

南京邮电大学

实验报告

(2023 / 2024 学年 第 二 学期)

课程名称	面向对象程序设计及 C++
实验名称	实验三： 多态性实验
实验时间	2024 年 6 月 4 日
指导单位	计算机学院、软件学院、网络空间安全学院
指导教师	吴家皋

学生姓名	于明宏	班级学号	B23041011
学院(系)	计软网安学院	专 业	信息安全

实 验 报 告

实验名称	多态性实验		
实验类型	综合	实验学时	2
<p>一、 实验目的和要求</p> <p>(1) 熟悉类的设计、运用继承与派生机制设计派生类，合理设置数据成员和成员函数。</p> <p>(2) 掌握双目运算符、单目运算符的重载方法，对常用算术运算符能在自定义类中通过友元函数、成员函数进行重载，以实现静态多态性。</p> <p>(3) 掌握通过继承、虚函数、基类的指针或引用实现动态多态性的方法。</p> <p>(4) 理解并掌握有纯虚函数的抽象类的作用，在各派生类中重新定义各纯虚函数的方法，以及此时实现的动态多态性。</p>			
<p>二、实验环境(实验设备)</p> <p>硬件： 微型计算机</p> <p>软件： Windows 操作系统、Microsoft Visual Studio 2010</p>			
<p>三、实验原理及内容</p> <p>实验题目 1: 定义点类 Point，有两个 double 类型的数据成员 x 和 y，分别表示横坐标和纵坐标，要求完成如下内容。</p> <p>(1) 定义坐标默认值为原点(0.0,0.0)的构造函数。</p> <p>(2) 以成员函数形式重载：前置“++”运算符和双目运算符“-”。</p> <p>(3) 用友元函数形式重载：双目运算符“+”（两种版本，详见实验指导部分）、插入运算符。</p> <p>(4) 先根据 main()主函数代码和运行结果，补充类的定义和相关函数的定义，写出完整程序。</p> <p>(5) 程序正确后，删除 main()函数体，根据运行结果，自己重新完成 main()函数。</p> <p>main()主函数代码如下。(中文五号宋体，英文五号 Consolas 字体，单倍行距)</p> <pre>int main() { Point pt1(10.5,20.8),pt2(-5.3,18.4),pt3; cout<<"original pt1,pt2,pt3 are:\n"; cout<<pt1<<pt2<<pt3; pt3=pt1+100.8; cout<<"after pt3=pt1+100.8, pt3 is:"<<pt3; pt3=pt1+pt2; cout<<"after pt3=pt1+pt2, pt3 is:"<<pt3; pt3=++pt1;</pre>			

实验报告

```
++pt2;  
cout<<"after ++ pt1,pt2,pt3 are:\n";  
cout<<pt1<<pt2<<pt3;  
pt3=pt1-pt2;  
cout<<"after pt3=pt1-pt2, pt3 is:"<<pt3;  
return 0 ;  
}
```

程序运行结果如下。

```
original pt1,pt2,pt3 are:  
(10.5,20.8)  
(-5.3,18.4)  
(0,0)  
after pt3=pt1+100.8, pt3 is:(111.3,121.6)  
after pt3=pt1+pt2, pt3 is:(5.2,39.2)  
after ++ pt1,pt2,pt3 are:  
(11.5,21.8)  
(-4.3,19.4)  
(11.5,21.8)  
after pt3=pt1-pt2, pt3 is:(15.8,2.4)
```

实验解答:

(1) 类 Point 的构造函数:

```
Point::Point(double x0, double y0) {  
    x = x0;  
    y = y0;  
}
```

(2) 用成员函数重载: 前置 “++” 运算符和双目运算符 “-”:

```
Point Point::operator++() {  
    ++x;  
    ++y;  
    return *this;  
}
```

```
Point Point::operator-(const Point& a) {  
    Point temp;  
    temp.x = x - a.x;  
    temp.y = y - a.y;  
    return temp;  
}
```

