枚举（Enum）

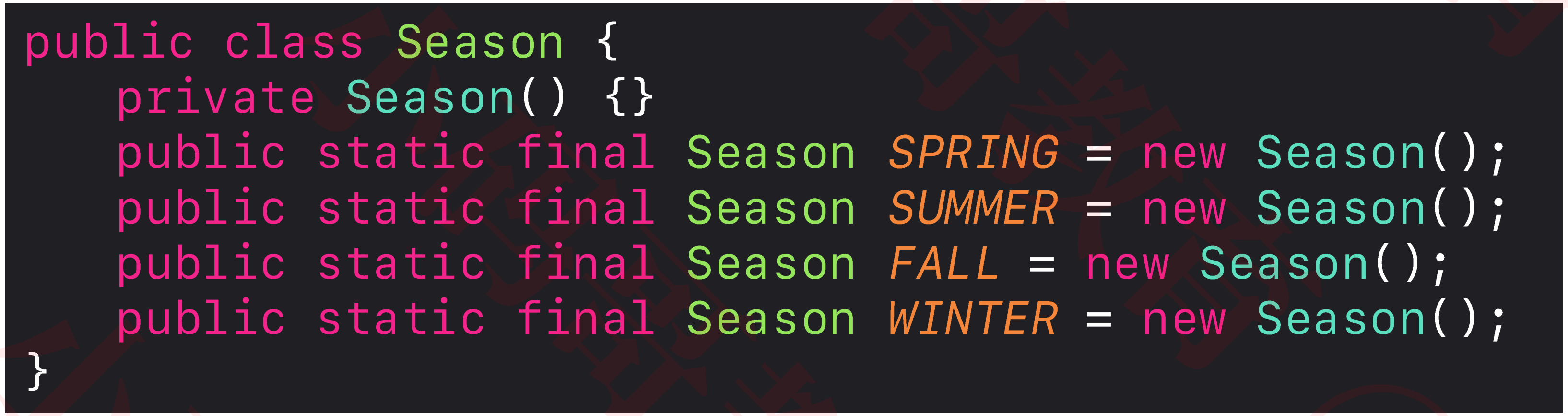
@M了个J

李明杰

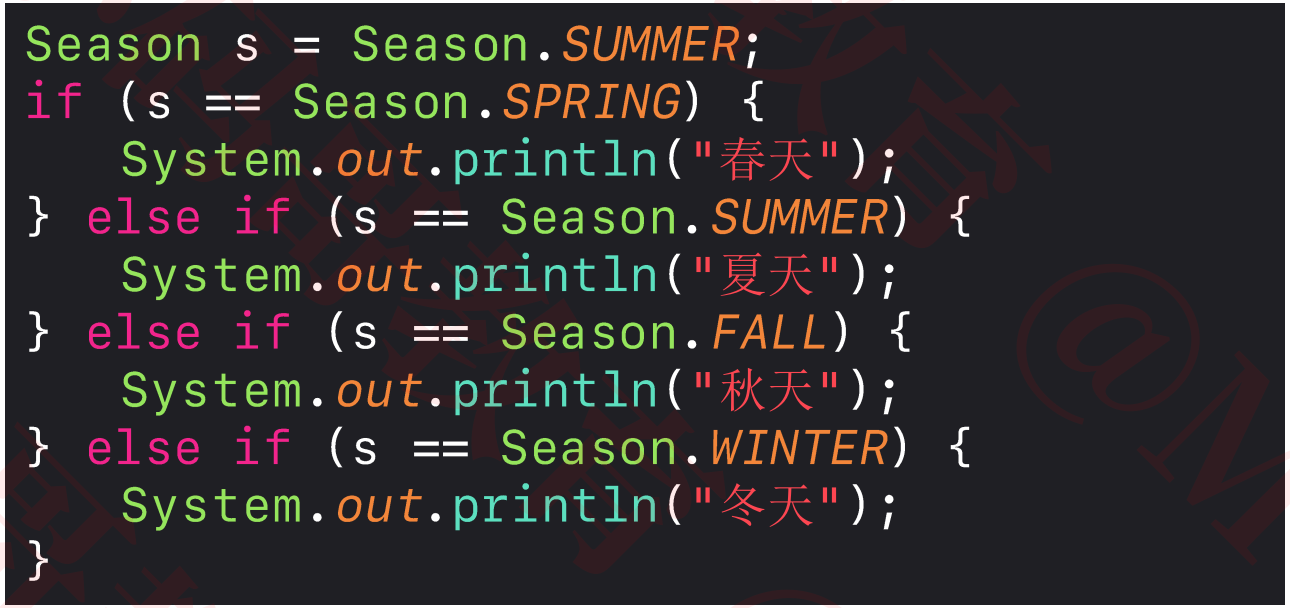
https://github.com/CoderMJLee

http://cnblogs.com/mjios

自定义类实现枚举效果



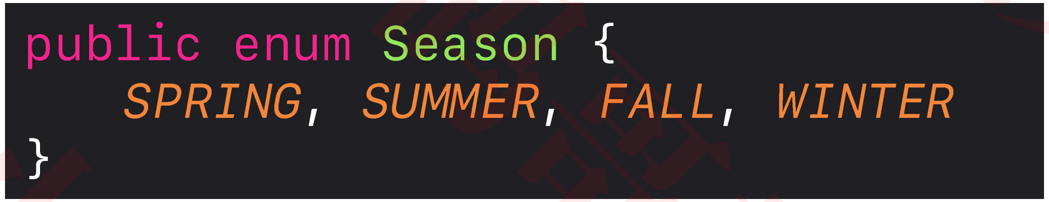
```
public class Season {
    private Season() {}
    public static final Season SPRING = new Season();
    public static final Season SUMMER = new Season();
    public static final Season FALL = new Season();
