#include<iostream>

#include<math.h>

using namespace std;

class Square {

protected:

int edge;

public:

virtual double GetArea();

Square(int edg);

~Square( ) { }

};

Square::Square(int edg){

edge=edg;

}

double Square::GetArea(){

double result;

result=edge\*edge;

return result;

}

class Rectangle:public Square{

protected:

int length;

public:

double GetArea();

Rectangle(int edg,int leng);

~Rectangle(){ }

};

double Rectangle::GetArea(){

double result;

result=length\*edge;

return result;

}

Rectangle::Rectangle(int edg,int leng):Square(edg){

length=leng;

}

double objGetArea(Square\* obj){

return obj->GetArea();

}

int main()

{

Square Sq(100);

Rectangle Rect(200,300);

cout<<"正方形Squa1(100)的面积为："<<Sq.GetArea()<<endl;

cout<<"正方形Squa1(100)的面积为："<<objGetArea(&Sq)<<endl;

cout<<"矩形Rect(200,300) 的面积为：:"<<Rect.GetArea()<<endl;

cout<<"矩形Rect(200,300) 的面积为：:"<<objGetArea(&Rect)<<endl;

Square \*s=new Rectangle(300, 400);

cout<<"不使用全局函数的方式:"<<s->GetArea()<<endl;

return 0;

}

#include <iostream>

using namespace std;

class X{

private:

int \*p;

public:

X(){p=new int[2];cout<<"X(). ";}

virtual ~X(){delete [] p;cout<<"~X(). \n";}

};

class Y:public X{

private:

int \*q;

public:

Y(){q=new int [1023];cout<<"Y():T::q="<<q<<". ";}

~Y(){delete [] q;cout<<"~Y().";}

};

int main(int argc, const char \* argv[]) {

for(int i=0;i<8;i++){

Y \*t=new Y;

X \*r=t;

delete r;

}

return 0;

}

#include <iostream>

using namespace std;

class empolyee{

protected:

char \*name; int individualEmpNo; int grade;

float accumPay; static int empolyeeNo; public:

empolyee();

virtual ~empolyee(){

if(!name)

delete [] name;

empolyeeNo--;

}

virtual void pay();

void display(); void promote(int increment);

virtual void dispalayStatus();

};

empolyee::empolyee()

{

char namestr[50];

cout<<"请输入下一个雇员的姓名:";

cin>>namestr;

name=new char[strlen(namestr)+1];

strcpy(name, namestr);

individualEmpNo= empolyeeNo++;

grade=1;

accumPay=0.0; }

void empolyee ::promote(int increment)

{

grade += increment ;

}

class technician:public empolyee{

private:

float hourlyRate;

int workHours;

public:

technician();

~technician(){}

virtual void pay();

virtual void displayStatus();

};

technician :: technician()

{ hourlyRate=100;

}

void technician::pay()

{ cout<<"请输入"<<name<<"本月的工作时数:";

cin>>workHours;

accumPay=hourlyRate\* workHours;

cout<<"兼职技术人员"<<name<<"编号"<<individualEmpNo

<<"本月工资"<<accumPay<<endl;

}

void technician::displayStatus()

{ cout<<"兼职技术人员"<<name<<"编号" << individualEmpNo << "级别为" <<grade<<

"级，已付本月工资"<<accumPay<<endl;

}

class manager:virtual public empolyee{

protected:

float monthlyPay;

public:

manager();

~manager(){}

virtual void pay();

virtual void displayStatus();

};

manager::manager()

{

monthlyPay=8000; }

void manager :: pay()

{

accumPay=monthlyPay; cout<<"经理"<<name<<"编号"<<individualEmpNo

<<"本月工资"<<accumPay<<endl;

}

void manager :: displayStatus()

{

cout<<"经理"<<name<<"编号"<<individualEmpNo

<<"级别为"<<grade<<"级，已付本月工资"<<accumPay<<endl;

}

class salesman:virtual public empolyee{

protected:

float CommRate;

float sales;

public:

salesman();

~salesman(){}

virtual void pay();

virtual void displayStatus();

};

salesman::salesman()

{

CommRate=0.04; }

void salesman::pay()

{

cout<<"请输入"<<name<<"本月的销售额:";

cin>>sales;

accumPay=sales\*CommRate;

cout<<"推销员"<<name<<"编号"<<individualEmpNo

<<"本月工资"<<accumPay<<endl;

}

void salesman::displayStatus()

{ cout<<"推销员"<<name<<"编号"<<individualEmpNo

<<"级别为"<<grade<<"级，已付本月工资"<<accumPay<<endl;

}

class salesmanager:public manager,public salesman{

public:

salesmanager();

~salesmanager(){}

virtual void pay();

virtual void displayStatus();

};

salesmanager ::salesmanager()

{ monthlyPay=5000;

CommRate= 0.005;

}

void salesmanager ::pay()

{

cout<<"请输入"<<empolyee ::name<<"所管辖部门本月的销售总额:";

cin>>sales;

accumPay=monthlyPay+CommRate\*sales; cout<<"销售经理"<<name<<"编号"<<individualEmpNo

<<"本月工资"<<accumPay<<endl;

}

void salesmanager::displayStatus()

{ cout<<"销售经理"<< name<<"编号"<<individualEmpNo

<<"级别为"<<grade<<"级，已付本月工资"<<accumPay<<endl;

}

int empolyee::empolyeeNo=0;

int main(int argc, const char \* argv[]) {

empolyee \*e[4]{ (new manager()),new manager(),new technician(),new salesman(),new salesmanager()};

e[0]->promote(3);

e[0]->pay();

e[0]->dispalayStatus();

e[1]->promote(2);

e[1]->pay();

e[1]->dispalayStatus();

e[3]->promote(2);

e[3]->pay(); e[3]->dispalayStatus(); e[2]->pay(); e[2]->dispalayStatus(); for(int i=0;i<4;i++){

delete e[i];

}

return 0;

}