

# C++语言程序设计

---

## 实验

---

王焦乐

哈尔滨工业大学（深圳）

机电工程与自动化学院

邮箱: **wangjiaole@hit.edu.cn**

# 实验四：多态性

---

## 实验内容

- 定义抽象基类Shape，由它派生出3个派生类：Circle、Triangle、Rectangle。用虚函数分别计算多个不同类图形的面积，并求它们的面积和。
  - 要求用基类指针数组，使它的每一个元素指向一个派生类的对象，使用一个循环完成对多个图像的面积计算与求和。
  - 图型的数据在定义对象时给定
  - 定义顶点Point类，并在3个派生类中使用
  - 圆以 **圆心坐标，半径** 的方式给定，其余以 **顶点坐标** 的方式给定
  - 各个派生类的构造函数中应判断给定顶点组合是否满足图形属性

---

## □ 应用实例

- 改写例6.1中的“点-圆-圆柱体”类的层次结构，使用虚函数和抽象基类。顶层是抽象基类Shape。Point、Circle、Cylinder都是Shape类的直接派生类和间接派生类。

### (1) 声明抽象基类Shape

```
class Shape
{
public:
    virtual float area() const { return 0.0; }    //虚函数
    virtual float volume() const { return 0.0; } //虚函数
    virtual void shapeName() const = 0;          //纯虚函数
};
```

- 
- 改写例6.1中的“点-圆-圆柱体”类的层次结构，使用虚函数和抽象基类。顶层是抽象基类Shape。Point、Circle、Cylinder都是Shape类的直接派生类和间接派生类。

## (2) 声明Point类

```
class Point : public Shape // Point是Shape的公用派生类
{
protected:
    float x, y;
public:
    Point(float = 0, float = 0);
    void setPoint(float, float);
    float getX() const { return x; }
    float getY() const { return y; }
    virtual void shapeName() const { cout << "Point:"; } // 对纯虚函数进行定义
    friend ostream &operator<<(ostream &, const Point &);
};
```

---

- 
- 改写例6.1中的“点-圆-圆柱体”类的层次结构，使用虚函数和抽象基类。顶层是抽象基类Shape。Point、Circle、Cylinder都是Shape类的直接派生类和间接派生类。

### (3) 声明Circle类

```
class Circle : public Point // 声明Circle类
{
protected:
    float radius;

public:
    Circle(float x = 0, float y = 0, float r = 0);
    void setRadius(float);
    float getRadius() const;
    virtual float area() const;
    virtual void shapeName() const { cout << "Circle: "; } // 对纯虚函数进行再定义
    friend ostream &operator<<(ostream &, const Circle &);
};
```

---

- 
- 改写例6.1中的“点-圆-圆柱体”类的层次结构，使用虚函数和抽象基类。顶层是抽象基类Shape。Point、Circle、Cylinder都是Shape类的直接派生类和间接派生类。

### (3) 声明Cylinder类

```
class Cylinder : public Circle // 声明Cylinder类
{
protected:
    float height;
public:
    Cylinder(float x = 0, float y = 0, float r = 0, float h = 0);
    void setHeight(float);
    float getHeight() const;
    virtual float area() const;
    virtual float volume() const;
    virtual void shapeName() const { cout << "Cylinder:"; } // 对纯虚函数进行再定义
    friend ostream &operator<<(ostream &, const Cylinder &);
};
```

---