**实验六 继承与派生**

### 题目1 图形类

主函数：

#include<iostream>

#include<math.h>

#include "Shape.h"

using namespace std;

#define PI 3.14

int main() {

cout << "Please input the coordinates and the radius of the circle:" << endl;

int x,y,r;

cin >> x >> y >> r;

circle a(x,y,r);

a.Show();

cout << "Please input the coordinates and the radius of the circle for the test:" << endl;

cin >> x >> y >> r;

a.Set(x, y, r);

a.Show();

cout << "Please input the length and width of the rectangle:" << endl;

int length, width;

cin >> length >> width;

rectangle b(length,width);

b.Show();

cout << "Please input the length and width of the rectangle again the test:" << endl;

cin >> length >> width;

b.Set(length, width);

b.Show();

cout << "please input the coordinates of those three points:" << endl;

int x11,y11,x22,y22,x33,y33;

cin >> x11 >> y11 >> x22 >> y22 >> x33 >> y33;

triangle c(x11, y11, x22, y22, x33, y33);

c.Show();

cout << "please input the coordinates of those three points again for the test:" << endl;

cin >> x11 >> y11 >> x22 >> y22 >> x33 >> y33;

c.Set(x11, y11, x22, y22, x33, y33);

c.Show();

return 0;

}

头文件.h

#pragma once

#include<string>

#include<string.h>

#include <iostream>

using namespace std;

class Shape {

public:

double GetArea();

void Show();

void Set();

};

class circle :public Shape //继承shape类

{

public:

circle();

circle(int x, int y, int r0);

double GetArea();

void Show();

void Set(int a, int b, int c);

private:

int x1;

int y1;

int r;

};

class rectangle :public Shape

{

public:

rectangle();

rectangle(int x, int y);

double GetArea();

void Show();

void Set(int a, int b);

private:

int length;

int width;

};

class triangle :public Shape

{

public:

triangle();

triangle(int x11, int y11, int x22, int y22, int x33, int y33);

double GetArea();

void Show();

void Set(int a, int b, int c, int d, int e, int f);

private:

int x31;

int y31;

int x32;

int y32;

int x33;

int y33;

double edgelen1;

double edgelen2;

double edgelen3;

};

头文件.cpp

#include "Shape.h"

#include <iostream>

using namespace std;

#define PI 3.14

circle::circle()//无参构造

{

x1 = 1;

y1 = 1;

r = 1;

}

circle::circle(int x, int y, int r0) {

x1 = x;

y1 = y;

r = r0;

}

double circle::GetArea()

{

double s;

s = PI;

s = s \* r \* r;

return s;

}

void circle::Show()

{

double c;

c = PI;

c = c \* 2 \* r;

cout << "(" << x1 << "," << y1 << ")" << endl;//坐标；

cout << "2R =" << 2 \* r << endl;//直径；

cout << "circum =" << c << endl;//周长；

cout << "the area of the circle:" << GetArea() << endl;

}

void circle::Set(int a, int b, int c)

{

this->x1 = a;

this->y1 = b;

this->r = c;

}

rectangle::rectangle()

{

length = 0;

width = 0;

}

rectangle::rectangle(int x0, int y0)

{

length = x0;//依次输入长宽

width = y0;

}

double rectangle::GetArea()

{

int s;

s = width \* length;

return s;

}

void rectangle::Show()

{

cout << "length = " << length << endl;

cout << " width= " << width << endl;

cout << "rectangle circum = " << 2 \* (length + width) << endl;

cout << "rectangle area = " << GetArea() << endl;

}

void rectangle::Set(int a, int b)

{

length = a;

width = b;

}

triangle::triangle() {

x31 = 0;

y31 = 0;

x32 = 0;

y32 = 0;

x33 = 0;

y33 = 0;

edgelen1 = 0;

edgelen2 = 0;

edgelen3 = 0;

}

triangle::triangle(int x11, int y11, int x22, int y22, int x34, int y34) {

x31 = x11;

y31 = y11;

x32 = x22;

y32 = y22;

x33 = x34;

y33 = y34;

}

double triangle::GetArea()

{

double s;

double p;

p = (edgelen1 + edgelen2 + edgelen3) / 2;

s = p \* (p - edgelen1) \* (p - edgelen2) \* (p - edgelen3);

s = sqrt(s);

return s;

}

void triangle::Show()

