**枚举**

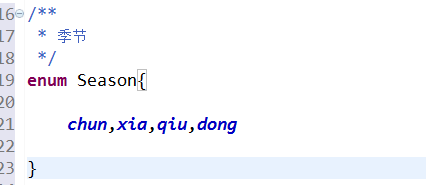
在实际编程中，往往存在着这样的“数据集”，它们的数值在程序中是稳定的，而且“数据集”中的元素是有限的。

例如星期一到星期日七个数据元素组成了一周的“数据集”，春夏秋冬四个数据元素组成了四季的“数据集”。

enum 的全称为 enumeration， 是 JDK 1.5  中引入的新特性，存放在 java.lang 包中，在上面的这种情况下，enum就能派上用场了。枚举类型的用途不仅如此，具体可用场景可看下面的介绍。

**语法定义：**

创建枚举类型要使用 enum 关键字，隐含了所创建的类型都是 java.lang.Enum 类的子类（java.lang.Enum 是一个抽象类）。枚举类型符合通用模式 Class Enum<E extends Enum<E>>，而 E 表示枚举类型的名称。枚举类型的每一个值都将映射到 protected Enum(String name, int ordinal) 构造函数中，在这里，每个值的名称都被转换成一个字符串，并且序数设置表示了此设置被创建的顺序。



　　解释：创建枚举类型要使用 enum 关键字，隐含了所创建的类型都是 java.lang.Enum 类的子类（java.lang.Enum 是一个抽象类）。枚举类型符合通用模式 Class Enum<E extends Enum<E>>，而 E 表示枚举类型的名称。枚举类型的每一个值都将映射到 protected Enum(String name, int ordinal) 构造函数中，在这里，每个值的**名称**都被转换成一个**字符串**，并且**序数**设置**表示了**此设置**被创建的顺序**。

　　上面那段代码声明的enum对象实际上调用了4次Enum(String name, int ordinal) 这个构造函数。（name 是常量，ordinal是被创建顺序）

enum对象的常用方法介绍：

int **compareTo**(E o)   
          比较此枚举与指定对象的顺序。

Class<E> **getDeclaringClass**()   
          返回与此枚举常量的枚举类型相对应的 Class 对象。

String **name**()   
          返回此枚举常量的名称，在其枚举声明中对其进行声明。

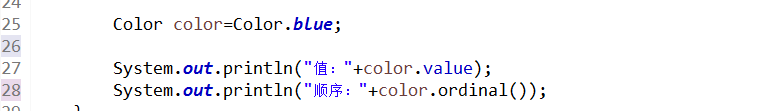
int **ordinal**()   
          返回枚举常量的序数（它在枚举声明中的位置，其中初始常量序数为零）。

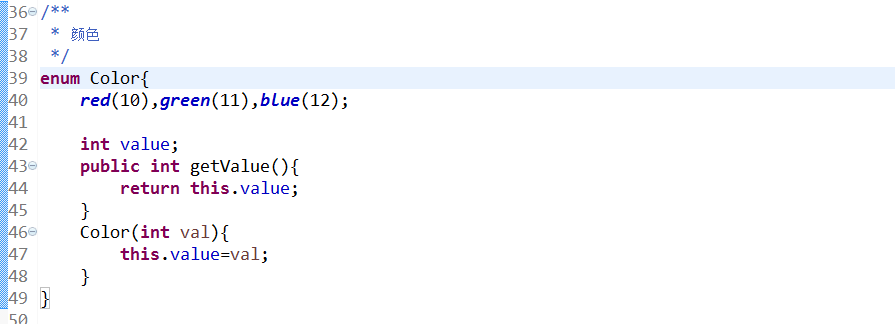
String **toString**()

           返回枚举常量的名称，它包含在声明中。

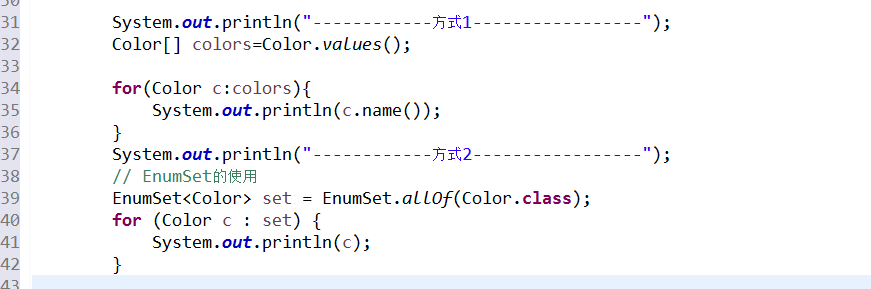
static <T extends Enum<T>> T **valueOf**(Class<T> enumType, String name)   
          返回带指定名称的指定枚举类型的枚举常量。

给enum 定义属性和方法：





遍历枚举中的所有元素：



总结：

可以把 enum 看成是一个普通的 class，它们都可以定义一些属性和方法，不同之处是：enum 不能使用 extends 关键字继承其他类，因为 enum 已经继承了 java.lang.Enum（java是单一继承），但可以继承接口。