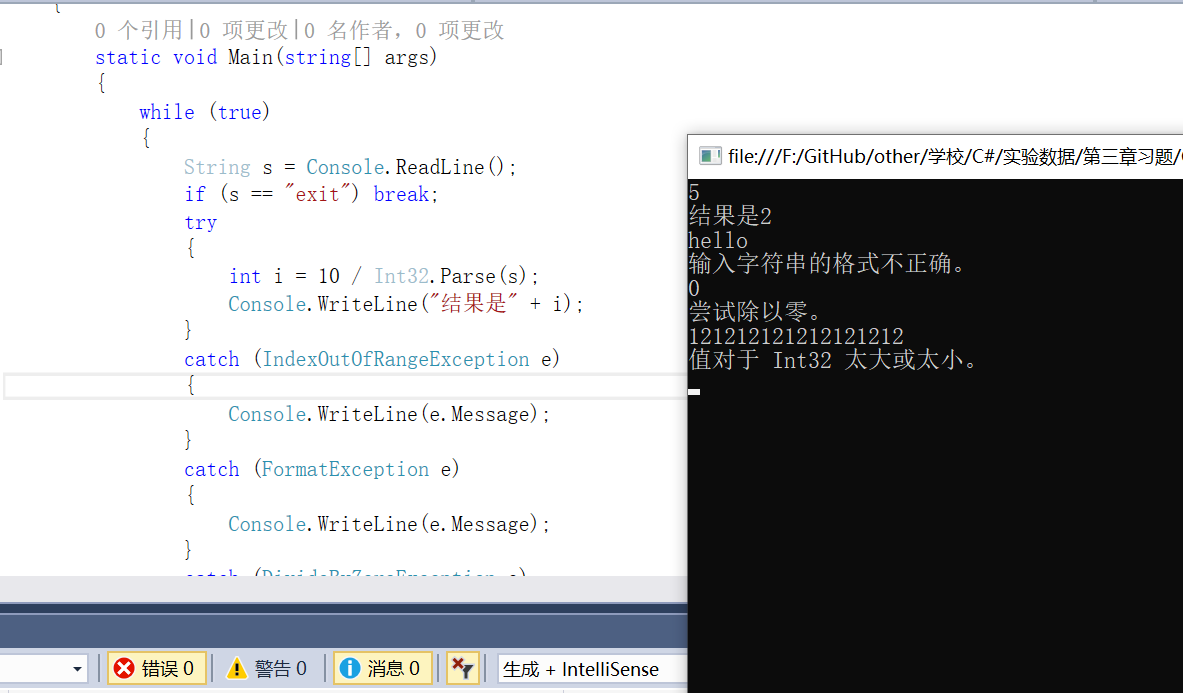
**四、程序分析**

**1.下面程序，输入不同参数，观察并分析运行结果。**



using System;

namespace Test

{

class Program

{

static void Main(string[] args)

{

while (true)

{

String s = Console.ReadLine();

if (s == "exit") break;

try

{

int i = 10 / Int32.Parse(s);

Console.WriteLine("结果是" + i);

}

catch (IndexOutOfRangeException e)

{

Console.WriteLine(e.Message);

}

catch (FormatException e)

{

Console.WriteLine(e.Message);

}

catch (DivideByZeroException e)

{

Console.WriteLine(e.Message);

}

catch (OverflowException e)

{

Console.WriteLine(e.Message);