{

cout << "location:\n(" << x31 << "," << y31 << ")" << endl;

cout << "(" << x32 << "," << y32 << ")" << endl;

cout << "(" << x33 << "," << y33 << ")" << endl;

double a, b, c, c1;

a = pow((x31 - x32), 2) + pow((y31 - y32), 2);

a = sqrt(a);

b = pow((x31 - x33), 2) + pow((y31 - y33), 2);

b = sqrt(b);

c = pow((x32 - x33), 2) + pow((y32 - y33), 2);

c = sqrt(c);

c1 = a + b + c;

edgelen1 = a;

edgelen2 = b;

edgelen3 = c;

cout << "triangle circum = " << c1 << endl;

cout << "The area is :" << GetArea() << endl;

}

void triangle::Set(int a, int b, int c, int d, int e, int f)

{

x31 = a;//依次输入每个点的横纵坐标

y31 = b;

x32 = c;

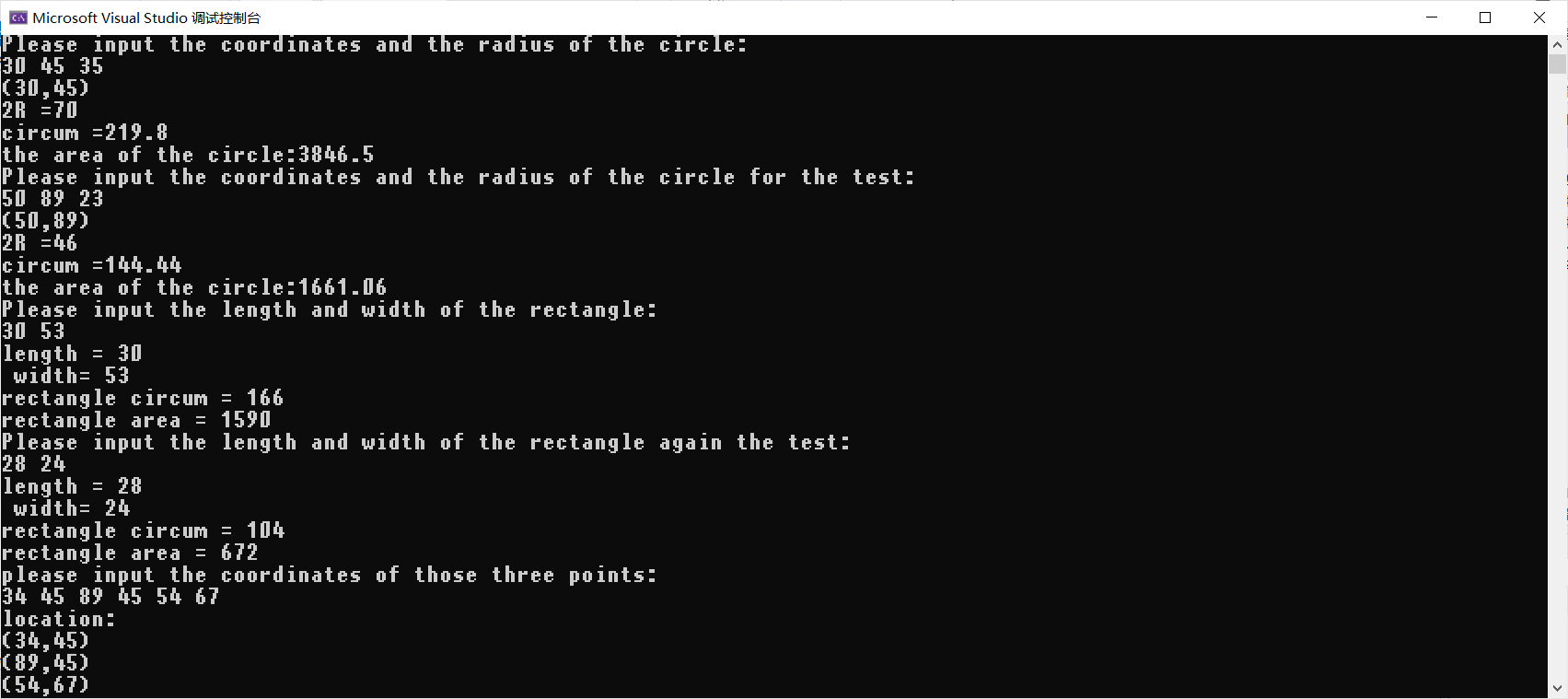
y32 = d;

