

类中成员变量的初始化

1、普通的变量：一般不考虑啥效率的情况下可以在构造函数中进行赋值。考虑一下效率的可以再构造函数的初始化列表中进行。

```
class CA
{
public:
int data;
.....
public:
CA();
.....
};
CA::CA():data(0)//.....#1.....初始化列表方式
{
//data = 0;//.....#1.....赋值方式
};
```

2、static静态变量：

static变量属于类所有，而不属于类的对象，因此不管类被实例化了多少个对象，该变量都只有一个。在这种性质上理解，有点类似于全局变量的唯一性。

```
class CA
{
public:
static int sum;
.....
public:
CA();
.....
};
int CA::sum=0;//.....#2.....类外进行初始化
```

3、const常量变量：

const常量需要在声明的时候即初始化。因此需要在变量创建的时候进行初始化。一般采用在构造函数的初始化列表中进行。

```
class CA
{
public:
const int max;
.....
public:
CA();
.....
};
CA::CA():max(100)
{
.....
}
```

4、Reference引用型变量：

引用型变量和const变量类似。需要在创建的时候即进行初始化。也是在初始化列表中进行。但需要注意用Reference类型。

```
class CA
{
public:
int init;
int& counter;
```

```

.....
public:
CA();
.....
};
CA::CA():counter(&init)
{
.....
}

```

5、const static integral变量:

对于既是const又是static而且还是整形变量，C++是给予特权的(但是不同的编译器可能会有会有不同的支持，VC 6好像就不支持)。可以直接在类的定义中初始化。short可以，但float的不可以哦。

//例float类型只能在类外进行初始化

```
// const float CA::fmin = 3.14;
```

```

class CA
{
public:
//static const float fmin = 0.0;// only static const integral data members can be initialized within a
class
const static int nmin = 0;
.....
public:
.....
};

```

总结起来，可以初始化的情况有如下四个地方：

- 1、在类的定义中进行的，只有const且static且integral的变量。
- 2、在类的构造函数初始化列表中，包括const对象和Reference对象。
- 3、在类的定义之外初始化的，包括static变量。因为它是属于类的唯一变量。
- 4、普通的变量可以在构造函数的内部，通过赋值方式进行。当然这样效率不高。

类的定义体中只能初始化const integral data型的量。对于static型的量，那就放在.cpp文件中吧！当然了，还不能放在成员函数中，因为static量是类的，不是某个对象的。那样的话每个对象都来操作属于所有对象（类）的东西，岂不是会乱套，所以不能允许这种行为。

但是，static量可以在类的构造函数中赋值，当然是不可以放在初始化成员列表中的，可是在构造函数中赋值时不可以使用copy construction，提示这样的错误：

```
term does not evaluate to a function taking 1 arguments
```

那么，对于类里面的static函数的声明和定义是这样的：

static函数的声明可以像普通成员函数一样声明，只是在前面加上一个static关键字。

如：

```

private:
static int GetXYZ();

```

而在，定义时，像static变量那样，也是不可以加上static关键字，若写成：

```

static int A::GetXYZ()
{
.....
}

```

就会提示：

```
'static' should not be used on member functions defined at file scope
```

所以应该写成是这样：

```

int A::GetXYZ()
{//他是static型函数的性质，就用其他方法来辨别吧，比如在这儿写上：this is a static function
.....
}

```

至于static函数的使用，可以再你所编写的代码段中这样插入：

```

.....
A::GetXYZ(); //可以看出他是类的东东，不是对象的
.....

```

当然，对于public型的static量（假设叫CString S_str），可以这样使用：

```
A::S_str = "Hello !";
```

```
CString str = A::S_str;
```

c++成员变量初始化问题分类：c/c小结2009-11-03 17:19

C++为类中提供类成员的初始化列表

类对象的构造顺序是这样的：

1. 分配内存，调用构造函数时，隐式 / 显示的初始化各数据成员
2. 进入构造函数后在构造函数中执行一般计算
1. 类里面的任何成员变量在定义时是不能初始化的。
2. 一般的数据成员可以在构造函数中初始化。
3. const数据成员必须在构造函数的初始化列表中初始化。
4. static要在类的定义外面初始化。
5. 数组成员是不能在初始化列表里初始化的。
6. 不能给数组指定明显的初始化。

这6条一起，说明了一个问题：C++里面是不能定义常量数组的！因为3和5的矛盾。这个事情似乎说不过去啊？没有办法，我只好转而求助于静态数据成员。

到此，我的问题解决。但是我还想趁机复习一下C++类的初始化：

1. 初始化列表：CSomeClass::CSomeClass() : x(0), y(1) {}
2. 类外初始化：int CSomeClass::myVar=3;
3. const常量定义必须初始化，C++类里面使用初始化列表；
4. C++类不能定义常量数组。

关于const，复习一下常量指针：

如果const位于星号的左侧，则const就是用来修饰指针所指向的