}

}

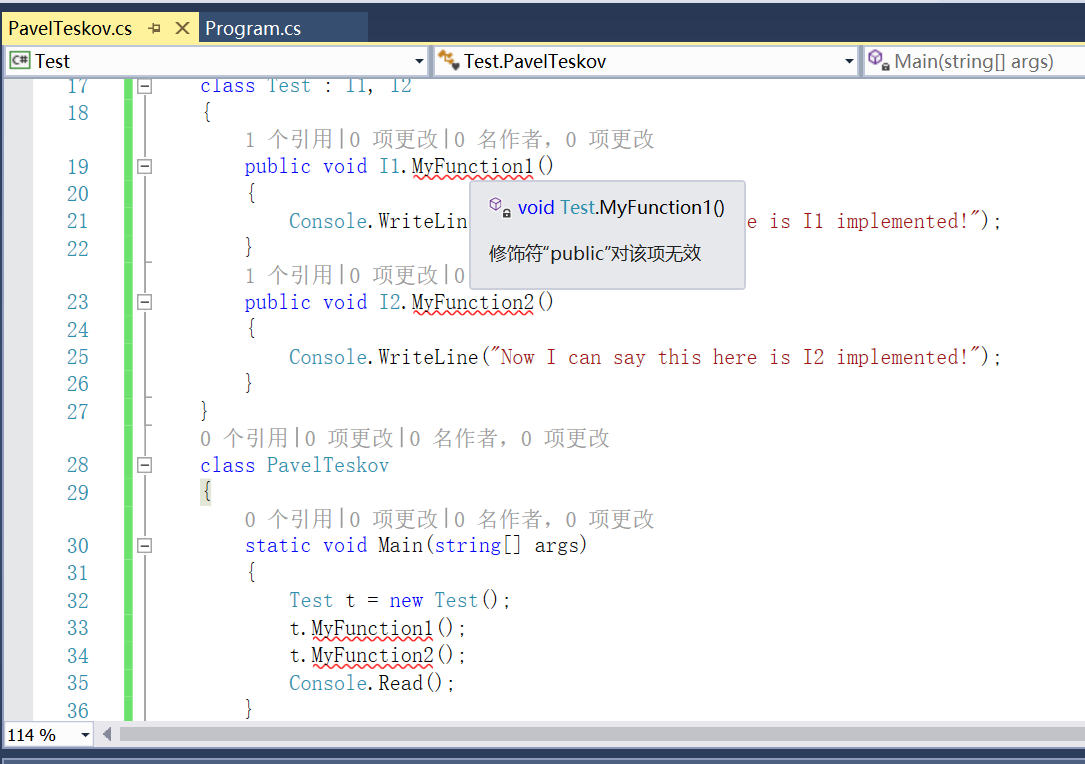
Console.Read();

}

}

}

**2.**





**代码改为：**

using System;

namespace Test

{

interface I1

{

void MyFunction1();

}

interface I2

{

void MyFunction2();

}

class Test : I1, I2

{

public void MyFunction1()

{

Console.WriteLine("Now I can say this here is I1 implemented!");

}

public void MyFunction2()

{

Console.WriteLine("Now I can say this here is I2 implemented!");

}

}

class PavelTeskov

{

static void Main(string[] args)

{

Test t = new Test();