    public static final Season WINTER = new Season();
}
```



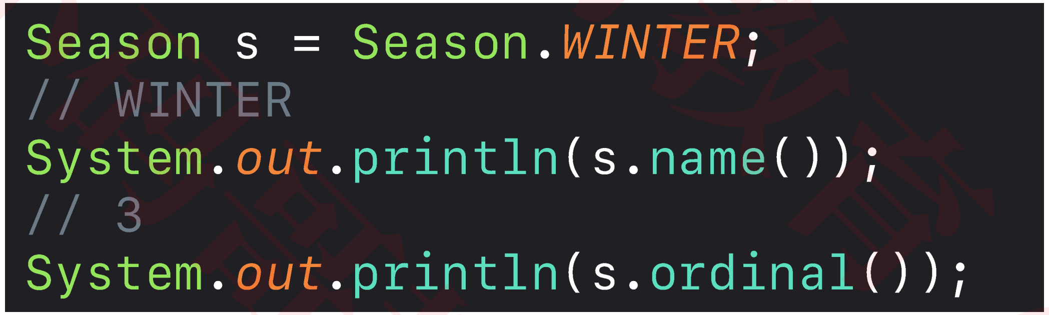
```
Season s = Season.SUMMER;
if (s == Season.SPRING) {
    System.out.println("春天");
} else if (s == Season.SUMMER) {
    System.out.println("夏天");
} else if (s == Season.FALL) {
    System.out.println("秋天");
} else if (s == Season.WINTER) {
    System.out.println("冬天");
}
```

