

1、课程名称：Enum



2、本课程预计讲解的知识点

本章目标

- 掌握enum关键字与Enum类的关系
- 了解Enum类中提供的常用方法

3、具体内容

Enum

从前面已经清楚的知道，使用enum关键字可以定义一个枚举，实际上此关键字表示的是java.lang.Enum类型，即：使用enum声明的枚举类型，就相当于定义一个类，而此类则默认继承java.lang.Enum类。

java.lang.Enum类的定义如下：

Enum 类的构造方法

```
public abstract class Enum<E extends Enum<E>>
    extends Object
    implements Comparable<E>, java.io.Serializable
```

此类定义的时候使用了泛型机制，而且实现了 Comparable 接口以及 Serializable 接口，证明此种类型是可以比较，可以被序列化的。

枚举类的主要操作方法

No.	方法	类型	描述
1	protected Enum(String name,int ordinal)	构造	接收枚举的名称和枚举的常量创建枚举对象
2	protected final Object clone() throws CloneNotSupportedException	普通	克隆枚举对象
3	public final int compareTo(Enum o)	普通	对象比较
4	public final boolean equals(Object other)	普通	比较两个枚举对象
5	public final int hashCode()	普通	返回枚举常量的哈希码
6	public final String name()	普通	返回此枚举的名称
7	public final int ordinal()	普通	返回枚举常量的序数
8	public static <T extends Enum<T>> T valueOf(Class<T> enumType,String name)	普通	返回带指定名称的指定枚举类型的枚举常量

如果此时，希望做一些改进，希望可以使用一些文字表示颜色的信息，则可以按照最早的 Color 类的形式，在枚举类中定义属性及自己的构造方法，但是一旦定义有参构造之后，在声明枚举对象的时候就必须要明确的调用构造方法，并传递参数。

```
enum Color{
    RED("红色"),GREEN("绿色"),BLUE("蓝色");
    private Color(String name){
        this.setName(name);
    }
    private String name;
    public void setName(String name){
        this.name = name;
    }
    public String getName(){
        return this.name;
    }
}

public class ConstructorEnum{
    public static void main(String args[]){
        for(Color c:Color.values()){
            System.out.println(c.ordinal()+"--> "+c.name());
        }
    }
}
```

如果现在不想通过构造设置内容，而是希望通过 setter()方法设置内容，则就必须按下的方式执行：

```
enum Color{
    RED,GREEN,BLUE;
    private String name;
    public void setName(String name){
        switch(this){
            //判断操作的是那个枚举对象
            case RED:{
                if("红色".equals(name)){
                    this.name = name; //允许设置名字
                }else{
                    System.out.println("设置内容错误。");
                }
                break;
            }
        }
        this.name = name;
    }
    public String getName(){
        return this.name;
    }
}
```

对于枚举 class 是随后要讲解的 java 反射机制中的内容，现在只需要按照此格式编写即可。

在枚举中实际上已经实现好了 Comparable 接口，所以枚举中的内容是可以排序的。

```
import java.util.Iterator;
import java.util.Set;
import java.util.TreeSet;
enum Color{
    RED,GREEN,BLUE;
}

public class ComparableEnum{
    public static void main(String args[]){
        Set<Color> t = new TreeSet<Color>(); //设置类型
        t.add(Color.RED); //加入绿色
        t.add(Color.GREEN); //加入红色
        t.add(Color.BLUE); //加入蓝色
        Iterator<Color> iter = t.iterator();
        while(iter.hasNext()){
            System.out.print(iter.next()+"、");
        }
    }
}
```

可以发现，按照序号进行排序的操作。

4、总结

- Enum 类的作用
- enum 关键字与 Enum 类的关系即可

5、预习任务

下一章内容

类集对枚举的支持