t.MyFunction1();

t.MyFunction2();

Console.Read();

}

}

}

**3.** **用户自定义从 Digit 到 byte 的隐式转换**

using System;

namespace Test

{

public class Digit

{

byte value;

public Digit(byte value)

{

if (value < 0 || value > 9)

throw new ArgumentException();

this.value = value;

}

public static implicit operator byte(Digit d)

{

return d.value;

}

}

class dig

{

static void Main(string[] args)

{

Digit dig = new Digit(7);

byte num = dig;

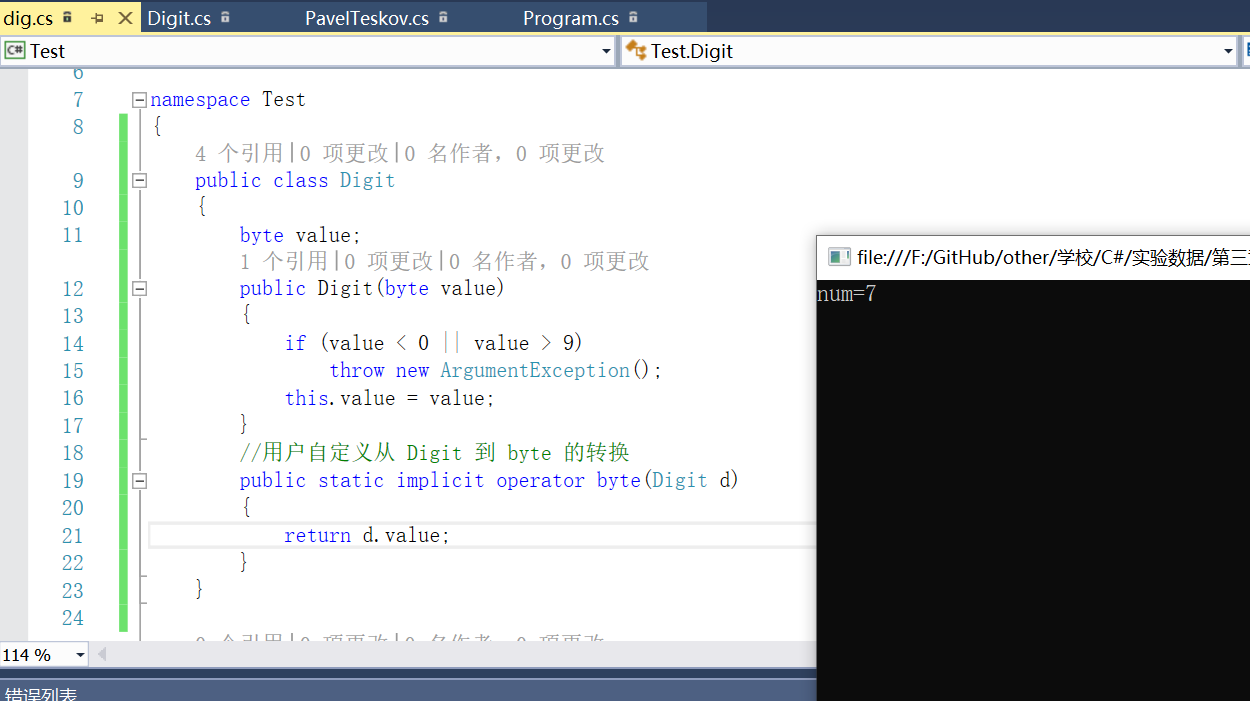
Console.WriteLine("num={0}", num);

