

枚举

► 什么是枚举

- C++可以提供了关键字`const`限定符处理符号常量。

```
const 数据类型 符号常量名 = 常量值;
```

```
const int Months = 12;
```

- 常量被初始化后，其值就被固定了，不允许再修改该常量的值。
- C++的`enum`工具提供了另一种创建符号常量的方式，可以代替`const`，它还允许定义新类型。
- 定义枚举类型的格式：

```
enum 枚举名 {枚举量1, 枚举量2, ...};
```

```
enum colors {red, orange, yellow, green, blue, purple};
```

► 枚举的使用

- 使用枚举名声明该类型的变量：

```
colors band;
```

- 只能将定义枚举时使用的枚举量赋值给这种枚举的变量：

```
band = red;           band = 5; ❌
```

- 可以通过强制类型转换，将有效的int值转换成枚举量：

```
band = colors(4);
```

- 枚举量是整型，可被提升为int类型，但int类型不能自动转换为枚举类型：

```
int col = blue;       col = 3 + red;
```

► 设置枚举量的值

- 可以使用赋值运算符来显式地设置枚举量的值，指定的值必须是整数：

```
enum bits{one = 1, two = 2, four = 4, eight = 8};
```

- 也可以只显式地定义其中一些枚举量的值：

```
enum bigstep{first, second = 100, third};
```

`first`默认情况下为0，后面没有初始化的枚举量的值比其前面的枚举量大1，`third`的值为101。

- 可以创建多个值相同的枚举量：

```
enum {zero, null = 0, one, numero_uno = 1};
```

► 枚举的取值范围

- 每个枚举都有取值范围，通过强制类型转换，可以将取值范围中的任何整数赋值给枚举变量，即使这个值不是枚举值：

```
enum bits{one = 1, two = 2, four = 4, eight = 8};  
bits myflag = bits(6);
```

■ 枚举取值范围

- 上限：大于枚举量的最大值的、最小的2的幂，减去1。

假设最大枚举量为101，2的幂中比它大的最小值为128，上限为 $128 - 1 = 127$ 。

- 下限：枚举量最小值为0，则下限为0；否则为小于它的、最大的2的幂，加上1。

假设最小枚举量为-6，2的幂中比它小的最大的值是-8，因此下限为 $-8 + 1 = -7$ 。

Thanks