枚举类型（Enum Type）

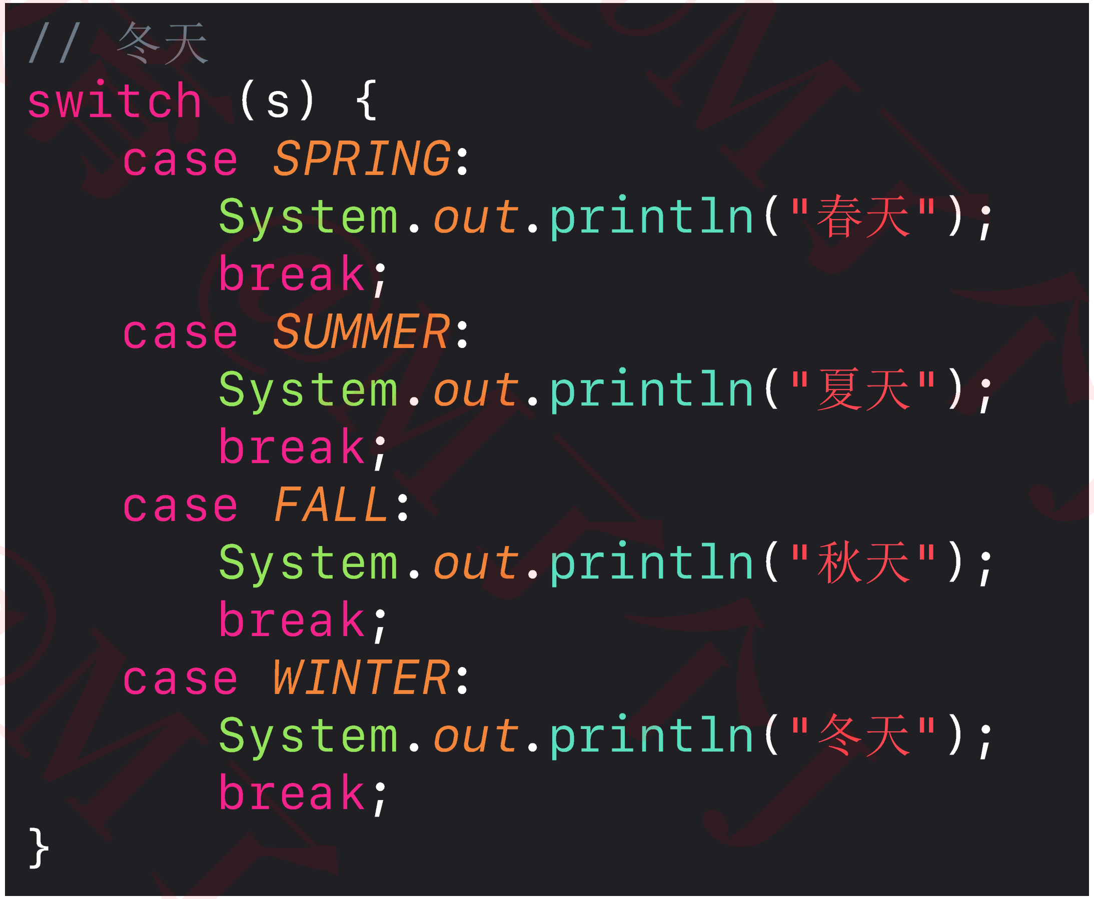
- 如果一个变量的取值只可能是固定的几个值，可以考虑使用枚举类型
  - 枚举由一组预定义的常量构成



```java
public enum Season {
    SPRING, SUMMER, FALL, WINTER
}
```



```java
Season s = Season.WINTER;
// WINTER
System.out.println(s.name());
// 3
System.out.println(s.ordinal());
```



```java
// 冬天
switch (s) {
    case SPRING:
        System.out.println("春天");
        break;
    case SUMMER:
        System.out.println("夏天");
        break;
    case FALL:
        System.out.println("秋天");
        break;
    case WINTER:
        System.out.println("冬天");
        break;
}
```