Console.ReadLine();

}

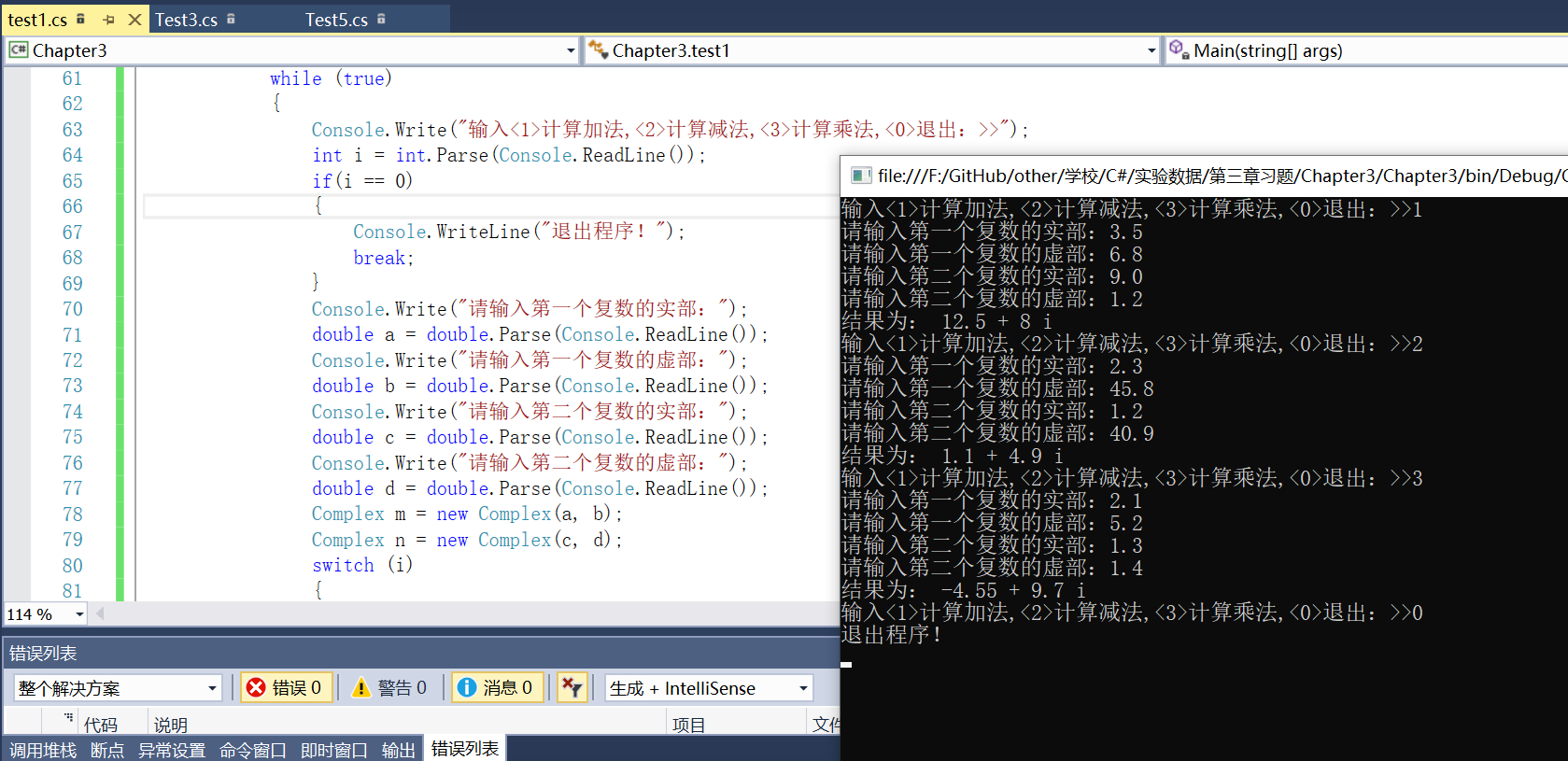
}

}



**五、编程题**

**1.定义描述复数的类,并实现复数的输入和输出。设计三个方法分别完成复数的加、减和乘法运算。**



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Chapter3

{

public class Complex

{

private double real;

private double image;

public Complex(double real, double image)

{

this.real = real;

this.image = image;

}

public double Real

{

get

{

return real;

}

set

{

real = value;

}

}

public double Image

{

get

{

return image;

}

set

{

image = value;

}

}

public static Complex add(Complex c1, Complex c2)

{

return new Complex(c1.real + c2.real, c1.image + c2.image);

}

public static Complex sub(Complex c1, Complex c2)

{

return new Complex(c1.real - c2.real, c1.image - c2.image);

}

public static Complex mul(Complex c1, Complex c2)

{

return new Complex(c1.real \* c2.real - c1.image \* c2.image, c1.image \* c2.real + c1.real \* c2.image);

}

public static void print(Complex c)

{

Console.WriteLine("结果为： "+c.real + " + " + c.image + " i");

}

}

class test1

{

static void Main(string[] args)

{

while (true)

{

Console.Write("输入<1>计算加法,<2>计算减法,<3>计算乘法,<0>退出：>>");

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

if(i == 0)

{

Console.WriteLine("退出程序！");

break;

}

Console.Write("请输入第一个复数的实部：");

double a = double.Parse(Console.ReadLine());

Console.Write("请输入第一个复数的虚部：");

double b = double.Parse(Console.ReadLine());

Console.Write("请输入第二个复数的实部：");

double c = double.Parse(Console.ReadLine());

Console.Write("请输入第二个复数的虚部：");

double d = double.Parse(Console.ReadLine());

Complex m = new Complex(a, b);

Complex n = new Complex(c, d);

switch (i)

{

case 1: Complex.print(Complex.add(m, n)); break;

case 2: Complex.print(Complex.sub(m, n)); break;

case 3: Complex.print(Complex.mul(m, n)); break;

}

}

Console.Read();

