枚举成员是常量，用来初始化枚举成员的值必须是常量表达式。

常量表达式：编译器在编译时就能计算出结果的整型表达式。

不能改变枚举成员的值。

base.h

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 文件名 ：base.h

/\* 作者 ：刘珅珅

/\* 日期 ：2011/12/14

/\* 功能描述 ：基础类

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef BASE\_H

#define BASE\_H

class Base

{

public:

// 枚举

enum open\_mode {

input,

output,

append

};

void func();

};

#endif

base.cpp

#include "base.h"

#include <iostream>

using namespace std;

void Base :: func()

{

// 默认情况下第一个枚举成员赋值为0，后面的每一个枚举成员的赋值比前面大1

if (input == 0)

{

cout << "打开模式为输入" << endl;

}

if (output == 1)

{

cout << "打开模式为输出" << endl;

}

if (append == 2)

{

cout << "打开模式为追加" << endl;

}

}

test.cpp

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 文件名 ：test.cpp

/\* 作者 ：刘珅珅

/\* 日期 ：2011/12/14

/\* 功能描述 ：测试枚举类型

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "base.h"

#include <iostream>

using namespace std;

enum Points{

point2d = 2, point2w,

point3d = 3, point3w

};

int main(int argc, char \*argv[])

{

Base base;

base.func();

//base.input = 1; // error：不能改变枚举成员的值

if (point2w == 3 && point3w == 4)

{

cout << "point2w = 3, point3w = 4" << endl;

}

Points pt3d = point3d;

Points pt2w;

pt2w = pt3d;

//Points pt2d = 2; // error：枚举类型的对象的初始化或赋值只能通过其枚举成员或同一枚举类型的其他对象来进行

return 0;

}