课程名称：面对象程序设计与应用　指导教师：张潇

班级：信工 16-2 班　姓名：王璐　学号：1610480214

实验项目名称：

实验四 多态性

实验目的及要求：

1. 掌握友元的概念，理解友元函数和友元类。
2. 掌握运算符重载，掌握运算符重载为成员函数和友元函数的方法。
3. 理解多态的概念，理解函数的静态联编和动态联编。
4. 掌握虚函数的定义，理解虚函数在类的继承层次中的作用、虚函数的引入对程序运行时的影响。
5. 理解纯虚函数的概念并掌握它的使用方法，了解并应用抽象类。

实验原理：

C++面对象程序设计的原理，友元函数的实现方法，运算符重载的原理，虚函数的定义和使用方法等

实验内容（方法和步骤）：

验证性题目：

1.（1）源代码:

```
#include <iostream>
#include <string>
using namespace std;
class Employee{
    public:
        Employee(string Name,string id){name=Name; Id=id;}
        string getName(){return name;}
        string getID(){return Id;}
        float getSalary(){return 0.0;}
        void print(){cout<<"姓名："<<name<<"\t\t 编号："<<Id<<endl;}
    private:
        string name;
        string Id;
};
class Manager:public Employee{
    public:
        Manager(string Name,string id,float s=0.0):Employee(Name,id){
            weeklySalary=s;
```

```
        }
        void setSalary(float s){weeklySalary=s;}
        float getSalary(){return weeklySalary;}
        void print(){
            cout<<"经理："<<getName()<<"\t\t 编号："<<getID()<<"\t\t 周工资"<<getSalary()<<endl;
        }
    private:
        float weeklySalary;
};
int main(){
    Employee e("黄春秀","NO0009"),*pM;
    Manager m("刘大海","NO0001",128);
    m.print();
    pM=&m;
    pM->print();
    Employee &rM=m;
    rM.print();
    return 0;
}
```

（2）源代码：

```
#include<iostream>
#include <cstring>
using namespace std;
class Sales{
    private:
        char name[10];
        char id[18];
        int age;
    public:
        Sales(char *Name,char *ID,int Age);
        friend ostream &operator<<(ostream &os,Sales &s);
        friend istream &operator>>(istream &is,Sales &s);

};
Sales::Sales(char *Name,char *ID,int Age){
    strcpy(name,Name);
    strcpy(id,ID);
    age=Age;
}
ostream& operator<<(ostream &os,Sales &s){
    os<<s.name<<"\t";
    os<<s.id<<"\t";
    os<<s.age<<endl;
```

```
    return os;
}
istream &operator>>(istream &is,Sales &s){
    cout<<"输入雇员的姓名，身份证号，年龄"<<endl;
    is>>s.name>>s.id>>s.age;
    return is;
}
int main()
{
    Sales s1("杜康","214198012111711",40);
    cout<<s1;
    cout<<endl;
    cin>>s1;
    cout<<s1<<endl;
    return 0;
}
```

（3）源代码：

```
#include <iostream>
using namespace std;
class Base{
    protected:
        int n;
    public:
        Base(int m){n=m++;}
        virtual void g1(){cout<<"Base::g1()..."<<n<<endl; g4();}
        virtual void g2(){cout<<"Base::g2()..."<<++n<<endl; g3();}
        virtual void g3(){cout<<"Base::g3()..."<<++n<<endl; g4();}
        virtual void g4(){cout<<"Base::g4()..."<<++n<<endl; }
};
class Derive:public Base{
        int j;
    public:
        Derive(int n1,int n2):Base(n1){j=n2;}
        void g1(){cout<<"Deri::g1()..."<<++n<<endl;g2();}
        void g3(){cout<<"Deri::g2()..."<<++n<<endl;g4();}
};
int main()
{
    Derive Dobj(1,0);
    Base Bobj=Dobj;
    Bobj.g1();
    cout<<"-------------------------"<<endl;
    Base *bp=&Dobj;
```

```
    bp->g1();
    cout<<"-------------------------"<<endl;
    Base &bobj2=Dobj;
    bobj2.g1();
    cout<<"-------------------------"<<endl;
    Dobj.g1();
    return 0;
}
```

（4）第一题，修改后的源代码：（增加了拷贝构造函数）

```
#include <iostream>
#include <cstring>
using namespace std;
class X{
    private:
        char *s;
    public:
        X(const char *b){
            s=new char[sizeof(b)+1];
            strcpy(s,b);
        }
        X(const X &p);
        ~X(){delete s;}
        void display(){cout<<"s="<<s<<endl;}
};
X::X(const X &p){
    s=new char[strlen(p.s)+1];
    strcpy(s,p.s);
    }
int main(){
    X x1("ok");
    X x2(x1);
    X x3=x1;
    x2.display();
    x3.display();
    return 0;
}
```

