Color color=Color.*GREEN*;  
switch (color){  
 case *BLACK*: **//只有在switch里面才可以这样使用BLACK** System.*out*.println("black");  
 break;  
 case *BLUE*:  
 System.*out*.println("blue");  
 break;  
 case *RED*:  
 System.*out*.println("red");  
 break;  
 default:  
 System.*out*.println("NOT FOUND");  
}

带名称和索引的枚举：

向枚举中添加普通方法：

public enum WeekEnum {  
 *SUN*("周日",0),*MON*("周一",1),*TUE*("周二",2),*WEN*("周三",3),*THU*("周四",4),  
 *FRI*("周五",5),*SAT*("周六",6);  
  
 private String name;  
 private int index;  
  
 private WeekEnum(String name,int index){  
 this.name=name;  
 this.index=index;  
 }  
  
 public static String getName(int index){  
 for(WeekEnum wk:WeekEnum.*values*()){  
 if(wk.index==index){  
 return wk.name;  
 }  
 }  
  
 return null;  
 }  
  
 public String getName(){  
 return this.name;  
 }  
  
 public void setName(String name){  
 this.name=name;  
 }  
  
 public int getIndex(){  
 return this.index;  
 }  
  
 public void setIndex(int index){  
 this.index=index;  
 }

然后就可以在下面使用了

WeekEnum wk=WeekEnum.*FRI*;  
System.*out*.println(wk);  
System.*out*.println(wk.getIndex());  
wk.setName("haha");  
wk.setIndex(10);  
System.*out*.println(wk.getName());  
System.*out*.println(WeekEnum.*MON*.getName());  
System.*out*.println(wk.getIndex());

重写枚举的toString()方法：

@Override  
public String toString(){  
 return "name: "+this.name+" index: "+this.index;  
}

枚举也可以继承接口或者类，枚举和类似乎有某些相似性

public enum EnumJiCheng implements IBehaviour {  
 *LAGRGE*, *MIDDLE*, *SMALL*;  
  
 public void print() {  
 System.*out*.println(*LAGRGE*.toString() + *MIDDLE*.toString() + *SMALL*.toString());  
 }  
  
 public String getInfo() {  
 return "EnumJiCheng";  
 }  
}

使用接口组织枚举：

//使用接口组织枚举  
public interface Food {  
 enum Coffee implements Food{  
 *BLACK\_COFFEE*,*DECAF\_COFFEE*,*LATTE*,*CAPPUCCINO* }  
  
 enum Dessert implements Food{  
 *FRUIT*,*CAKE*,*GELATO* }  
}