枚举(enum)

是一个特殊的类，一般表示一组常量，比如一年的 4 个季节，一个年的 12
一个星期的 7 天，方向有东南西北等。
类使用 enum 关键字来定义，各个常量使用逗号 ，来分割。
一个颜色的枚举类。

```
olor

), GREEN, BLUE;
```

Color 颜色常量有 RED, GREEN, BLUE，分别表示红色，绿色，蓝色。

```
olor

), GREEN, BLUE;


class Test

执行输出结果
lic static void main(String[] args)

  Color c1 = Color.RED;
  System.out.println(c1);
```

码输出结果为:

```
```

使用枚举

以声明在内部类中:

```
class Test

m Color
```

```
RED, GREEN, BLUE;
```

执行输出结果

```
lic static void main(String[] args)

  Color c1 = Color.RED;
  System.out.println(c1);
```

码输出结果为：

是通过 Class 在内部实现的，且所有的枚举值都是 public static final 的。

类 Color 转化在内部类实现：

```
olor

ublic static final Color RED = new Color();
ublic static final Color BLUE = new Color();
ublic static final Color GREEN = new Color();
```

元素

or 语句来迭代枚举元素：

```
lor

, GREEN, BLUE;

class MyClass {
c static void main(String[] args) {
 (Color myVar : Color.values()) {
ystem.out.println(myVar);
```

码输出结果为：

h 中使用枚举类

用于 switch 语句中：

反馈/建议

```
lor

, GREEN, BLUE;

class MyClass {
c static void main(String[] args) {
or myVar = Color.BLUE;

tch(myVar) {
ase RED:
  System.out.println("红色");
  break;
ase GREEN:
   System.out.println("绿色");
  break;
ase BLUE:
  System.out.println("蓝色");
  break;
```

码输出结果为:

```
```

, ordinal() 和 valueOf() 方法

的枚举类默认继承了 java.lang.Enum 类，并实现了 java.lang.Serializable 和 java.lang.Comparable 两个接口。

rdinal() 和 valueOf() 方法位于 java.lang.Enum 类中:

s() 返回枚举类中所有的值。

al()方法可以找到每个枚举常量的索引，就像数组索引一样。

Of()方法返回指定字符串值的枚举常量。

```
lor

, GREEN, BLUE;


class Test

lic static void main(String[] args)

 // 调用 values()
 Color[] arr = Color.values();

 // 迭代枚举
 for (Color col : arr)
 {
     // 查看索引
     System.out.println(col + " at index " + col.ordinal()
```

反馈/建议

```
}

// 使用 valueOf() 返回枚举常量，不存在的会报错 IllegalArgu
eption
System.out.println(Color.valueOf("RED"));
// System.out.println(Color.valueOf("WHITE"));
```

码输出结果为：

```
index 0
at index 1
t index 2
```

员

类一样可以用自己的变量、方法和构造函数，构造函数只能使用 private 访
所以外部无法调用。

包含具体方法，也可以包含抽象方法。 如果枚举类具有抽象方法，则枚举
例都必须实现它。

```
lor

, GREEN, BLUE;

构造函数
vate Color()

System.out.println("Constructor called for : " + this.toS
);


lic void colorInfo()

System.out.println("Universal Color");


class Test

输出
lic static void main(String[] args)

Color c1 = Color.RED;
System.out.println(c1);
c1.colorInfo();
```

码输出结果为：

```
uctor called for : RED
uctor called for : GREEN
```

反馈/建议

```
uctor called for : BLUE

sal Color
```

a 接口 Java 包(package) ➔

1 篇笔记 写笔记

举类中的抽象方法实现，需要枚举类中的每个对象都对其进实现。

```
enum Color{
    RED{
        public String getColor(){//枚举对象实现抽象方法
            return "红色";
        }
    },
    GREEN{
        public String getColor(){//枚举对象实现抽象方法
            return "绿色";
        }
    },
    BLUE{
        public String getColor(){//枚举对象实现抽象方法
            return "蓝色";
        }
    };
    public abstract String getColor();//定义抽象方法
}

public class Test{
    public static void main(String[] args) {
        for (Color c:Color.values()){
            System.out.print(c.getColor() + "、");
        }
    }
}
```

doggy 1年前 (2020-11-30)

关注微信

反馈/建议