java枚举

**关键字enum可以将一组具名的值的有限集合创建为一种新的类型，而这些具有名的值程序的常规组件使用**

例一：

public enum Shrubery {  
 *GROUND*,*CRAWLING*,*HANGING*}

public class EnumClass {  
 public static void main(String[] args) {  
 for (Shrubery s : Shrubery.*values*()) {  
 System.*out*.println(s + ":" + s.ordinal());//枚举的顺序值  
 System.*out*.println(s.compareTo(Shrubery.*CRAWLING*));//返回的是两个枚举值的顺序之差  
 System.*out*.println(s.equals(Shrubery.*CRAWLING*));//是不是同一个枚举  
 System.*out*.println(s == Shrubery.*CRAWLING*);//同上==判断是否是同一个实例  
 System.*out*.println(s.getDeclaringClass());  
 System.*out*.println(s.name());  
 }  
 }  
}

values()方法返回enum包含的实例数组，而且有严格的顺序但是在Enum类中并没有values（）方法，values方法是由编译器添加的

**静态导入enum**

public enum Spiciness {  
 *NOT*,*MILD*,*MEDIUM*,*HOT*,*FLAMING*}

import static main.Enum.Spiciness.\*;  
public class Burrito {  
 Spiciness degree;  
 public Burrito(Spiciness degree) {  
 this.degree = degree;  
 }  
 @Override  
 public String toString() {  
 return "Burrito{" +  
 "degree=" + degree +  
 '}';  
 }  
 public static void main(String[] args) {  
 System.*out*.println(new Burrito(NOT));  
 System.*out*.println(new Burrito(MEDIUM));  
 System.*out*.println(new Burrito(HOT));  
 }  
}

**向enum中添加新的方法**

public enum OzWitch {  
 *WEST*("a"), *NORTH*("b"), *EAST*("c"), *SOUTH*("d");  
 private String description;  
  
 private OzWitch(String description) {  
 this.description = description;  
 }  
  
 public String getDescription() {  
 return description;  
 }  
  
 public static void main(String[] args) {  
 for (OzWitch witch : OzWitch.*values*()) {  
 System.*out*.println(witch + ":" + witch.getDescription());  
 }  
 }  
}

enum除了不能继承以外基本就是一个常规类，所以可以向enum中添加新的方法甚至是main（）函数，enum和普通类没有太大区别除了有少许限制以外，都可以吧enum认为是一个普通类，从构造器private可以看出enum实际上在enum之外并不允许创建任何实例，就算不声明成private，也只能在enum内部使用其构造器构建enum实例，一旦enum定义结束，编译器不允许构造器构建任何实例

**switch语句中的enum**

在Java7之前，switch只能支持 byte、short、char、int或者其对应的封装类以及Enum类型

在Java7中加入了switch（string）

也可以利用enum实现switc（String）

public enum Animal {  
 *dog*, *cat*, *bear*;  
  
 public static Animal getAnimal(String animal) {  
 return *valueOf*(animal.toLowerCase());  
 }  
}

public class Client {  
 public void caseAnimal(String animal) {  
 switch (Animal.*getAnimal*(animal)) {  
 case *cat*:  
 System.*out*.println("this is a cat" + Animal.*cat*);  
 break;  
 case *dog*:  
 System.*out*.println("this is a dog");  
 break;  
 case *bear*:  
 System.*out*.println("this is a bear");  
 break;  
 }  
 }  
  
 /\*\*  
 \* @param args  
 \*/  
 public static void main(String[] args) {  
 Client client = new Client();  
 client.caseAnimal("cat");  
 }  
}

**enum不能继承其他的类，却可以实现一个或者多个接口**