(3) 用友元函数形式重载: 双目运算符 “+” (两种版本, 详见实验指导部分)

```
Point operator+(const Point& a, const Point& b) {  
    Point temp;
```

实验报告

```
temp.x = a.x + b.x;
temp.y = a.y + b.y;
return temp;
}

Point operator+(const Point& a, double n) {
    Point temp;
    temp.x = a.x + n;
    temp.y = a.y + n;
    return temp;
}
```

(4) 友元函数重载插入运算符:

```
ostream& operator<<(ostream& out, const Point& point) {
    out << "(" << point.x << ", " << point.y << ")" << endl;
    return out;
}
```

(5) 程序正确后, 根据运行结果, 重新完成的 main() 函数:

```
int main() {
    Point pt1(10.5, 20.8), pt2(-5.3, 18.4), pt3;
    cout << "original pt1,pt2,pt3 are:\n";
    cout << pt1 << pt2 << pt3;
    pt3 = pt1 + 100.8;
    cout << "after pt3=pt1+100.8, pt3 is:" << pt3;
    pt3 = pt1 + pt2;
    cout << "after pt3=pt1+pt2, pt3 is:" << pt3;
    pt3 = ++pt1;
    ++pt2;
    cout << "after ++ pt1,pt2,pt3 are:\n";
    cout << pt1 << pt2 << pt3;
    pt3 = pt1 - pt2;
    cout << "after pt3=pt1-pt2, pt3 is:" << pt3;
    return 0;
}
```

实验题目 2: 定义一个抽象类容器类, 其中定义了若干纯虚函数, 实现求表面积、体积、输出等功能。由此抽象类派生出正方体、球体和圆柱体等多个派生类, 根据需要定义自己的成员变量, 在各个派生类中重新定义各纯虚函数, 实现各自类中相应功能, 各个类成员的初始化均由本类构造函数实现。

① 在主函数中, 定义容器类的指针和各个派生类的对象, 使指针指向不同对象处调用相同的函数能执行不同的函数代码, 从而实现动态多态性。

② 定义一个顶层函数 void TopPrint(Container &r);使得主函数中调用该函数时, 根据实在参数所有的类自动调用对应类的输出函数。

实 验 报 告

③ 主函数中定义一个 Container 类对象，观察编译时的错误信息，从而得出什么结论？

实验解答：

(1) 基类 Container 的定义见实验教材。

(2) 各个派生类的定义，根据提示进行填写完整代码：

① //正方体类，从 Container 类公有继承，定义构造函数，重新定义基类的 3 个纯虚函数

```
class Cube : public Container {
protected:
    double length;
public:
    Cube(double l) : Container(0), length(l) {}
    double area() {
        return 6 * length * length;
    }
    double volume() {
        return length * length * length;
    }
    void print() {
        cout << "Cube" << endl;
        cout << "Length: " << length << endl;
        cout << "Cube Surface Area: " << area() << endl;
        cout << "Cube Volume: " << volume() << endl << endl;
    }
};
```

② //球类，从 Container 类公有继承，定义构造函数，重新定义基类的 3 个纯虚函数

```
class Sphere : public Container {
public:
    Sphere(double r) : Container(r) {
        radius = r;
    }
    double area() {
        return 4 * PI * radius * radius;
    }
    double volume() {
        return 4.0 * PI * radius * radius * radius / 3.0;
    }
    void print() {
        cout << "Sphere" << endl;
        cout << "Radius: " << radius << endl;
        cout << "Sphere Surface Area: " << area() << endl;
        cout << "Sphere Volume: " << volume() << endl << endl;
    }
};
```

实 验 报 告

```
}  
};
```

③ //圆柱体类，从 Container 类公有继承，需要增加的成员变量，定义构造函数，重新定义基类的三个纯虚函数

```
class Cylinder : public Container {  
protected:  
    double height;  
public:  
    Cylinder(double radius, double h) : Container(radius), height(h) {}  
    double area() {  
        return 2 * PI * radius * height + radius * radius * 2 * PI;  
    }  
    double volume() {  
        return PI * radius * radius * height;  
    }  
    void print() {  
        cout << "Cylinder" << endl;  
        cout << "Height: " << height << " Radius: " << radius << endl;  
        cout << "Cylinder Surface Area: " << area() << endl;  
        cout << "Cylinder Volume: " << volume() << endl;  
    }  
};
```