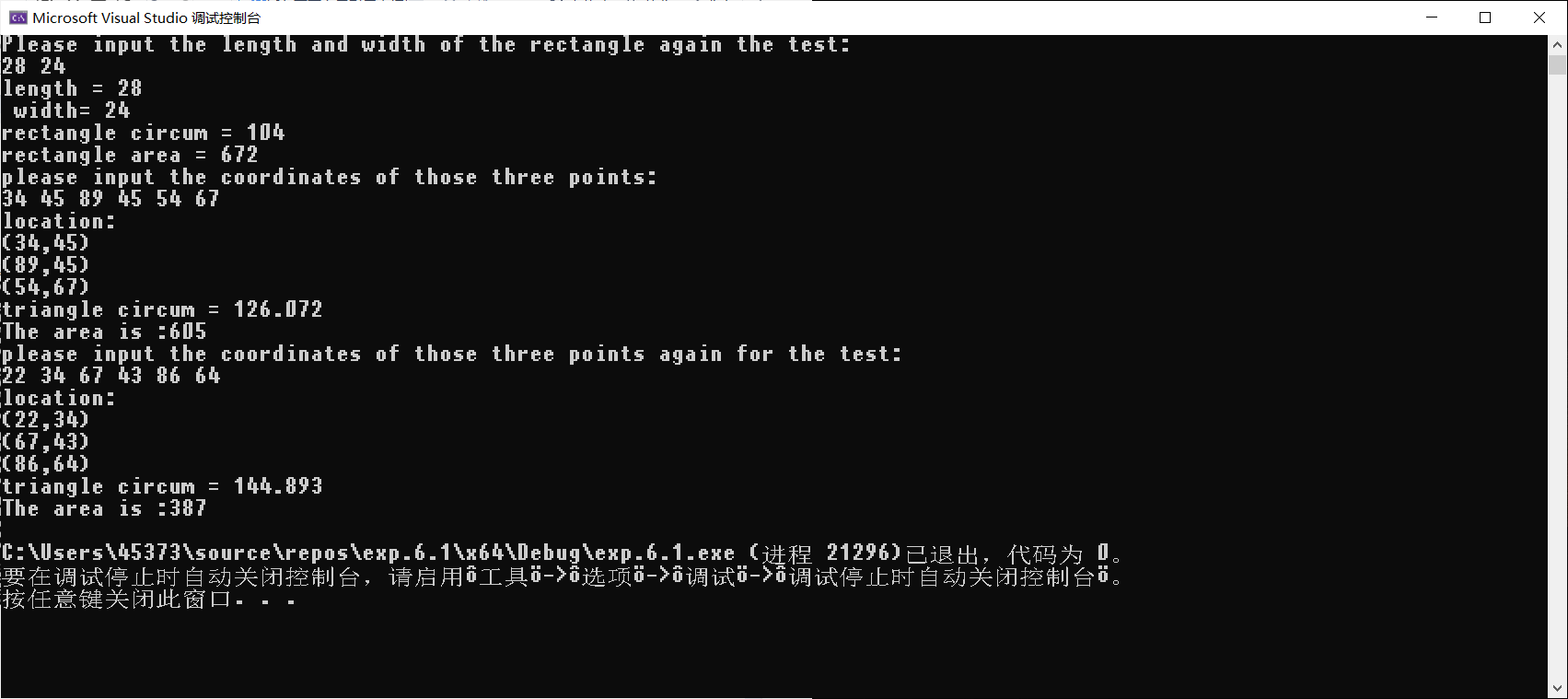
x33 = e;

y33 = f;

}

测试结果





### 题目2 教师干部类

主函数:

#include<iostream>

#include<string>

#include"Teacher.h"

using namespace std;

int main() {

cout << "请输入人员的个数" << endl;

int n;

cin >> n;

cout << endl;

//创立对象数组：

Teacher\_Cadre Inf[100];

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

cout << "请输入第" << i + 1 << "个人员的姓名、年龄、性别、地址、电话、职称、职务" << endl;

string name;

int age;

string sex;

string address;

long long tele;

string title;

string post;

cin >> name >> age >> sex >> address >> tele >> title >> post;

Inf[i].Set(name, age, sex, address, tele, title, post);

cout << endl;

