

二级C 精品课程2-6：枚举类型的相关知识 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/181/2021_2022__E4_BA_8C_E7_BA_A7C___E7_c97_181148.htm 我们常常需要一些整型常量表示某一个的数据范围。例如，假定需要表示汽车的颜色：

```
const int cRED = 0. const int cBLUE = 1. ... int auto_colour.  
auto_colour = cBLUE.
```

 上面的程序语句是用整型变量

auto_colour表示颜色，下面的语句也是合法的：

```
auto_colour = -1. ...
```

 显然，-1并不是所需要的颜色值。如果能够限定auto_colour在限定的整数范围（即规定的几种颜色集合）内取值就比较好，C中的枚举类型能够让我们做到这一点。

枚举类型是一种用户定义的数据类型，其一般定义形式为：

```
enum 枚举类型名 { 标识符[=整型常数], 标识符[=整型常数], ... 标识符[=整型常数], }
```

 "枚举类型名"右边花括号中的内容称之为枚举表，枚举表中的每一项称之为枚举成员，枚举成员是常量，也就是说，枚举表是整型常量的集合。枚举成员之间用逗号隔开，方括号中的"整型常数"是枚举成员的初值。

如果不给枚举成员赋初值，即省掉了标识符后的"=整型常数"时，则编译器为每一个枚举成员给一个不同的整型值，第一个成员为0，第二个为1，等等。当枚举类型中的某个成员赋值后，其后的成员则按依次加1的规则确定其值。下面是一个枚举类型定义实例：

```
enum Colour { eRED, eBLUE,  
eYELLOW, eGREEN, eSILVERGREY, eBURGUNDY }
```

 这样就定义了一个称之为Colour的枚举类型，编译器给枚举成员eRED，...

，...，eBURGUNDY的值分别为0，...，5。定义了枚举类型，就可以定义枚举类型的变量，其值限定为某一个枚举成员

。例如：`Colour auto_colour. ... auto_colour = eBURGUNDY.`
`Colour`类型变量`auto_colour`只能取`eRED`，`...`，`eBURGUNDY`六个值中的一个，下面的语句将会出现编译错误：
`auto_colour = 4.`我们也可以在定义枚举类型时，为枚举成员指定初值，例如：`enum Colour { eRED=5, eBLUE, eYELLOW, Egreen=30, Esilvergrey=40,eBURGUNDY }`。这时，`eBLUE=6`、`Eyellow=7`、`Eburgundy=41`。由上述几种基本数据类型通过各种组合可以得到复合数据类型，这些数据类型在以后章节会有详细讲解。如果一个变量只有几种可能的值，就可以把它定义为枚举类型。“枚举”，顾名思义，就是把这种类型数据可取的值一一列举出来。例如，可以定义一个表示日期的枚举类型`weekday`：`enum weekday{sun,mon,tru,wed,thu,fri,sat}.`这里`enum`是关键字，用于定义枚举数据类型，`weekday`是枚举类型的名字，枚举元素用标识符表示。接下来我们可以用`weekday`来定义变量，例如：`weekday today,nextday.` C语言也允许在定义枚举类型的同时定义枚举变量，例如：`enum weekday{sun,mon,tru,wed,thu,fri,sat}today,nextday.`这样，变量`today`和`nextday`就具有`weekday`类型，它们的取值只能是`sun`，`mon`，`...`，`sta`，例如可有：`today = sun. nextday = mon.`
`if(today == sat) nextday = sun` //如果`today`是`sat`的话，那么`nextday`赋值为`sun` C编译器对枚举元素实际上是按整型常数处理。所以，就象前面所说的那样，当遇到枚举元素列表时，编译程序就把其中第一个标识符赋0值，第二、三、...个标识符依次赋1、2、...。因此，当枚举元素赋给枚举变量时，该变量实际得到一个整数值。例如：`today = sun.`是将0赋给`today`，而不是将字符串“sun”赋给`today`。赋值后：`cout`语句

的输出结果是0。 我们也可以在枚举类型定义时指定枚举元素的值，例如：`enum weekday{sun = 7,mon = 1,tue,wed,thu,fri,sat}`. 这时sun的值是7，mon的值是1，而tue以后各元素的值，从mon的值开始，每次递增1，即tue的值为2，wed的值为3，...。如果不写`mon = 1`，则mon的值为8，tue的值为9，依此类推。 由于枚举元素是常量，所以我们在程序中不能对它们进行赋值，如"`sun = 0 ; mon = 1 ;`"将产生错误。 既然枚举元素就是整型值，那么它有什么存在的必要呢？至少有两个原因：一个是用标识符表示数值增加了程序的可读性，例如：`if(today == sat) next = sun` 就比 `if(today == 6) nextday = 0`. 清楚多了；另一个重要的原因是：它限制了变量取值的范围，如现在today只能取sun~sat中的值。 注意

：enum在C和C++中使用的不同。下面的C语言语句定义了名为day的变量：`enum Days { Monday, Tuesday, Wednesday, Thursday, Friday }. enum Days day`. 在C++中，定义为：`enum Days { Monday, Tuesday, Wednesday, Thursday, Friday }. Days day`. C++中定义了枚举类型数据后，程序可以把它当作一种类型使用，不需在类型名Days前加enum关键字。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com