类成员的访问控制程序实例

- 【例2-8】合理设置圆类的各成员的访问控制方式
- 问题求解思路：圆类中的数据成员一般设置为私有成员，在内部对其进行维护。所以，将圆类描述圆属性的数据成员m_x、m_y和m_radius设置成私有成员，类外不能直接访问对象的数据成员。将成员函数setCenter(double x, double y)、setRadius(double radius)和getArea()，设置为公有成员，作为类对外的接口。由于类外还需要得到圆的圆心和半径的信息，因此，还需要在类中再提供getX()、getY()和getRadius()等3个接口。外界可以通过6个接口向对象发出消息，实现对圆属性的调整、求圆的面积以及获取圆的属性信息等操作。

```
#include <iostream>
using namespace std;
class Circle
{
public:
    Circle( ) //无参构造函数
    {
        m_x=0;
        m_y=0;
        m_radius=1;
    }
```

```
    //有参构造函数
    Circle(double x, double y, double radius)
    {
        m_x=x;
        m_y=y;
        m_radius=radius;
    }
     //设置圆心
    void setCenter(double x,double y)
    {
        m_x=x;
        m_y=y;
    }
```

```
void setRadius(double radius)  //设置半径
{
    m_radius=radius;
}
double getX()   //获取圆心的x坐标
{
    return m_x;
}
double getY()   //获取圆心的y坐标
{
    return m_y;
}
```

```
    double getRadius()    // 获取半径
    {
        return m_radius;
    }
    double getArea( ) //求圆面积
    {
        return 3.14 * m_radius * m_radius;
    }
private:
    double m_x,m_y;          //圆心
    double m_radius;         //半径
};
```

```
int main()    // 正确访问示例
{
    Circle circleA;
    circleA.setCenter(5,5); //设置circleA的圆心
    circleA.setRadius(15.75); //设置circleA的半径
    cout<<"圆A的圆心为：("
        <<circleA.getX()<<','<<circleA. getY()
        <<')'<<endl;
    cout<<"圆A的半径为："
        <<circleA.getRadius() <<endl;
    cout<<"圆A的面积为："
        <<circleA.getArea() <<endl;
    return 0;
}
```

```
int main()    // 错误访问示例
{
    Circle circleA;
    circleA.m_x=5;
    circleA.m_y=5;
    circleA.m_radius=15.75;
    cout<<"圆A的面积为："
        <<circleA.getArea()<<endl;
    return 0;
}
```

- 由于在类外访问了对象的私有成员m_x、m_y和m_radius，破坏了封装性，编译时会报如下错误：
- error C2248: “Circle::m_x” : 无法访问private成员 (在“Circle”中声明)
- error C2248: “Circle::m_y” : 无法访问private成员 (在“Circle”中声明)
- error C2248: “Circle::m_radius” : 无法访问private成员 (在“Circle”中声明)