**Java 枚举（enum）**

Java 枚举是一个特殊的类，一般表示一组常量，比如一年的 4 个季节，一年的 12 个月份，一个星期的 7 天，方向有东南西北等。

Java 枚举类使用 enum 关键字来定义，各个常量使用逗号 , 来分割。

例如定义一个颜色的枚举类。

enum Color

{

RED, GREEN, BLUE;

}

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

使用实例：

实例

enum Color

{

RED, GREEN, BLUE;

}

public class Test

{

// 执行输出结果

public static void main(String[] args)

{

Color c1 = Color.RED;

System.out.println(c1);

}

}

执行以上代码输出结果为：

RED

内部类中使用枚举

枚举类也可以声明在内部类中：

实例

public class Test

{

enum Color

{

RED, GREEN, BLUE;

}

// 执行输出结果

public static void main(String[] args)

{

Color c1 = Color.RED;

System.out.println(c1);

}

}

执行以上代码输出结果为：

RED

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

以上的枚举类 Color 转化在内部类实现：

class Color

{

public static final Color RED = new Color();

public static final Color BLUE = new Color();

public static final Color GREEN = new Color();

}

迭代枚举元素

可以使用 for 语句来迭代枚举元素：

实例

enum Color

{

RED, GREEN, BLUE;

}

public class MyClass {

public static void main(String[] args) {

for (Color myVar : Color.values()) {

System.out.println(myVar);

}

}

}

执行以上代码输出结果为：

RED

GREEN

BLUE

在 switch 中使用枚举类

枚举类常应用于 switch 语句中：

实例

enum Color

{

RED, GREEN, BLUE;

}

public class MyClass {

public static void main(String[] args) {

Color myVar = Color.BLUE;

switch(myVar) {

case RED:

System.out.println("红色");

break;

case GREEN:

System.out.println("绿色");

break;

case BLUE:

System.out.println("蓝色");

break;

}

}

}

执行以上代码输出结果为：

蓝色

values(), ordinal() 和 valueOf() 方法

enum 定义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。

values(), ordinal() 和 valueOf() 方法位于 java.lang.Enum 类中：

values() 返回枚举类中所有的值。

ordinal()方法可以找到每个枚举常量的索引，就像数组索引一样。

valueOf()方法返回指定字符串值的枚举常量。

实例

enum Color

{

RED, GREEN, BLUE;

}

public class Test

{

public 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());

}

// 使用 valueOf() 返回枚举常量，不存在的会报错 IllegalArgumentException

System.out.println(Color.valueOf("RED"));

// System.out.println(Color.valueOf("WHITE"));

}

}

执行以上代码输出结果为：

RED at index 0

GREEN at index 1

BLUE at index 2

RED

枚举类成员

枚举跟普通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 访问修饰符，所以外部无法调用。

枚举既可以包含具体方法，也可以包含抽象方法。 如果枚举类具有抽象方法，则枚举类的每个实例都必须实现它。

实例

enum Color

{

RED, GREEN, BLUE;

// 构造函数

private Color()

{

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

}

public void colorInfo()

{

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

}

}

public class Test

{

// 输出

public static void main(String[] args)

{

Color c1 = Color.RED;

System.out.println(c1);

c1.colorInfo();

}

}

执行以上代码输出结果为：

Constructor called for : RED

Constructor called for : GREEN

Constructor called for : BLUE

RED

Universal Color