

浙江理工大学信息电子学院

实验指导书

实验名称：类的多态性的实现

学时安排：3

实验类别：设计性实验

实验要求：1 人 1 组

学号：2012329620003

姓名：李畅

一、实验目的

1. 理解重载运算符的意义。
2. 掌握使用成员函数、友员函数重载运算符的特点。
3. 掌握重载运算符函数的调用方法。
4. 掌握动态联编的概念。
5. 掌握虚函数和纯虚函数的使用方法。

二、实验原理介绍

设计性实验

具体原理请见实验内容和步骤

实现对抽象类的继承，通过 operator 函数调用的形式，实现运算符的重载

三、实验设备介绍

软件需求：Visual C++ 6.0

硬件需求：对于硬件方面的要求，建议配置是 Pentium III 450 以上的 CPU 处理器，64MB 以上的内存，200MB 的自由硬盘空间、CD-ROM 驱动器、能支持 24 位真彩色的显示卡、彩色显示器、打印机。

四、实验内容

1. 某公司有老板 Boss、雇员 Employee、小时工 Hourly Worker 和营销人员 CommWorker，他们的薪金计算方法如下：

老板实行年薪制，如一年 15 万；雇员按月计酬，方法是：基本工资+奖金；小时工按工作时间计算报酬，方法是：工作小时*每小时单价；营销人员按月计

酬，方法是：基本工资+销售利润*5%。

每类人员都有姓名、职工编号、年龄、性别、工资等数据，各类人员使用统一接口 `getpay()` 计算各类人员的应得报酬，`print()` 打印输出各位工作人员的基本数据。设计一个统计并输出该公司每个人员当月薪金情况的报表类 `Report`，该类提供 `add` 接口向 `Report` 类的数组容器中添加每个员工的信息，并提供 `print` 接口用于输出每个员工的姓名、职工编号和其当月获得的报酬。在主函数中对实现的类进行测试，创建各类人员对象，通过 `Report` 类的 `add` 接口向报表中添加这些人员信息，然后通过 `Report` 类的 `print` 接口打印报表。

2. 设计一个三角形类 `Triangle`，包含三角形三条边长的私有数据成员，另定义一个重载运算符“+”，以实现求任意多个三角形的面积之和，和重载运算符“<<”，实现三角形面积的输出，并在主函数中对实现的类进行测试。

五、注意事项和要求

要求学生要提前准备实验的内容

实验完成后要求写出实验报告

六、参考书目

【C++ Prime 题解】侯捷译 华中科技大学出版社

【C++ 程序设计与应用】张耀仁著 华中科技大学出版社

七、实验过程

(1)

```
#include "iostream"
#include "cstdlib"
#include "string"
#include "list"
using namespace std;
class People{
protected:
    string name;
    string sex;
    int age;
    int ID;
    double pay;
public:
    People(string name=0, string sex=0, int age=0, int ID=0, double pay=0) {
        this->name=name;
```

```

        this->sex=sex;
        this->age=age;
        this->ID=ID;
        this->pay=pay;
        this->pay=getpay();
    }
    string getName() {return name;}
    string getSex() {return sex;}
    int getAge() {return age;}
    int getID() {return ID;}
    virtual double getpay() {return 0;}
    virtual void print() {}
};

class Boss:public People{
public:
    Boss(string name, string sex, int age, int ID, double pay):People(name, sex, age, ID, pay) {
        this->name=name;
        this->sex=sex;
        this->age=age;
        this->ID=ID;
        this->pay=pay;
        this->pay=getpay();
    }
    double getpay() {return pay/12;}
    void print() {
        cout<<"BOSS:
"<<"\t"<<name<<"\t"<<sex<<"\t"<<age<<"\t"<<ID<<"\t"<<pay<<endl;
    }
};

class Employee:public People{
    double reward;
public:
    Employee(string name, string sex, int age, int ID, double pay, double
reward):People(name, sex, age, ID, pay) {
        this->name=name;
        this->sex=sex;
        this->age=age;
        this->ID=ID;
        this->pay=pay;
        this->reward=reward;
        this->pay=getpay();
    }
    double getpay() {return pay+reward;}
    void print() {

