**《C#程序设计》课程实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号：** | **12021051064** |  | **姓名：** | **华心童** |

**实验九、类的继承与派生**

**一、实验目的**

1、类的继承与派生，理解方法与析构

3、将实验报告云盘，文件名是**学号姓名实验9**命名。

**二、实验内容**

实验准备：在F盘创建学号文件夹，用于存放实验结果文件。

运行VS2013，创建解决方案是“**Ex学号-9**”。

1. **基本实验**

**任务1、派生类，【项目名P901】**

**要求：根据以下类，完成类的设计**。

|  |
| --- |
| Employee |
| - bSalary：double //基本工资，初始值1000  - pSalary：double //实际工资  - n：int //工作年限 |
| +PN：int //成员，获取工作年限属性（get、set）  +CompSalary(); //计算普通职工工资 |

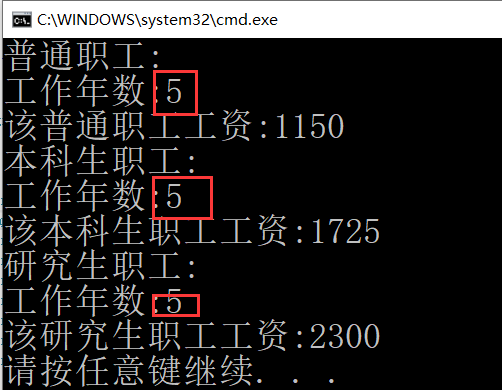
+UEmployee //Employee派生类，计算本科生职工工资

+VEmployee //Employee派生类，计算研究生职工工资

计算方法： 普通职工工资=基本工资+30\*工作年限

本科生职工工资=1.5\*普通职工工资

研究生职工工资=2\*普通职工工资



**程序源程序代码：**粘贴program.cs中代码

class Employee

{

private double bSalary = 1000;

private double pSalary;

public double PSalary

{

get { return pSalary; }

set { pSalary = value; }

}

private int n;

public int PN

{

get { return n; }

set { n = value; }

}

public void CompSalary()

{

Console.WriteLine("普通职工：");

Console.Write("工作年数：");

n = int.Parse(Console.ReadLine());

pSalary = bSalary + 30 \* n;

Console.WriteLine("该普通职工工资：" + pSalary);

}

}

class UEmployee : Employee

{

public new void CompSalary()

{

base.CompSalary();

Console.WriteLine("本科生职工：");

Console.WriteLine("工作年数："+PN);

Console.WriteLine("该本科生职工工资：" +1.5\* PSalary);

}

}

class VEmployee : UEmployee

{

public new void CompSalary()

{

base.CompSalary();

Console.WriteLine("研究生职工：");

Console.WriteLine("工作年数："+PN);

Console.WriteLine("该研究生职工工资：" + 2\*PSalary);

}

}

class Program

{

static void Main(string[] args)

{

VEmployee employee1 = new VEmployee();

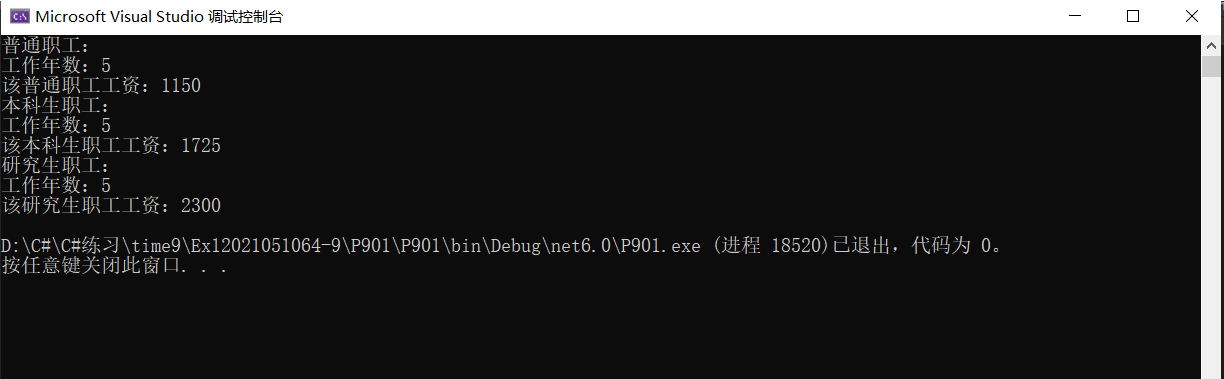
employee1.CompSalary();

Console.ReadKey();

}

}

//粘贴运行结果界面截图



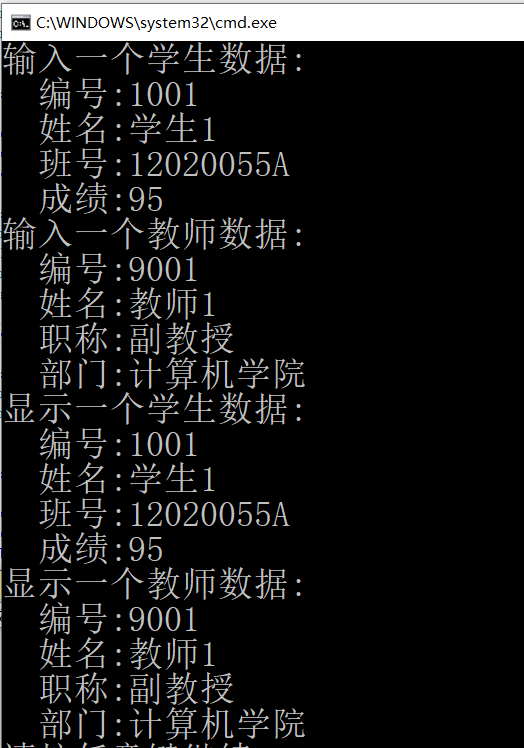
**任务2、类的继承，【项目名P902】**

**要求：**

|  |
| --- |
| Person |
| -no：int //编号  -name：string //姓名 |
| +Input() //接受输入信息  +Display(); //显示输出信息 |

