类的对象成员
–对象成员的声明、初始化和访问

- 下面以例2-16为例，分别说明对象成员的声明、初始化和访问方法。
- 1．对象成员的声明
- 对象成员的声明与其他成员相同，其语法格式为：
- <类名> <对象成员名表>；
- 例2-16中的语句“Point m_center;”，就是在类Circle中声明了一个成员m_center，它是类Point类型的对象，是类Circle中的对象成员。

- 2．对象成员的初始化
- 一个对象数据成员的初始化是通过调用构造函数来完成的，即一个对象成员的初始化是“大对象”被创建时一同被创建的。具体方法是，在定义“大对象”所在类的构造函数时，需要在函数体外通过成员初始化列表将参数传递到对象成员的构造函数中。成员初始化列表的格式为：
- <对象成员1>（<初值表>）[,…, <对象成员n>（<初值表>）]

- 在例2-16中，Circle类中的构造函数为：
- Circle(double cx,double cy,double cr):m_center(cx,cy)
- 其中，“:m_center(cx,cy)”就是将大对象构造函数“Circle(double cx,double cy,double cr)”中的参数cx和cy传递给对象成员m_center的构造函数“Point(int a,int b)”，通过此构造函数来初始化m_center对象。所以，如果某个类中含有对象成员，则该类的构造函数就应包含一个初始化列表，负责对类中包含的对象成员进行初始化。

- 提示：
- 如果一个对象成员有无参的构造函数，则该对象可以不出现在初始化列表。
- 在创建一个含有对象成员的对象时，构造函数被系统自动调用。先按照对象成员的声明顺序执行相应的构造函数完成各对象成员的初始化工作，然后再执行自己的构造函数完成对象的初始化工作。析构函数的调用顺序与构造函数的调用顺序相反。

- 3．对象成员的访问
- 在类内，如果访问对象成员的公有成员，可以通过对象成员名直接访问对象成员的公有成员；如果访问对象成员的私有成员，则需要通过调用公有成员函数间接地对对象成员的私有成员进行访问。在例2-16中，由于m_x和m_y是Point类中的私有成员，所以，Circle类中的DisplayCircleInfo函数需要通过对象成员m_center调用Point类的公有成员函数，即m_center.GetX()和m_center.GetY()，间接地访问m_center对象成员的私有成员m_x和m_y。

如果m_x和m_y是公有成员，则Circle类中的DisplayCircleInfo()可以修改为：

```
void DisplayCircleInfo()
{
    cout<<"圆心为:"<<m_center.m_x<<"，"<<m_center.m_y<<endl;
    cout<<"半径为 :"<<m_radius<<endl;
}
```

在类外，如果对象成员是公有成员，可以通过对象名或对象指针直接访问对象成员。