}

}

}

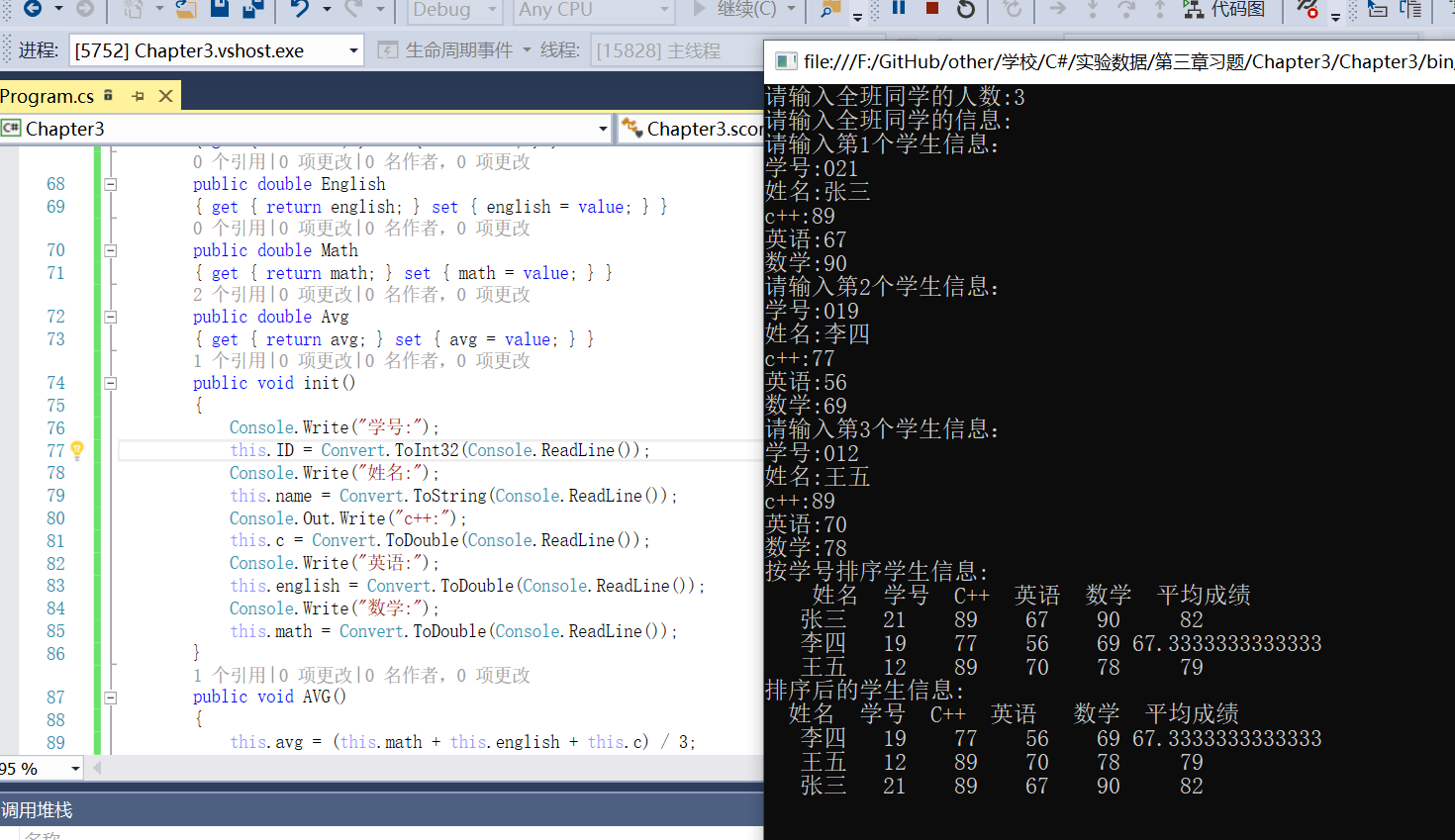
**2.定义全班学生成绩类,包括姓名、学号、C++成绩、英语成绩、数学成绩和平均成绩。设计下列4个方法**

**(1)全班成绩的输入;**

**(2)求出每一个同学的平均成绩;**

**(3)按平均成绩的升序排序**

**(4)输出全班成绩。**



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Chapter3

{

class Program

{

static void Main(string[] args)

{

int t = 0; int k = 1;

Console.Write("请输入全班同学的人数:");

t = Convert.ToInt32(Console.ReadLine());

score[] stu = new score[t];

Console.WriteLine("请输入全班同学的信息:");

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

{

Console.WriteLine("请输入第{0}个学生信息：", k++);

stu[i] = new score();

stu[i].init();

stu[i].AVG();

}

Console.WriteLine("按学号排序学生信息:");

Console.WriteLine(" 姓名 学号 C++ 英语 数学 平均成绩");

for (int j = 0; j < t; j++)

{

stu[j].display();

}

Console.WriteLine("排序后的学生信息:");

paixu(t, stu);

Console.Read();

}

public static void paixu(int t, score[] stu)

{

score stud = new score();

if (stu.Length <= 0)

return;

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

{

for (int j = 0; j < t - i - 1; j++)

{

if (stu[j].Avg > stu[j + 1].Avg)

{ stud = stu[j + 1]; stu[j + 1] = stu[j]; stu[j] = stud; }

}

}

Console.WriteLine(" 姓名 学号 C++ 英语 数学 平均成绩");

foreach (score x in stu)

x.display();

}

}

class score

{

private string name;

private int ID;

private double c;

private double english;

private double math;

double avg;

public score()

{ this.avg = 0; }

public string Name

{ get { return name; } set { name = value; } }

public int id

{ get { return ID; } set { ID = value; } }

public double C

{ get { return c; } set { c = value; } }

public double English

{ get { return english; } set { english = value; } }

public double Math

{ get { return math; } set { math = value; } }

public double Avg

{ get { return avg; } set { avg = value; } }

public void init()

{

Console.Write("学号:");

this.ID = Convert.ToInt32(Console.ReadLine());

Console.Write("姓名:");

this.name = Convert.ToString(Console.ReadLine());

Console.Out.Write("c++:");

this.c = Convert.ToDouble(Console.ReadLine());

Console.Write("英语:");

this.english = Convert.ToDouble(Console.ReadLine());