+Student // 学生类，Person派生类，另有成员：-sclass（班级）、-degree（成绩）

+Teacher // 教师类，Person派生类，另有成员：-prof（职称）、-depart（部门）



**程序源程序代码：**粘贴program.cs中代码

class Person

{

private int no;

public int No

{

get { return no; }

set { no = value; }

}

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

public void Input()

{

Console.Write(" 编号：");

No = int.Parse(Console.ReadLine());

Console.Write(" 姓名：");

Name = Console.ReadLine();

}

public void Display()

{

Console.WriteLine(" 编号：" + No);

Console.WriteLine(" 姓名：" + Name);

}

}

class Student : Person

{

private string sclass;

public string Sclass

{

get { return sclass; }

set { sclass = value; }

}

private int degree;

public int Degree

{

get { return degree; }

set { degree = value; }

}

public new void Input()

{

base.Input();

Console.Write(" 班号：");

Sclass = Console.ReadLine();

Console.Write(" 成绩：");

Degree = int.Parse(Console.ReadLine());

}

public new void Display()

{

base.Display();

Console.WriteLine(" 班号："+Sclass);

Console.WriteLine(" 成绩："+Degree);

}

}

class Teacher :Person

{

private string prof;

public string Prof

{

get { return prof; }

set { prof = value; }

}

private string depart;

public string Depart

{

get { return depart; }

set { depart = value; }

}

public new void Input()

{

base.Input();

Console.Write(" 职称：");

Prof = Console.ReadLine();

Console.Write(" 部门：");

Depart = Console.ReadLine();

}

public new void Display()

{

base.Display();

Console.WriteLine(" 职称：" + Prof);

Console.WriteLine(" 部门：" + Depart);

}

}

class Program

{

static void Main(string[] args)

{

Student s1 = new Student();

Teacher T1 = new Teacher();

Console.WriteLine("输入一个学生的数据：");

s1.Input();

Console.WriteLine("输入一个教师的数据：");

T1.Input();

Console.WriteLine("显示一个学生的数据：");

s1.Display();

Console.WriteLine("显示一个教师的数据：");

T1.Display();

Console.ReadKey();

}

//粘贴运行结果界面截图



**任务3、构造与析构，【项目名P903】**

**要求：**设计一个学生类，包括学号、姓名、性别属性。设计一个大学生类，继承于学生类，其属性除具备学生类的属性外，还有专业、高考成绩。设计一个研究生类，继承于大学生类，其属性除具备大学生类的属性外，还有研究方向，导师姓名属性。对类进行测试（生成对象、输出对象属性）。对各个类进行析构，并输出相关提示信息。

**源程序代码：**粘贴program.cs中代码

class Student

{

private string id;

public string Id

{

get { return id; }

set { id = value; }

}

private string name;

public string Name

{

get { return name; }

set { name = value; }

}

public string sex;

public Student()

{

Console.WriteLine("调用Stduent的构造函数");

Console.Write(" 学号：");

Id = Console.ReadLine();

Console.Write(" 姓名：" );

Name = Console.ReadLine();

Console.Write(" 性别：");

sex = Console.ReadLine();

}

~Student()

{

Console.WriteLine("调用Stduent的析构函数");

}

public void print()

{

Console.WriteLine("——此处是Student类——");

Console.WriteLine(" 学号：" + Id);

Console.WriteLine(" 姓名：" + Name);

Console.WriteLine(" 性别：" + sex);

}

}

class Graduate:Student

{

private int pScore;

public int PScore

{

get { return pScore; }

set { pScore = value; }

}

private int eScore;

public int EScore

{

get { return eScore; }

set { eScore = value; }

}

public Graduate()

{

Console.WriteLine("调用Graduate的构造函数");

Console.Write(" 专业成绩：");

PScore=int.Parse(Console.ReadLine());

Console.Write(" 高考成绩：");

EScore=int.Parse(Console.ReadLine());

}

~Graduate()

{

Console.WriteLine("调用Graduate的析构函数");

}

public new void print()

{

base.print();

Console.WriteLine("——此处是Graduate类——");

Console.WriteLine(" 专业成绩：" + PScore);

Console.WriteLine(" 高考成绩：" + EScore);

}

}

class Postgraduate : Graduate

{

private string research;

public string Research

{

get { return research; }

set { research = value;}

}

private string instructorname;

public string Instructorname

{

get { return instructorname; }

set { instructorname = value; }

}

public Postgraduate()

{

Console.WriteLine("调用Postgraduate的构造函数");

Console.Write(" 研究方向：");

Research = Console.ReadLine();

Console.Write(" 导师姓名：");

Instructorname = Console.ReadLine();

}

~Postgraduate()

{

Console.WriteLine("调用Postgraduate的析构函数");

}

public new void print()

{

base.print();

Console.WriteLine("——此处是 Postgraduate类——");

Console.WriteLine(" 研究方向：" + research);

Console.WriteLine(" 导师姓名：" + instructorname);

}

}

class Program

{

static void Main(string[] args)

{

Postgraduate P1=new Postgraduate();

//P1.Id = "12021051001";

//P1.Name = "华心童";

//P1.sex = "男";

//P1.PScore = 90;

//P1.EScore = 99;

//P1.Research = "Computer";

//P1.Instructorname = "张三";

P1.print();

Console.ReadKey();

}

}

//粘贴运行结果界面截图