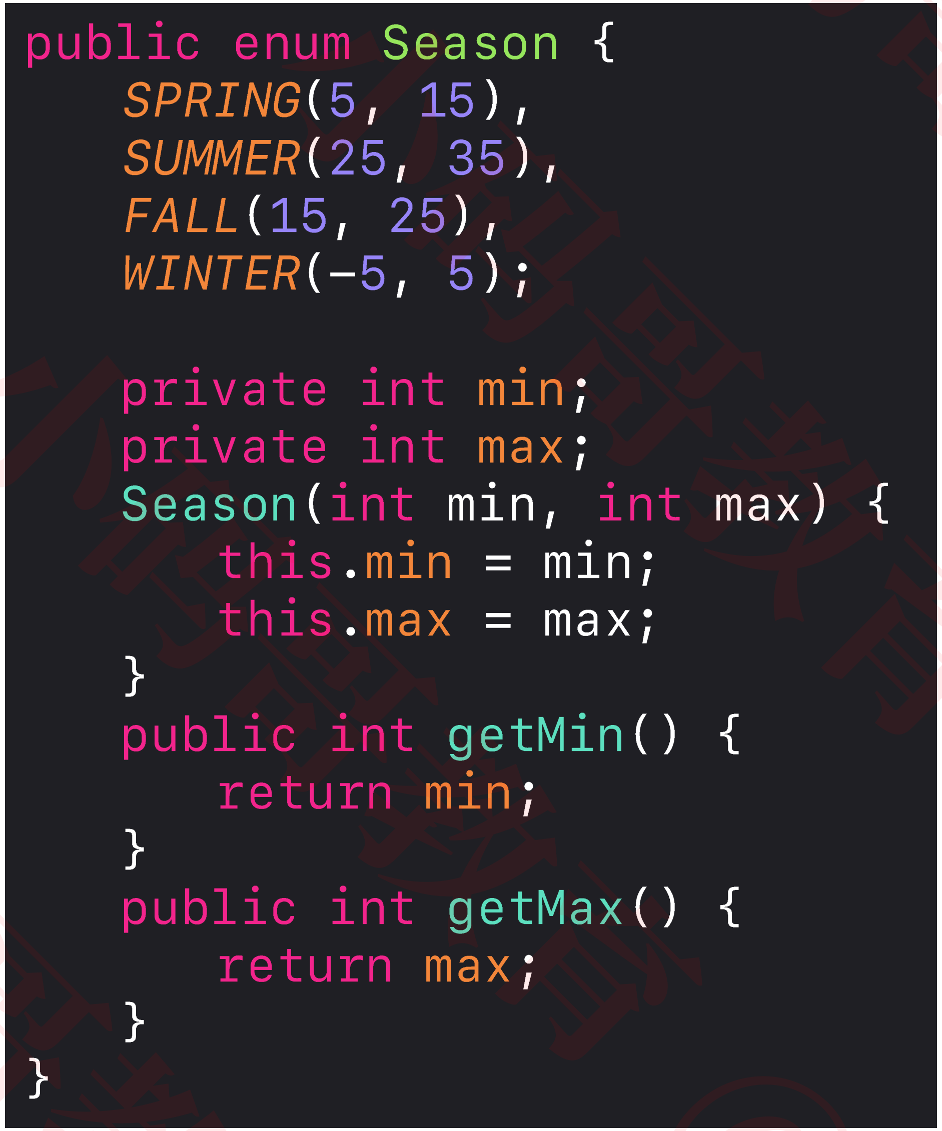
枚举的使用注意

- 枚举的本质是类，所有枚举类型最终都隐式继承自 java.lang.Enum

- 枚举定义完常量后，可以再定义成员变量、方法等内容（这时最后一个常量要以分号结束）

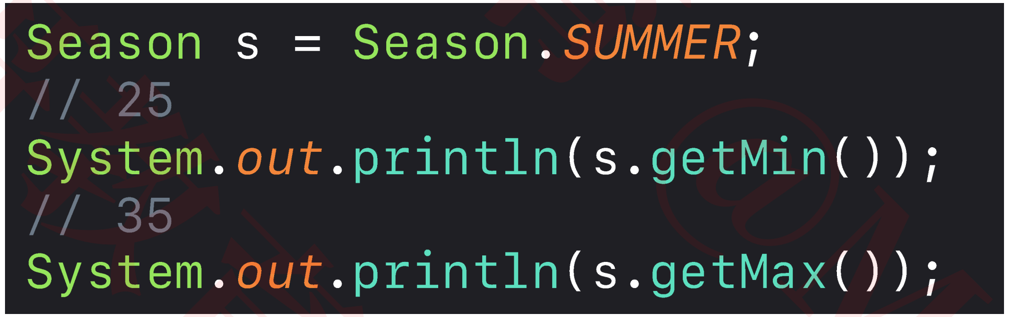
- 枚举的构造方法权限必须是 无修饰符 或者 `private`
  - Java 会主动调用构造方法初始化每一个常量，你不能主动调用构造方法

自定义了构造方法的枚举



```
public enum Season {
    SPRING(5, 15),
    SUMMER(25, 35),
    FALL(15, 25),
    WINTER(-5, 5);

    private int min;
    private int max;
    Season(int min, int max) {
        this.min = min;
        this.max = max;
    }
    public int getMin() {
        return min;
    }
    public int getMax() {
        return max;
    }
}
```



```
Season s = Season.SUMMER;
// 25
System.out.println(s.getMin());
// 35
System.out.println(s.getMax());
```