Console.Write("数学:");

this.math = Convert.ToDouble(Console.ReadLine());

}

public void AVG()

{

this.avg = (this.math + this.english + this.c) / 3;

}

public void display()

{

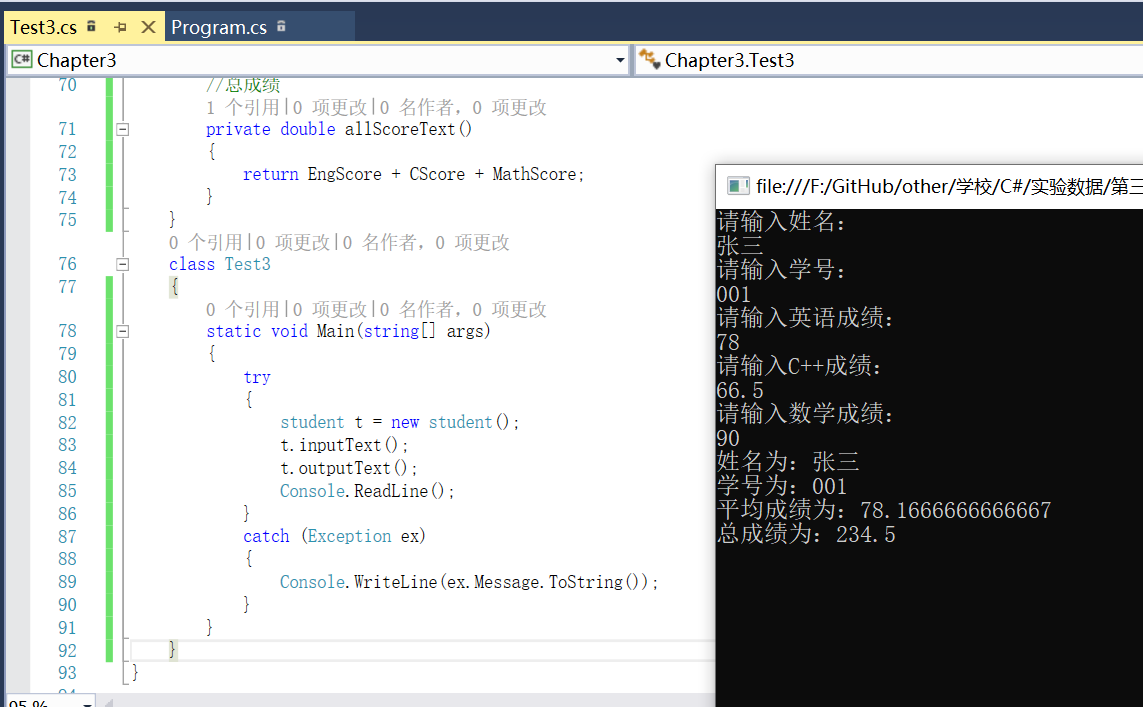
Console.WriteLine("{0,5}{1,5}{2,6}{3,6}{4,6} {5,6}", name, ID, c, english, math, avg);

}

}

}

**3.定义一个描述学生基本情况的类,数据成员包括姓名、学号以及C++、英语和数学成绩,成员函数包括输出数据、设置姓名和学号、设置三门课的成绩,求出总成绩和平均成绩。**



using System;

namespace Chapter3

{

class student

{

//字段定义

private string name; //姓名

private string no; //学号

private double engScore; //英语成绩

private double cScore; //C++成绩

private double mathScore; //数学成绩

//定义属性,通过属性访问、修改相应字段

public string Name

{

set { name = value; }

get { return name; }

}

public string No

{

set { no = value; }

get { return no; }

}

public double EngScore

{

set { engScore = value; }

get { return engScore; }

}

public double CScore

{

set { cScore = value; }

get { return cScore; }

}

public double MathScore

{

set { mathScore = value; }

get { return mathScore; }

}

//输入相应的数据

public void inputText()

{

Console.WriteLine("请输入姓名：");

Name = Console.ReadLine();

Console.WriteLine("请输入学号：");

No = Console.ReadLine();

Console.WriteLine("请输入英语成绩：");

EngScore = double.Parse(Console.ReadLine());

Console.WriteLine("请输入C++成绩：");

CScore = double.Parse(Console.ReadLine());

Console.WriteLine("请输入数学成绩：");

MathScore = double.Parse(Console.ReadLine());

}

public void outputText()

{

Console.WriteLine("姓名为：{0}", Name);

Console.WriteLine("学号为：{0}", No);