}

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

Inf[i].show();

}

return 0;

}

头文件:

#pragma once

#include<string>

#include<string.h>

#include <iostream>

using namespace std;

class Teacher {

public:

void display();

protected:

string name;

int age;

string sex;

string address;

long long tele;

string title;

};

class Cadre {

public:

void display();

protected:

string name;

int age;

string sex;

string address;

long long tele;

string post;

};

class Teacher\_Cadre :public Teacher, Cadre {

/\*、在两个基类中的姓名、年龄、性别、地址、电话等数据成员使用相同名字，在派生

类中引用这些数据成员时，采用指定作用域的方式；\*/

//调用教师的display函数

public:

void show();

void Set(string name1, int age1, string sex1, string address1, long long tele1, string title1, string post1);

private:

int wage;

};

头文件.cpp

#include "Teacher.h"

#include <iostream>

using namespace std;

void Teacher::display() {

cout << "姓名：" << name << " " << "年龄：" << age << " " << "性别：" << sex;

cout << "地址：" << address << " " << "电话：" << tele << " " << "职称：" << title << " " << endl;

}

void Cadre::display() {

cout << "姓名：" << name << " " << "年龄：" << age << " " << "性别：" << sex;

cout << "地址：" << address << " " << "电话：" << tele << " " << "职务：" << post << " ";

}

void Teacher\_Cadre::show() {

Teacher::display();

cout << "职务:" << post << endl;

}

void Teacher\_Cadre::Set(string name1, int age1, string sex1, string address1, long long tele1, string title1, string post1) {

Teacher::name = name1;

Teacher::age = age1;

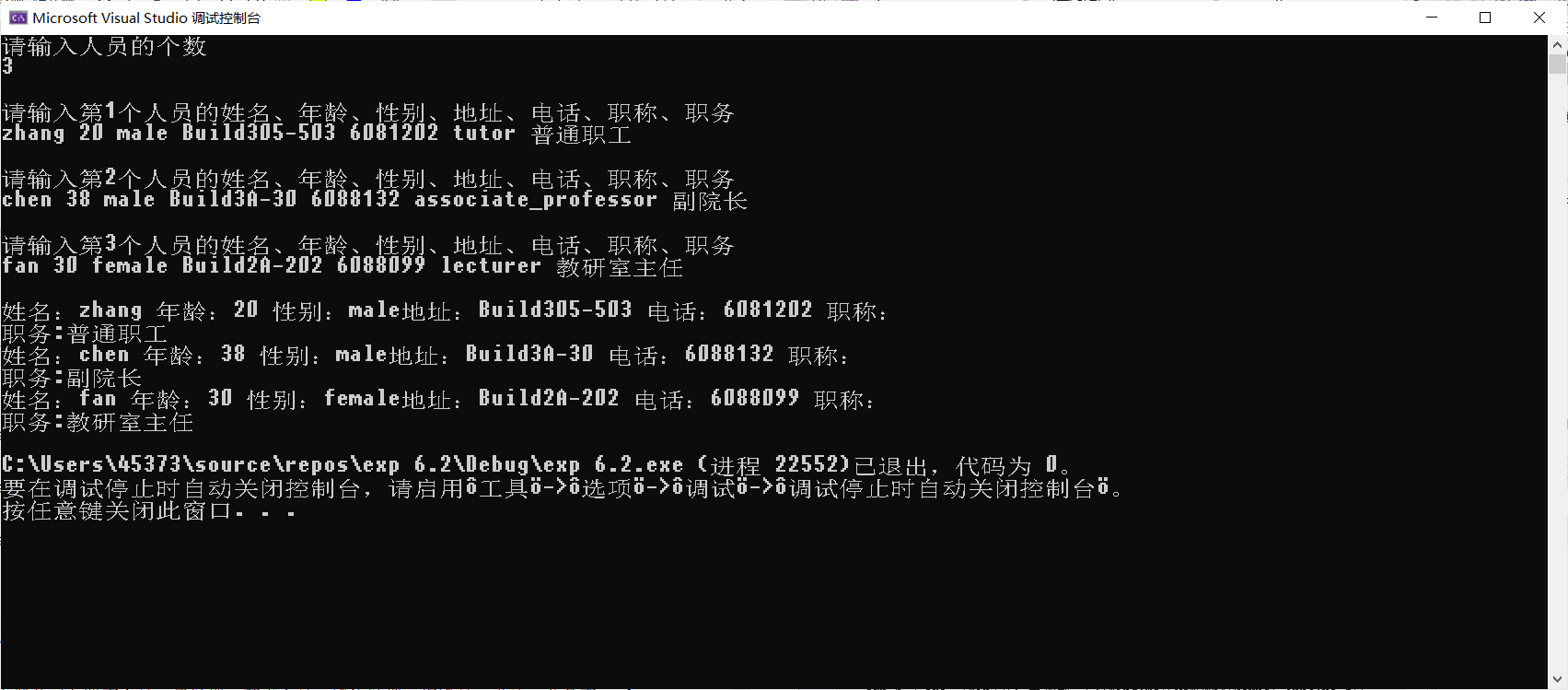
Teacher::sex = sex1;

