, 在enum类中实现抽象方法

情况二: 让枚举类的对象分别实现接口中的抽象方法

1、和普通Java类一样, 枚举类可以实现一个或多个接口

2、若每个枚举值在调用实现的接口方法呈现相同的行为方式，则只要统一实现该方法即可

3、若需要每个枚举值在调用实现的接口方法呈现出不同的行为方式，则可以让每个枚举值分别来实现该方法

