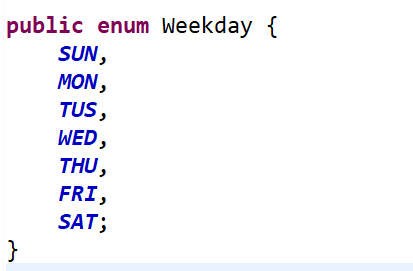
# 枚举类介绍

## 类的定义



1. 使用的是 enum 关键字而不是 class。
2. **多个枚举变量直接用逗号隔开，以分号结束**。
3. 枚举变量最好大写，多个单词之间使用”\_” 隔开（比如：INT\_SUM）。
4. 定义完所有的变量后，以分号结束，如果只有枚举变量，而没有自定义变量，分号可以省略（例如上面的代码就忽略了分号）。
5. 在其他类中使用 enum 变量的时候，只需要【类名. 变量名】就可以了，和使用静态变量一样。

## 类的方法

1. Weekday.valueOf() 方法

作用是**传来一个字符串，然后将它转变为对应的枚举变量**。前提是你传的字符串和定义枚举变量的字符串一抹一样，区分大小写。如果你传了一个不存在的字符串，那么会抛出异常。

1. Weekday.values()方法

这个方法**会返回包括所有枚举变量的数组**。在该例中，返回的就是包含了七个星期的 Weekday[]。可以方便的用来做循环。

1. 枚举变量的 toString() 方法

和枚举变量的 name() 方法一样，区别是toStirng可以重写，name不能重写。

该方法直接返回枚举定义枚举变量的字符串，比如 MON 就返回【”MON”】。

1. 枚举变量的ordinal() 方法

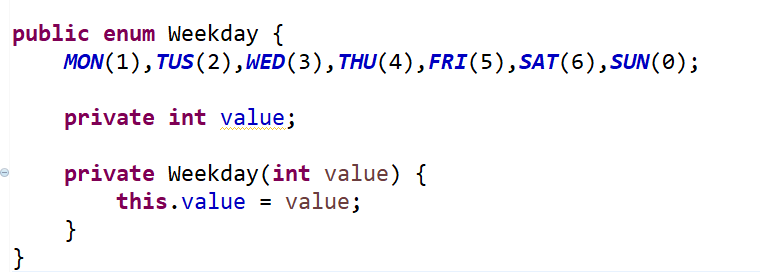
默认请款下，枚举类会给所有的枚举变量一个默认的次序，该次序从 0 开始，类似于数组的下标。而. ordinal() 方法就是获取这个次序（或者说下标）

1. 枚举变量的 compareTo() 方法

该方法用来比较两个枚举变量的” 大小”，实际上比较的是两个枚举变量的次序，返回两个次序相减后的结果，如果为负数，就证明变量 1” 小于” 变量 2 （变量 1.compareTo(变量 2)，返回【变量 1.ordinal() - 变量 2.ordinal()】）