```

```

        cout<<"Employee:"<<"\t"<<name<<"\t"<<sex<<"\t"<<age<<"\t"<<ID<<"\t"<<pay<<endl;
    }
};

class HourlyWorker:public People{
    double dayhour;
    int day;
public:
    HourlyWorker(string name,string sex,int age,int ID,double pay,double dayhour,int
day):People(name,sex,age,ID,pay){
        this->name=name;
        this->sex=sex;
        this->age=age;
        this->ID=ID;
        this->pay=pay;
        this->dayhour=dayhour;
        this->day=day;
        this->pay=getpay();
    }
    double getpay(){return pay*dayhour*day;}
    void print(){

        cout<<"HourlyWorker:"<<"\t"<<name<<"\t"<<sex<<"\t"<<age<<"\t"<<ID<<"\t"<<pay<<endl;
    }
};

class Marketers:public People{
    double profit;
public:
    Marketers(string name,string sex,int age,int ID,double pay,double
profit):People(name,sex,age,ID,pay){
        this->name=name;
        this->sex=sex;
        this->age=age;
        this->ID=ID;
        this->pay=pay;
        this->profit=profit;
        this->pay=getpay();
    }
    double getpay(){return pay+profit*0.05;}
    void print(){

        cout<<"Marketers:"<<"\t"<<name<<"\t"<<sex<<"\t"<<age<<"\t"<<ID<<"\t"<<pay<<endl;
    }
};

class Report{

```

```

public:
    void add(list<People *> &people, People *p) {
        people.push_back(p);
    }
    void print(list<People *> people) {
        list<People *>::iterator point;
        for(point=people.begin();point!=people.end();point++) {
            (*point)->print();
        }
    }
};

void main()
{
    list<People *> people;
    People (*a)=new Boss("刘大海", "男", 40, 1001, 150000);
    People (*b)=new Employee("黄春秀", "女", 30, 1501, 2000, 500);
    People (*c)=new HourlyWorker("王小虎", "男", 25, 2001, 15, 8, 20);
    People (*d)=new Marketers("李小龙", "男", 28, 2501, 2000, 5000);    Report re;
    re.add(people, a);
    re.add(people, b);
    re.add(people, c);
    re.add(people, d);
    cout<<"*****"
    ***"<<endl;
    cout<<"TYPE"<<" NAME " <<"\t"<<"SEX"<<"\t"<<"AGE"<<"\t"<<"ID"<<"\t"<<"SALARY"<<endl;
    cout<<"*****"
    ***"<<endl;
    re.print(people);
    system("pause");
}

```

(2)

```

#include"iostream"
#include"cstdlib"
#include"math.h"
using namespace std;
class Triangle{
private:
    double a;
    double b;
    double c;
    double p;

```

```

        double area;
public:
    Triangle() {} ;
    Triangle(double l1, double l2, double l3) {
        a=l1;
        b=l2;
        c=l3;
        p=(a+b+c)/2;
        area=sqrt(p*(p-a)*(p-b)*(p-c));
    }
    double getarea() {return area;}
    friend Triangle operator+(Triangle m, Triangle n);
    friend ostream &operator<<(ostream &os, Triangle t);
};
Triangle operator+(Triangle m, Triangle n) {
    Triangle t;
    t.area=m.area+n.area;
    return t;
}
ostream &operator<<(ostream &os, Triangle t) {
    os<<"the area of triangle(triangles) is "<<t.getarea()<<endl;
    return os;
}
void main()
{
    Triangle t1(3, 5, 7), t2(4, 4, 6), t3(5, 5, 8);
    cout<<t1<<endl;
    cout<<t2<<endl;
    cout<<t3<<endl;
    cout<<t1+t2<<endl;
    cout<<t1+t2+t3<<endl;
    system("pause");
}

```

八、实验结果

```
F:\c++\报表\Debug\报表.exe
*****
TYPE          NAME    SEX    AGE    ID    SALARY
*****
BOSS:         刘大海  男     40     1001  12500
Employee:     黄春秀  女     30     1501  2500
HourlyWorker: 王小虎  男     25     2001  2400
Marketers:    李小龙  男     28     2501  2250
请按任意键继续. . .
```

2、

```
F:\c++\三角形\Debug\三角形.exe
the area of triangle<triangles> is 6.49519
the area of triangle<triangles> is 7.93725
the area of triangle<triangles> is 12
the area of triangle<triangles> is 14.4324
the area of triangle<triangles> is 26.4324
请按任意键继续. . .
```

九、实验心得

- 1、要充分了解类和各组成部分之间关系，并学会灵活运用。
- 2、要学会所学重载运算符的用法，并了解原理
- 3、实验室设计代码是要理清思路，明确步骤