

# 上机作业九

姓 名	学 号	日 期
袁宇昊	201611130126	2018.11.13

## 实验目的

- 虚函数的定义和使用。
- 抽象类的定义和使用。

## 实验总结：

请在以下总结实验中发现的问题和解决办法或心得体会。请勿黏贴过多源码。

1. **心得**：使用类函数的类内定义可以少些个cpp文件。
2. **问题**：虚析构函数的作用是啥。

```
virtual ~Circle() {}
```

**解决**：在作业中，变量都是非指针变量，所以虚析构函数的作用没有体现出来。如果是继承类中有一个指针变量，在用基类指针指向继承类的时候调用析构函数，那么普通的析构函数只会把基类成员delete掉，而继承类的指针没有被delete，造成了指针悬挂。而虚析构函数则会调用继承类的析构函数，这样的话就可以在继承类的虚析构函数里面把指针变量delete掉了。

**总结来说就是在写一个类时，将其析构函数写为虚函数总不会错的。**

3. **问题**：解决绘图问题。

**解决**：因为要同时显示形状，而且每个形状都有各自相应的位置，所以采用一个二维数组M来记录每个位置的值。而各个形状的Draw()函数就是去标记更新M，最后调用DrawM()函数输出。

由于M数组是公用的，所以必须用static来让他变成静态变量。

对于每个形状的Draw()函数的实现，就是找出在相应图形里的所有点进行标记，例如circle的Draw()函数：

```

virtual void Circle::Draw()
{
    for(int i=0;i<=Msize_y;i++)
        for(int j=0;j<=Msize_x;j++)
        {
            double xx=(i-P.x()),yy=(j-P.y());
            if(xx*xx+yy*yy<=R*R+1)//距离圆心距离小于半径
            {
                M[i][j]=1;//标记成1
            }
        }
}

```

最后的DrawM()函数，就是以M数组作为基础，输出：

```

void Shape::DrawM()
{
    for(int i=Msize_y;i>=0;i--)//从上到下
    {
        for(int j=0;j<=Msize_x;j++)//从左向右
        {
            switch(M[i][j])
            {
                case 0: //空白
                    cout<<'+';break;
                case 1://圆
                    cout<<'C';break;
                case 2://矩形
                    cout<<'R';break;
                case 3://三角形
                    cout<<'T';break;
            }
        }
        cout<<endl;
    }
}

```