枚举类型

C-2 创建: 张林伟, 最后修改: 张林伟 2018-07-18 16:15

目录

简介

枚举类型通常指是一组类似的值的组合成的一种类型。在Java中使用关键字enum声明枚举类型。相较于常量定义一组值，它会更安全，更具有可读性。

枚举类型可用于switch语句。

Demo

定义枚举类型

代码块 Java

```java
/**
 * Desc:
 * ----------------------------------
 * Author:zhanglinwei02@meituan.com
 * Date:2018/7/18
 * Time:12:24
 */
public enum Color {
    RED(1, "红"),
    YELLOW(2, "黄"),
    BULE(3, "蓝");

    private int code;
    private String value;

    private Color(int code, String value) {
        this.code = code;
        this.value = value;
    }

    public String toString() {
        return "code: " + this.code + ", value: " + this.value;
    }

    public int getCode() {
        return this.code;
    }

    public String getValue() {
        return this.value;
    }
}
```

测试枚举类型

代码块 Java

```java
/**
 * Desc:
 * ----------------------------------
```

```
 * Author:zhanglinwei02@meituan.com
 * Date:2018/7/18
 * Time:12:10
 */
public class  EnumTest {
    public static void main(String[] args) {
        Color color = Color.RED;
        System.out.println(color.toString());
        System.out.println(color.getCode());
        System.out.println(color.getValue());
    }
}
```

打印结果

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_172.jdk/Contents/Home/bin/java ...
code: 1, value: 红
1
红

Process finished with exit code 0
```

理解

对于枚举类型中的RED、YELLOW和BLUE，JVM 都会在运行期构造成出一个简单的对象实例一一对应。这些对象都有唯一的 identity。

编译枚举类型时，编译器会创建一个类。

枚举类型可以具有构造函数，字段和方法。枚举类型仅在编译器生成的代码中实例化。

每个枚举类型都隐式地扩展java.lang.Enum类。 Enum类中定义的所有方法都可以与所有枚举类型一起使用。

枚举常量顺序是有序的，序号从0开始

代码块 Java

```
for (Color c: Color.values()) {
    System.out.println(c.ordinal());
    System.out.println(c.toString());
    }
```

结果为

```
0
code: 1, value: 红
1
code: 2, value: 黄
2
code: 3, value: 蓝
```

扩展

枚举主体

代码块 Java

```
/**
 * Desc:
 * ----------------------------------
 * Author:zhanglinwei02@meituan.com
 * Date:2018/7/18
 * Time:12:24
 */
public enum Color {
    RED(1, "红") {
        public String getComments() {
            return "国旗的主体颜色是红色";
        }
    },
    YELLOW(2, "黄") {
        public String getComments() {
```

```
            return "向日葵是黄色的";
        }
    },
    BULE(3, "蓝") {
        public String getComments() {
            return "晴空是蓝色的";
        }
    };

    private int code;
    private String value;

    Color(int code, String value) {
        this.code = code;
        this.value = value;
    }

    public String toString() {
        return "code: " + this.code + ", value: " + this.value;
    }

    public int getCode() {
        return this.code;
    }

    public String getValue() {
        return this.value;
    }

    public abstract String getComments();
}
```

级别枚举Color有个抽象方法getComments，每个实例常量都有一个实体为getDistance()方法提供实现。这样每个实例对象（RED、YELLOW和BLUE）都能调用getComments方法。

枚举比较

三种方式

- 使用Enum类的compareTo()方法
- 使用Enum类的equals()方法
- 使用==运算符

Enum类的compareTo()方法比较同一枚举类型的两个枚举常量。它返回两个枚举常量的序数差（ordinal方法值）。如果两个枚举常量相同，则返回零。

代码块 Java

```
System.out.println(Color.RED.compareTo(Color.BULE));
```

结果为-2。

枚举类equals方法（不重写）

```
/**
 * Returns true if the specified object is equal to this
 * enum constant.
 *
 * @param other the object to be compared for equality with this object.
 * @return  true if the specified object is equal to this
 *          enum constant.
 */
public final boolean equals(Object other) {
    return this==other;
}
```

注意点

1. enum 类型不支持 public 和 protected 修饰符的构造方法，因此构造函数一定要是 private 或 friendly 的。也正因为如此，所以枚举对象是无法在程序中通过直接调用其构造方法来初始化的。
2. 定义 enum 类型时候，如果是简单类型，那么最后一个枚举值后不用跟任何一个符号；但如果有定制方法，那么最后一个枚举值与后面代码要用分号';'隔开，不能用逗号或空格。
3. 由于 enum 类型的值实际上是通过运行期构造出对象来表示的，所以在 cluster 环境下，每个虚拟机都会构造出一个同义的枚举对象。因而在做比较操作时候就需要注意，如果直接通过使用等号 (' == ') 操作符，这些看似一样的枚举值一定不相等，因为这不是同一个对象实例。

参考资料

https://www.ibm.com/developerworks/cn/java/j-lo-enum/index.html

https://www.w3cschool.cn/java/java-enum-types.html