(3) 正确定义各派生类对象，记录程序的运行结果是：

正方体对象: Cube cu(13);

球体对象: __Sphere sp(14);

圆柱体对象: Cylinder cy(13, 14);

运行结果:

Cube

Length: 13

Cube Surface Area: 1014

Cube Volume: 2197

Sphere

Radius: 14

Sphere Surface Area: 2461.76

Sphere Volume: 11488.2

Cylinder

Height: 14 Radius: 13

Cylinder Surface Area: 2204.28

Cylinder Volume: 7429.24

实 验 报 告

(4) 主函数中定义一个 Container 类对象，编译器的报错信息：

不允许强制转换到抽象类 “Container”

试说明原因：

含有纯虚函数的类是抽象类，无法实例化对象。

四、实验小结（包括问题和解决方法、心得体会、意见与建议等）

（中文五号宋体，英文五号 Consolas 字体，单倍行距）

（一）实验中遇到的主要问题及解决方法

1. 在题目（2）中在主函数中定义 Container 类的对象，**会产生**报错信息，解释原因。
含有纯虚函数的类是抽象类，无法实例化对象。

2. 通过题目（2），你觉得纯虚函数与抽象类在编程中有什么价值和意义？

纯虚函数和抽象类在编程中的价值在于提供了一种规范和约束，强制子类实现特定方法，促进了代码的结构化和模块化，同时也支持多态性，增强了代码的灵活性和可维护性。

3. 在题目（1）中通过代码验证，请总结友元函数与成员函数在实现运算符重载时的区别。
形参个数不同；类外定义时声明部分有区别；显示方式的调用不同；第一运算对象不同；第二运算对象有区别。

4. 其它问题及解决方法：

对于双目运算符、单目运算符的重载方法掌握不够到位，对常用算术运算符能在自定义类中通过友元函数、成员函数进行重载的方法使用不够熟练，通过多翻阅参考书籍，使得问题得以解决。

（二）实验心得

这次实验让我深刻领悟到了面向对象编程的核心概念，特别是继承、多态和重载。在设计派生类时，我学会了如何合理设置数据成员和成员函数，以及如何运用继承与派生机制构建出更加模块化和可复用的代码结构。同时，在重载运算符方面，我掌握了重载双目运算符和单目运算符的方法，通过友元函数或成员函数实现常用算术运算符的重载，从而实现了静态多态性。

另外，通过学习继承、虚函数以及基类的指针或引用，我了解到如何实现动态多态性。这使得我能够在运行时根据对象的实际类型来调用相应的函数，而不需要在编译时确定。这种灵活性为我的代码带来了更大的通用性和可扩展性，也提高了代码的可维护性。这次实验让我更加深入地理解了面向对象编程的精髓，并为我未来在软件开发领域的发展打下了坚实的基础。

（三）意见与建议（没有可省略）

可以提供更多的时间上机操作，以确保更多程序设计思路得以实现，提升面向对象语言的掌握程度和编程能力。

实 验 报 告

五、支撑毕业要求指标点

信息安全：

1.2-M 掌握计算机软硬件相关工程基础知识，能将其用于分析信息安全领域的相关工程问题。

3.1-H 掌握信息安全领域所涉及的软硬件系统，从数字电路、计算机系统、到各类系统软件的基本理论与设计结构。

六、指导教师评语

评 分 细 则	评分项	优秀	良好	中等	合格	不合格
	遵守实验室规章制度					
	学习态度					
	算法思想准备情况					
	程序设计能力					
	解决问题能力					
	算法设计合理性					
	算法效能评价					
	报告书写认真程度					
	内容详实程度					
	文字表达熟练程度					
	其它评价意见					
	本次实验能力达成评价 (总成绩)		批阅人		日期	