Console.WriteLine("平均成绩为：{0}", avgScoreText());

Console.WriteLine("总成绩为：{0}", allScoreText());

}

private double avgScoreText()

{

return (EngScore + CScore + MathScore) / 3;

}

private double allScoreText()

{

return EngScore + CScore + MathScore;

}

}

class Test3

{

static void Main(string[] args)

{

try

{

student t = new student();

t.inputText();

t.outputText();

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message.ToString());

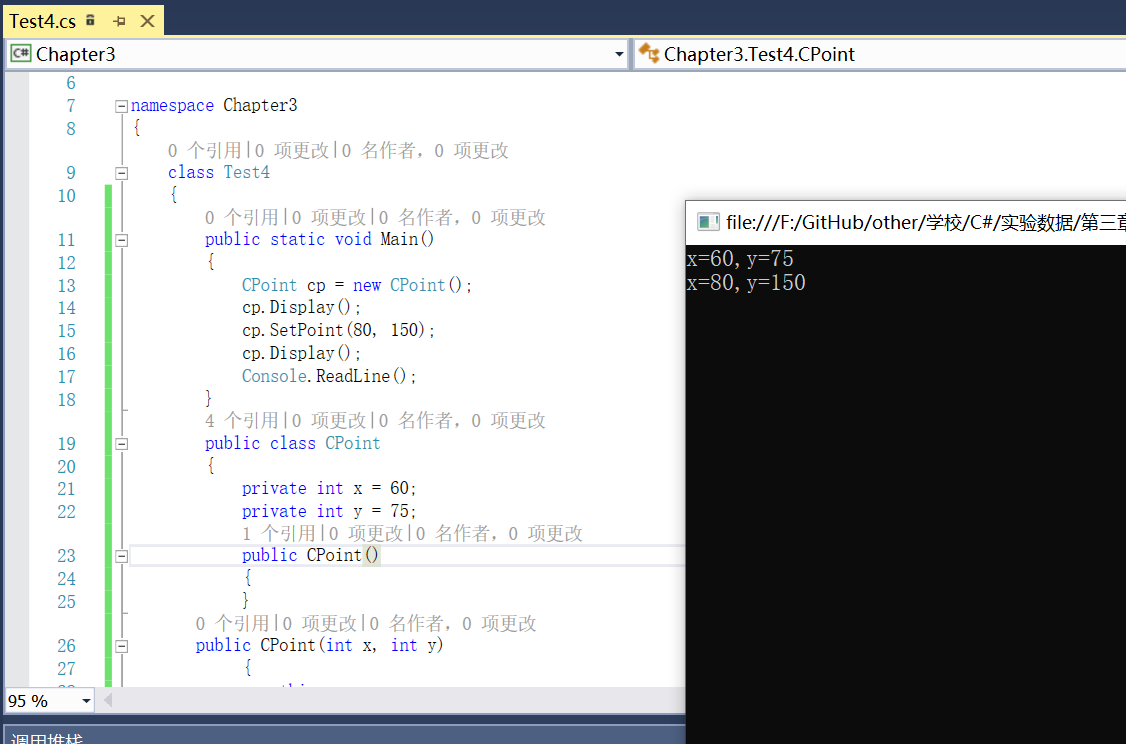
}

}

}

}

**4.设有一个描述坐标点的CPoint类,其私有变量x和y代表一个点的x、y坐标值。编写程序实现以下功能:利用构造函数传递参数,并设其默认参数值为60和75,利用成员函数display()输出这一默认值;利用公有成员函数setpoint0将坐标值修改为(80,150),并利用成员函数输出修改后的坐标值。**



using System;

namespace Chapter3

{

class Test4

{

public static void Main()

{

CPoint cp = new CPoint();

cp.Display();

cp.SetPoint(80, 150);

cp.Display();

Console.ReadLine();

}

public class CPoint

{

private int x = 60;

private int y = 75;

public CPoint()

{

}

public CPoint(int x, int y)

{

this.x = x;

this.y = y;

}

public void Display()

{

Console.WriteLine("x={0},y={1}", x, y);

}

public void SetPoint(int x, int y)

{

this.x = x;

this.y = y;

