Java 5新增了一个enum关键字（它与class、interface关键字的地位相同），用来定义枚举类。枚举类是一种特殊的类，它一样可以有自己的Field、方法，可以实现一个或者多个接口，也可以定义自己的构造器，一个java源文件中最多只能定义一个public访问权限的枚举类，且该java源文件也必须和该枚举类的类名相同。

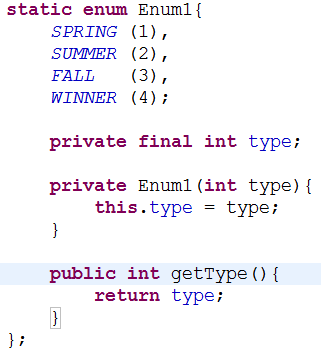
枚举类与普通类的区别：

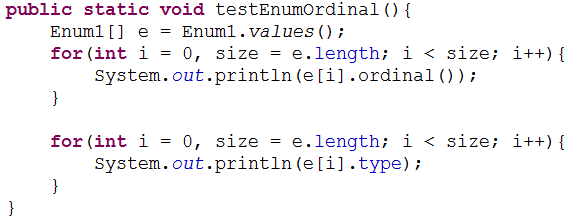
1. 枚举类可以实现一个或多个接口，使用enum定义的枚举类默认继承了java.lang.Enum类，而不是继承Object类，其中java.lang.Enum实现了java.lang.Serializable和java.lang.Comparable两个接口
2. 使用enum定义，**非抽象的枚举类**默认会使用final修饰，因此枚举类不能派生子类
3. 枚举类的构造器只能使用private访问控制器，如果省略了构造器的访问控制符，则默认使用private修饰，如果强制指定访问控制符，则只能指定private修饰符
4. 枚举类的所有实例必须在枚举类的第一行显式列出，否则这个枚举类永远都不能产生实例，列出这些实例时，系统会自动添加public static final修饰，无须程序猿显式添加

所有的枚举类都提供了一个values方法，该方法可以很方便地遍历所有的枚举类：。

枚举类通常应该设计带不可变类，也就是说，它的Field值不应该允许改变，这样会更加安全，而且代码更加简洁，为此，我们应该将枚举类的Field都是用private final修饰，因为我们将所有的Field都是用了final修饰符来修饰，所以必须在构造器里为这些Field指定初始值（或者在定义Field时指定默认值，或者在初始化块中指定初始值，但这两种情况并不常见），因为应该为枚举类显式定义带参数的构造器

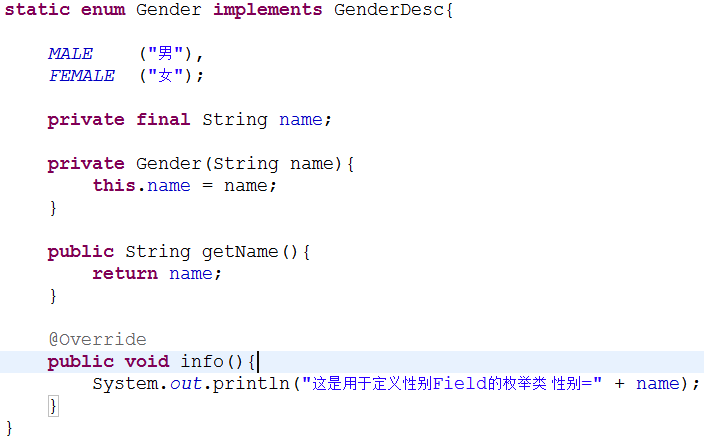
一旦为枚举类显式定义了带参数的构造器，列出枚举值时就必须对应地传入参数

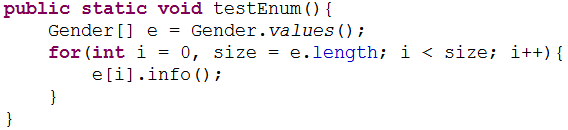






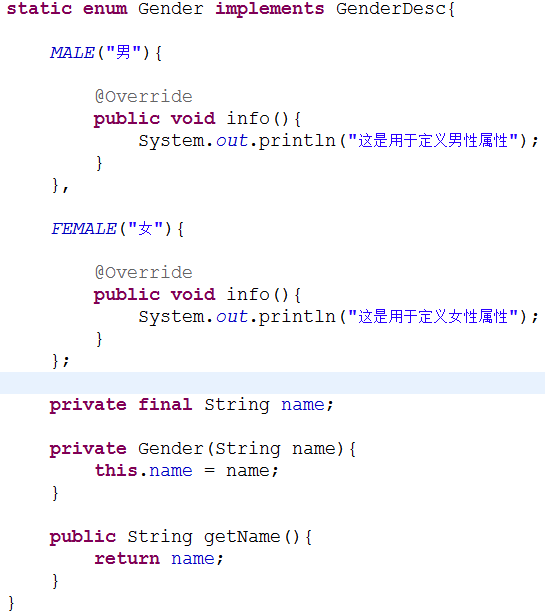
**实现接口的枚举类**







如果由枚举类来实现接口里的方法，则每个枚举值在调用该方法时都有相同的行为方式（因为方法体完全一样），如果需要每个枚举值在调用该方法时呈现不同的行为模式，则可以让每个枚举值分别来实现该方法，每个枚举值提供不同的实现方式，从而让不同枚举值调用该方法时具有不同的行为方式，比如：





上面代码看起来有点奇怪：当我们创建MALE和DEMALE两个枚举值时，后面又紧跟了一对花括号，实际上他是一个类体部分