Teacher::address = address1;

Teacher::tele = tele1;

Cadre::post = post1;

}



### 题目3 小型公司人员管理

主函数:

#include<iostream>

using namespace std;

#include"Post.h"

int main() {

cout << "请分别输入有多少个总经理，销售员，销售经理,技术人员：" << endl;

int n1,n2,n3,n4;

cin >> n1>>n2>>n3>>n4;

manager a1[100];

salesmanager a3[100];

salesman a2[100];

technician a4[100];

int departmensum1[100] = { 0 };

for (int i = 0; i < n1; i++) {//总经理

string eno1;

string name1;

string sex1;

Date birthdate1;

int year, month, day;

string title1;

cout << "请分别输入其编号，姓名，性别，出生日期，职务：" << endl;

cin >> eno1 >> name1 >> sex1 >> year >> month >> day >> title1;

birthdate1.SetYear(year);

birthdate1.SetMonth(month);

birthdate1.SetDay(day);

a1[i].set(eno1, name1, sex1, birthdate1, title1);

}

for (int i = 0; i < n2; i++) {//销售员

string eno1;

string name1;

string sex1;

Date birthdate1;

int year, month, day;

string title1;

int sales\_sum1;

string department1;

cout << "请分别输入其编号，姓名，性别，出生日期，职务,销售总额，部门：" << endl;

cin >> eno1 >> name1 >> sex1 >> year >> month >> day >> title1 >> sales\_sum1 >> department1;

birthdate1.SetYear(year);

birthdate1.SetMonth(month);

birthdate1.SetDay(day);

a2[i].set(eno1, name1, sex1, birthdate1, title1, sales\_sum1, department1);

departmensum1[mapping\_department(department1)] += sales\_sum1;

}

for (int i = 0; i < n3; i++) {//销售经理

string eno1;

string name1;

string sex1;

Date birthdate1;

int year, month, day;

string title1;

string department1;

cout << "请分别输入其编号，姓名，性别，出生日期，职务，部门：" << endl;

cin >> eno1 >> name1 >> sex1 >> year >> month >> day >> title1>>department1;

birthdate1.SetYear(year);

birthdate1.SetMonth(month);

birthdate1.SetDay(day);

a3[i].departmentsum = departmensum1[mapping\_department(department1)];

a3[i].set(eno1, name1, sex1, birthdate1, title1,department1);

}

for (int i = 0; i < n4; i++) {//技术人员

string eno1;

string name1;

string sex1;

Date birthdate1;

int year, month, day;

string title1;

int working\_time1;

cout << "请分别输入其编号，姓名，性别，出生日期，职务,工作时长：" << endl;

cin >> eno1 >> name1 >> sex1 >> year >> month >> day >> title1>>working\_time1;

birthdate1.SetYear(year);

birthdate1.SetMonth(month);

birthdate1.SetDay(day);

a4[i].set(eno1, name1, sex1, birthdate1, title1,working\_time1);

}

for (int i = 0; i < n1; i++) {

a1[i].cal\_and\_show();

}

for (int i = 0; i < n2; i++) {

a2[i].cal\_and\_show();

}

for (int i = 0; i < n3; i++) {

a3[i].cal\_and\_show();

}

for (int i = 0; i < n4; i++) {

a4[i].cal\_and\_show();

}

return 0;