（4）第二题，修改后的源代码：（修改了返回值为引用）

```
#include <iostream>
using namespace std;
class Number{
    int n;
public:
    Number(int x):n(x){};
```

```
    Number& operator++(){++n;return *this;}
    Number& operator++(int){n++; return *this;}
    friend Number &operator--(Number &o);
    friend Number &operator--(Number &o,int);
    void display(){cout<<"This Number is:"<<n<<endl;}
};
Number &operator--(Number &o){--o.n;return o;}
Number &operator--(Number &o,int){o.n--; return o;}
int main(){
    Number N1(10);
    ++ ++ ++N1;
    N1.display();
    N1++;
    N1.display();
    --N1;
    N1.display();
    N1-- -- --;
    N1.display();
    return 0;
}
```

2.设计性题目

（1）源代码:

Shape.h:

```
#ifndef SHAPE_H
#define SHAPE_H
class Shape{
    public:
        virtual double area() const=0;
        virtual double volume() const=0;
        virtual void printShapeName() const=0;
        virtual void print() const=0;
};
#endif
```

Point.h:

```
#include <iostream>
using namespace std;
#include "Shape.h"
#ifndef POINT_H
#define POINT_H
class Point : public Shape{
    public:
        Point(int = 0,int = 0);
```

```
        void setPoint(int,int);
        int getX() const{return x;}
        int getY() const{return y;}
        virtual double area() const;
        virtual double volume() const;
        virtual void printShapeName() const{cout<<"Point :";}
        virtual void print() const;
    private:
        int x,y;
};
#endif
```

Point.cpp:

```
#include "Point.h"
double Point::area()const{return 0;}
double Point::volume()const{return 0;}
Point::Point(int a,int b){setPoint(a,b);}
void Point::setPoint(int a,int b){x=a;y=b;}
void Point::print() const{cout<<"["<<x<<","<<y<<"]";}
```

Cylinder.h:

```
#ifndef CYLINDE_H
#define CYLINDE_H
#include "Circle.h"
class Cylinder:public Circle{
public:
    Cylinder(double h=0.0,double r=0.0,int x=0,int y=0);
    void setHeight(double);
    double getHeight();
    virtual double area() const;
    virtual double volume() const;
    virtual void printShapeName() const{cout<<"Cylinder:";}
    virtual void print() const;
private:
    double height;
};
#endif
```

Cylinder.cpp:

```
#include "Cylinder.h"
Cylinder::Cylinder(double h,double r,int x,int y):Circle(r,x,y){setHeight(h);}
void Cylinder::setHeight(double h){height=h>0?h:0;}
double Cylinder::getHeight(){return height;}
double Cylinder::area() const{
```

```
    return 2*Circle::area()+2*3.14159*getRadius()*height;
}
double Cylinder::volume()const{return Circle::area()*height;}
void Cylinder::print() const{
    Circle::print();
    cout<<";Height="<<height;
}
```

Circle.h:

```
#ifndef CIRCLE_H
#define CIRCLE_H
#include "Point.h"
class Circle:public Point{
    public:
        Circle(double r=0.0,int x=0,int y=0);
        void setRadius(double);
        double getRadius() const;
        virtual double area() const;
        virtual void printShapeName() const{cout<<"Circle:";}
        virtual void print() const;
    private:
        double radius;
};
#endif
```

Circle.cpp:

```
#include "Circle.h"
Circle::Circle(double r,int a,int b):Point(a,b){setRadius(r);}
void Circle::setRadius(double r){radius=r>0?r:0;}
double Circle::getRadius()const{return radius;}
double Circle::area() const{return 3.14159*radius*radius;}
void Circle::print()const{
    Point::print();
    cout<<";Radius="<<radius;
}
```