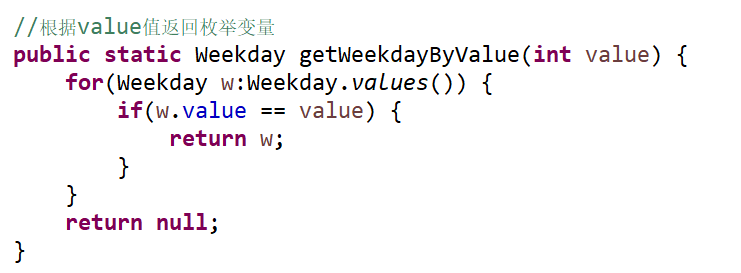
# 枚举类的高级使用

## 自定义变量



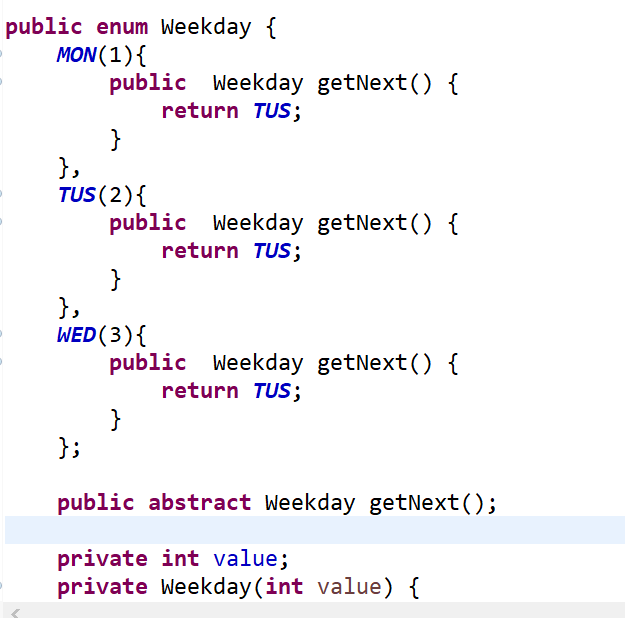
1. 一定要把枚举变量的定义放在第一行，并且以分号结尾。
2. 构造函数必须私有化。事实上，private 是多余的，你完全没有必要写，因为它默认并强制是 private，如果你要写，也只能写 private，写 public 是不能通过编译的。
3. 自定义变量与默认的 ordinal 属性并不冲突，ordinal 还是按照它的规则给每个枚举变量按顺序赋值。
4. 在枚举变量后添加一个对应的自定义变量，一定要定义为私有变量并在构造函数中初始化。

## 常用的枚举类方法

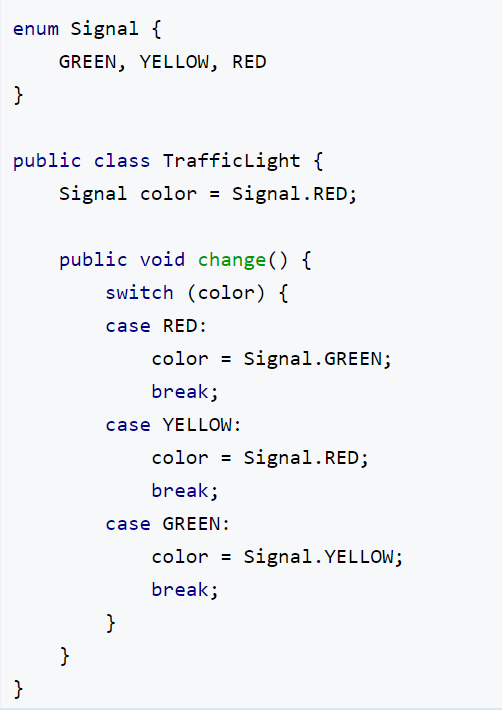


## 枚举类中的抽象方法

**枚举类不能继承其他类，也不能被其他类继承，直接在枚举变量后实现该方法。**

****

## Switch()语句中使用枚举类



1. Switch后面直接写枚举类名称，case枚举变量

# 总结

1. enum枚举类和平常类一样，但是不能继承其他类也不能被其他类继承，但是可以实现接口。在枚举类中也可以添加抽象方法，在每个枚举变量后面实现。
2. switch() 参数可以使用 enum 中的枚举变量
3. values() 方法是编译器插入到 enum 定义中的 static 方法，所以，当你将 enum 实例向上转型为父类 Enum 是，values() 就不可访问了。解决办法：在 Class 中有一个 getEnumConstants() 方法，所以即便 Enum 接口中没有 values() 方法，我们仍然可以通过 Class 对象取得所有的 enum 实例
4. 无法从 enum 继承子类，如果需要扩展 enum 中的元素，在一个接口的内部，创建实现该接口的枚举，以此将元素进行分组。达到将枚举元素进行分组。
5. enum 允许程序员为 eunm 实例编写方法。所以可以为每个 enum 实例赋予各自不同的行为。