}

//

/\*

1 3 1 1

10001 张可 男 75 6 10 总经理

10003 王刚 男 80 12 10 销售员 3000 华北地区

10004 陈浩月 女 82 1 10 销售员 4000 华北地区

10006 付强 男 82 4 12 销售员 5000 华东地区

10002 李兵 男 79 10 10 销售经理 华北地区

10005 宋书 男 79 1 19 技术人员 120

\*/

头文件：

#pragma once

#pragma once

#include<string>

#include<string.h>

#include <iostream>

using namespace std;

int mapping(string a);

string ave\_mapping(int a);

int mapping\_department(string a);

class Date {

public:

/\*Date(int y = 1900, int m = 1, int d = 1) {//带默认参数的构造函数

year = y;

month = m;

day = d;

}\*/

Date();

Date(int y0, int m0, int d0);

//Date();//缺失构造函数

friend class employee;

void Show();

void SetYear(int y) {

year = y;

}

void SetMonth(int m) {

month = m;

}

void SetDay(int d) {

day = d;

}

int GetYear() {

return year;

}

int GetMonth() {

return month;

}

int GetDay() {

return day;

}

private:

int year;

int month;

int day;

};

class employee {

protected:

string eno;

string name;

string sex;

Date birthdate;

int title;

long long wages;

public:

employee();

employee(string eno1, string name1);

employee(string eno1, string name1, string sex1, Date birthdate1, string title1, long long wages1);

void cal\_and\_show();

void set(string eno1, string name1, string sex1, Date birthdate1, string title1);

};

class manager :public employee {

/\* manager(string eno1, string name1, string sex1, Date birthdate1, string title1) {//赋值调用只能在函数里调用

eno = eno1;

name = name1;

sex = sex1;

birthdate.SetDay(birthdate1.GetDay());

birthdate.SetMonth(birthdate1.GetMonth());

birthdate.SetYear(birthdate1.GetYear());

title = title1;

wages = 8000;

}\*/

public:

void cal\_and\_show();

void set(string eno1, string name1, string sex1, Date birthdate1, string title1);

};

class technician : public employee {

protected:

int working\_time;

public:

/\*technician(string eno1, string name1, string sex1, Date birthdate1, string title1) {//赋值调用只能在函数里调用

eno = eno1;

name = name1;

sex = sex1;

birthdate.SetDay(birthdate1.GetDay());

birthdate.SetMonth(birthdate1.GetMonth());

birthdate.SetYear(birthdate1.GetYear());

title = title1;

wages = 8000;

}\*/

void cal\_and\_show();

void set(string eno1, string name1, string sex1, Date birthdate1, string title1, int working\_time1);

};

class salesman : public employee {

protected:

int sales\_sum;

string department;

public:

/\*salesman() {

eno = '111';

}\*/

/\*salesman(string eno1, string name1, string sex1, Date birthdate1, string title1) {//赋值调用只能在函数里调用

eno = eno1;

name = name1;

sex = sex1;

birthdate.SetDay(birthdate1.GetDay());

birthdate.SetMonth(birthdate1.GetMonth());

birthdate.SetYear(birthdate1.GetYear());

title = mapping(title1);

wages = 8000;

}\*/

void cal\_and\_show();

void set(string eno1, string name1, string sex1, Date birthdate1, string title1, int sales\_sum1, string department1);

};

class salesmanager :public salesman, manager {

public:

int departmentsum = 1;

public:

void cal\_and\_show();

void set(string eno1, string name1, string sex1, Date birthdate1, string title1, string department1);

};

头文件.cpp

#include "Post.h"

#include <iostream>

using namespace std;

int mapping(string a) {

if (a == "总经理")

return 1;

if (a == "销售员")

return 2;

if (a == "销售经理")

return 3;

if (a == "技术人员")

return 4;

}

string ave\_mapping(int a) {

if (a == 1)

return "总经理";

if (a == 3)

return "销售经理";

if (a == 2)

return "销售员";

if (a == 4)

return "技术人员";

}

int mapping\_department(string a) {

if (a == "华东地区")

return 1;

if (a == "华南地区")

return 2;

if (a == "华西地区")

return 3;

if (a == "华北地区")

return 4;

}

/\*Date(int y = 1900, int m = 1, int d = 1) {//带默认参数的构造函数

year = y;

month = m;

day = d;

}\*/

Date::Date() {

year = 1;

month = 1;

day = 1;

}

Date::Date(int y0, int m0, int d0) {//全参数

year = y0;

month = m0;

day = d0;

