实验九 多态性

学号 20181101047 姓名 范少帅 班级 18级计科3班

一、实验目的

1. 掌握掌握用虚函数实现动态联编;
2. 理解静态多态性与动态多态性的区别与实现机制
3. 掌握抽象类的概念与设计方法。

二、实验内容

1、编写一个程序实现小型公司的工资管理。该公司主要有4类人员，即经理、

技术人员、销售员、销售经理，这些人员都是职员，有编号、姓名、月工资信息。月工资的计算方法是经理固定月薪8000元，技术人员每小时100元，销售员按当月销售额的4%提成，销售经理既拿固定月工资5000元又拿销售提成，销售提成为所管辖部门当月销售额的5‰。

要求： 编程计算职员的月工资并显示全部信息；（多态性实现）

求公司某一个月所需发放的工资总额。 （多态性实现）

三、实验步骤 （记录）

#include <iostream>

#include <cstring>

using namespace std;

class Person

{

protected:

char name[20];

int num;

double salary;

public:

Person(char \*iname,int n)

{

strcpy(name,iname);

num = n;

}

virtual double calsalary() = 0;

virtual void show() = 0;

virtual ~Person()

{

cout<<"~Person called."<<endl;

}

};

class Manger:public Person

{

public:

Manger(char \*iname,int n ):Person(iname,n){

}

double calsalary()

{

salary = 8000;

return salary;

}

void show()

{

cout<<"职位："<<name<<" "<<"编号："<<num<<" "<<"月工资："<<this->calsalary()<<endl;

}

};

class Techor:public Person

{

private:

int hour;

public:

Techor(char \*iname,int num,int h):Person(iname,num)

{

hour = h;

}

double calsalary()

{

salary = 100\*hour;

return salary;

}

void show()

{

cout<<"职位："<<name<<" "<<"编号："<<num<<" "<<"月工资："<<this->calsalary()<<endl;

}

};

class salManger:public Person

{

private:

double sale;

public :

salManger(char \*iname,int n,double s):Person(iname,n)

{

sale = s;

}

double calsalary()

{

salary = 5000+0.005\*sale;

return salary;

}

void show()

{

cout<<"职位："<<name<<" "<<"编号："<<num<<" "<<"月工资："<<this->calsalary()<<endl;

}

};

class Saler:public Person

{

private:

double sale;

public:

Saler(char \*iname,int n,double s):Person(iname,n)

{

sale = s;

}

double calsalary()

{

salary = 0.04\*sale;

return salary;

}

void show()

{

cout<<"职位："<<name<<" "<<"编号："<<num<<" "<<"月工资："<<this->calsalary()<<endl;

}

};

int main()

{

int sum=0;

Person \*Personarr[4];

Personarr[0] = new Manger("经理",24234);

Personarr[1] = new Techor("技术人员",32434,600);

Personarr[2] = new salManger("销售经理",201811,100000);

Personarr[3] = new Saler("销售人员",201923,100000);

for(int i=0;i<4;i++)

{

sum+=Personarr[i]->calsalary();

Personarr[i]->show();

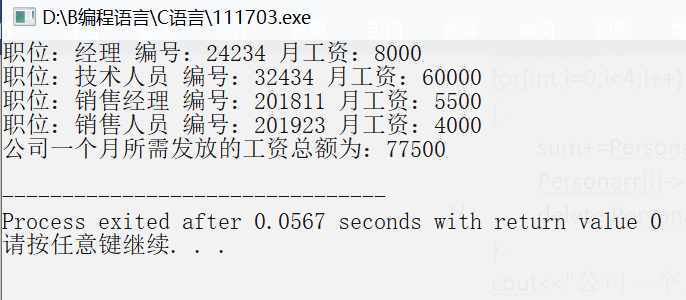
// delete Personarr[i];

}

cout<<"公司一个月所需发放的工资总额为："<<sum<<endl;

}

四、实验结果及分析



分析：每个派生类经过对纯虚函数calsalary()的重载，得到每个派生类各自的工资，之后通过累加求和将所有派生类的工资加起来得到工资总额。

五、实验总结

用virtual声明的函数为虚函数，虚函数是类的成员函数。后面加上=0为纯虚函数，一个类中若只有纯虚函数，则该类为抽象类。存在虚函数的类都有一个一维的虚函数表叫做虚表，类的对象有一个指向虚表开始的虚指针。虚表是和类对应的，虚表指针是和对象对应的。