Main.cpp:

```
#include <iostream>
#include <iomanip>
#include "Shape.h"
#include "Point.cpp"
#include "Circle.cpp"
#include "Cylinder.cpp"
using namespace std;
```

```
void vpf(const Shape *bptr){
	bptr->printShapeName();
	bptr->print();
	cout<<"\nArea="<<bptr->area()<<"\nVolume="<<bptr->volume()<<endl<<endl;
}
void vrf(const Shape &bref){
	bref.printShapeName();
	bref.print();
	cout<<"\nArea="<<bref.area()<<"\nVolume="<<bref.volume()<<endl<<endl;
}
int main(){
	cout << setiosflags(ios::fixed|ios::showpoint)<<setprecision(2);
	Point point(7,11);
	Circle circle(3.5,22,8);
	Cylinder cylinder(10,3.3,10,10);
	Shape *array0fShapes[3];
	array0fShapes[0]=&point;
	array0fShapes[1]=&circle;
	array0fShapes[2]=&cylinder;
	cout<<"-----通过基类指针访问虚函数-----"<<endl;
	for(int i=0;i<3;i++)
		vpf(array0fShapes[i]);
	cout<<"-----通过基类引用访问虚函数-----"<<endl;
	for(int j=0;j<3;j++)
		vrf(*array0fShapes[j]);
	return 0;
}
```

(2)源代码：

```
#include <iostream>
using namespace std;
class Calculator {
	unsigned short int count;
	public:
	void display(){cout<<"count:"<<count<<endl;}
	Calculator(unsigned short int x):count(x){};
	Calculator(){count=0;}
	Calculator &operator++(){
		++count;
		return *this;
	}
	Calculator operator++(int){
		Calculator t(*this);
```

```
        count++;
        return t;
    }
    Calculator &operator--(){
        --count;
        return *this;
    }
    Calculator operator--(int){
        Calculator t(*this);
        count--;
        return t;
    }
    Calculator operator+(const Calculator p){
        unsigned short temp;
        temp=p.count+count;
        return temp;
    }
    Calculator operator-(const Calculator d){
        unsigned short temp;
        temp=count-d.count;
        return temp;
    }
    Calculator &operator=(const Calculator &s){
        count=s.count;
    }
};
int main()
{
    Calculator door1,door2(20);
    for(int i=0;i<100;i++)
        door1++;
    door2++;
    door1.display();
    door2.display();
    for(int i=0;i<5;i++)
        door1--;
    door2--;
    door1.display();
    door2.display();
    --door1;
    door1.display();
    --door2;
    door2.display();
    Calculator door3;
```

```
    door3=door1+door2;
    door3.display();
    door3=door1-door2;
    door3.display();
    return 0;
}
```

实验结果与分析：

在写程序的过程中遇到了一些问题，最终经过查资料，解决了问题。

图片结果：

```
root@iZwz90po3b1w65lisi02d2Z:/home/something/cppe4# ./1
经理：刘大海              编号：N00001              周工资128
姓名：刘大海              编号：N00001
姓名：刘大海              编号：N00001
```

```
root@iZwz90po3b1w65lisi02d2Z:/home/something/cppe4# ./2
杜康    214198012111711 40

输入雇员的姓名，身份证号，年龄
玩海波 111333199999440 23
玩海波  111333199999440 23
```

```
root@iZwz90po3b1w65lisi02d2Z:/home/something/cppe4# ./3
Base::g1()...1
Base::g4()...2
------------------------
Deri::g1()...2
Base::g2()...3
Deri::g2()...4
Base::g4()...5
------------------------
Deri::g1()...6
Base::g2()...7
Deri::g2()...8
Base::g4()...9
------------------------
Deri::g1()...10
Base::g2()...11
Deri::g2()...12
Base::g4()...13
```

```
root@iZwz90po3b1w65lisi02d2Z:/home/something/cppe4# ./4_1
s=ok
s=ok
```

```
root@iZwz90po3b1w65lisi02d2Z:/home/something/cppe4# ./4_2
This Number is:13
This Number is:14
This Number is:13
This Number is:10
```

```
root@iZwz90po3b1w65lisi02d2Z:/home/something/cppe4/5# ./door
-----通过基类指针访问虚函数-----
Point :[7,11]
Area=0.00
Volume=0.00

Circle:[22,8];Radius=3.50
Area=38.48
Volume=0.00

Cylinder:[10,10];Radius=3.30;Height=10.00
Area=275.77
Volume=342.12

-----通过基类引用访问虚函数-----
Point :[7,11]
Area=0.00
Volume=0.00

Circle:[22,8];Radius=3.50
Area=38.48
Volume=0.00

Cylinder:[10,10];Radius=3.30;Height=10.00
Area=275.77
Volume=342.12
```

```
D:\User_Data\OneDrive\something\cppe4\6.exe
count:100
count:21
count:95
count:20
count:94
count:19
count:113
count:75

Process returned 0 (0x0)   execution time : 0.021 s
Press any key to continue.
```

成绩：　　批阅教师签名：　　年　　月　　日