}

//Date();//缺失构造函数

void Date::Show() {

cout << "年，月，日分别为:" << endl;

cout << "年:" << year << endl;

cout << "月:" << month << endl;

cout << "日:" << day << endl;

}

employee::employee() {

eno = 110;

name = 'Fitz';

sex = 'male';

birthdate.SetDay(1);

birthdate.SetMonth(1);

birthdate.SetYear(1);

title = 1;

wages = 100;

}

employee::employee(string eno1, string name1) {

eno = eno1;

name = name1;

}

employee::employee(string eno1, string name1, string sex1, Date birthdate1, string title1, long long wages1) {

eno = eno1;

name = name1;

sex = sex1;

birthdate.SetDay(birthdate1.GetDay());

birthdate.SetMonth(birthdate1.GetMonth());

birthdate.SetYear(birthdate1.GetYear());

title = mapping(title1);

wages = wages1;

}

void employee::cal\_and\_show() {

cout << "编号: " << eno << "姓名: " << name << "性别: " << sex << "出生日期: " << birthdate.GetYear() << "/" << birthdate.GetMonth() << "/ " << birthdate.GetDay() << "职务: " << ave\_mapping(title) << endl;

}

void employee::set(string eno1, string name1, string sex1, Date birthdate1, string title1) {

eno = eno1;

name = name1;

sex = sex1;

birthdate.SetDay(birthdate1.GetDay());

birthdate.SetMonth(birthdate1.GetMonth());

birthdate.SetYear(birthdate1.GetYear());

title = mapping(title1);

}

void manager::cal\_and\_show() {

employee::cal\_and\_show();

cout << "月薪为:" << wages << endl;

}

void manager::set(string eno1, string name1, string sex1, Date birthdate1, string title1) {

eno = eno1;

name = name1;

sex = sex1;

birthdate.SetDay(birthdate1.GetDay());

birthdate.SetMonth(birthdate1.GetMonth());

birthdate.SetYear(birthdate1.GetYear());

title = mapping(title1);

wages = 8000;

}

void technician::cal\_and\_show() {

employee::cal\_and\_show();

cout << "工作总时长为:" << working\_time << endl;

cout << "月薪为:" << wages << endl;

}

void technician::set(string eno1, string name1, string sex1, Date birthdate1, string title1, int working\_time1) {

eno = eno1;

name = name1;

sex = sex1;

birthdate.SetDay(birthdate1.GetDay());

birthdate.SetMonth(birthdate1.GetMonth());

birthdate.SetYear(birthdate1.GetYear());

title = mapping(title1);

working\_time = working\_time1;

wages = working\_time \* 25;

}

void salesman::cal\_and\_show() {

employee::cal\_and\_show();

cout << "销售额为:" << sales\_sum << endl;

cout << "部门为:" << ave\_mapping(title) << endl;

cout << "月薪为:" << wages << endl;

}

void salesman::set(string eno1, string name1, string sex1, Date birthdate1, string title1, int sales\_sum1, string department1) {

eno = eno1;

name = name1;

sex = sex1;

birthdate.SetDay(birthdate1.GetDay());

birthdate.SetMonth(birthdate1.GetMonth());

birthdate.SetYear(birthdate1.GetYear());

title = mapping(title1);

sales\_sum = sales\_sum1;

department = department1;

wages = sales\_sum \* 0.04;

}

void salesmanager::cal\_and\_show() {

cout << "编号:" << salesman::eno << "姓名:" << salesman::name << "性别:" << salesman::sex << "出生日期:" << salesman::birthdate.GetYear() << "/" << salesman::birthdate.GetMonth() << "/" << salesman::birthdate.GetDay() << "职务:" << ave\_mapping(salesman::title) << endl;

cout << "部门为:" << ave\_mapping(salesman::title) << endl;

cout << "部门销售总额为:" << departmentsum << endl;

cout << "月薪为:" << salesman::wages << endl;

}

void salesmanager::set(string eno1, string name1, string sex1, Date birthdate1, string title1, string department1) {

salesman::eno = eno1;

salesman::name = name1;

salesman::sex = sex1;

salesman::birthdate.SetDay(birthdate1.GetDay());

salesman::birthdate.SetMonth(birthdate1.GetMonth());

salesman::birthdate.SetYear(birthdate1.GetYear());

salesman::title = mapping(title1);

salesman::department = department1;

salesman::wages = 5000 + 0.004 \* departmentsum;

}