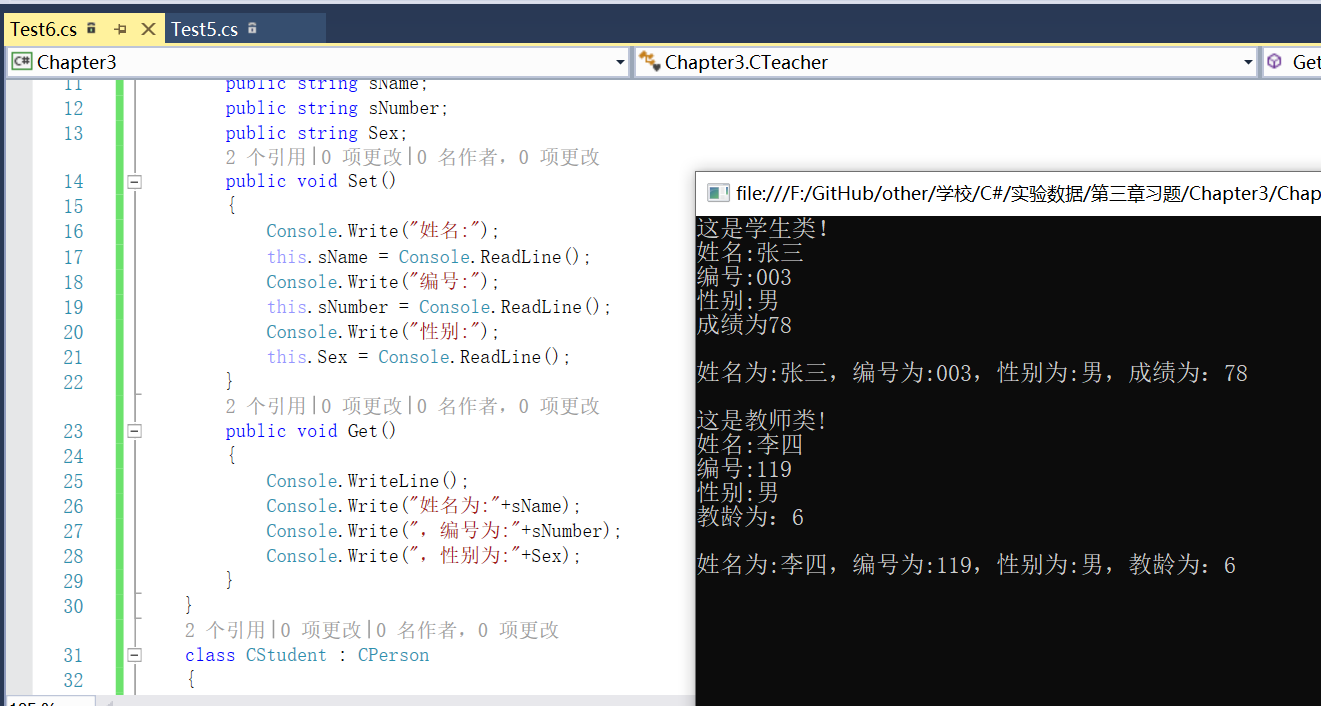
}

}

}

}

**5.定义一个人员类 CPerson,包括数据成员:姓名、编号、性别和用于输入输出的成员函数。在此基础上派生出学生类 CStudent(增加成绩)和教师类Cteacher (增加教龄),并实现对学生和教师信息的输入输出。**



using System;

namespace Chapter3

{

class CPerson

{

public string sName;

public string sNumber;

public string Sex;

public void Set()

{

Console.Write("姓名:");

this.sName = Console.ReadLine();

Console.Write("编号:");

this.sNumber = Console.ReadLine();

Console.Write("性别:");

this.Sex = Console.ReadLine();

}

public void Get()

{

Console.WriteLine();

Console.Write("姓名为:"+sName);

Console.Write("，编号为:"+sNumber);

Console.Write("，性别为:"+Sex);

}

}

class CStudent : CPerson

{

public float Score;

public new void Set()

{

Console.WriteLine("这是学生类！");

base.Set();

Console.Write("成绩为");

this.Score = Convert.ToInt32(Console.ReadLine());

}

public new void Get()

{

base.Get();

Console.WriteLine("，成绩为：" + Score);

}

}

class CTeacher : CPerson

{

public int age;

public new void Set()

{

Console.WriteLine("这是教师类!");

base.Set();

Console.Write("教龄为：");

this.age = Convert.ToInt32(Console.ReadLine());

}

public new void Get()

{

base.Get();

Console.WriteLine("，教龄为：" + age);

}

}

class Test6

{

static void Main(string[] args)

{

CStudent stu = new CStudent();

CTeacher tea = new CTeacher();

stu.Set();

stu.Get();

Console.WriteLine();

tea.Set();

tea.Get();

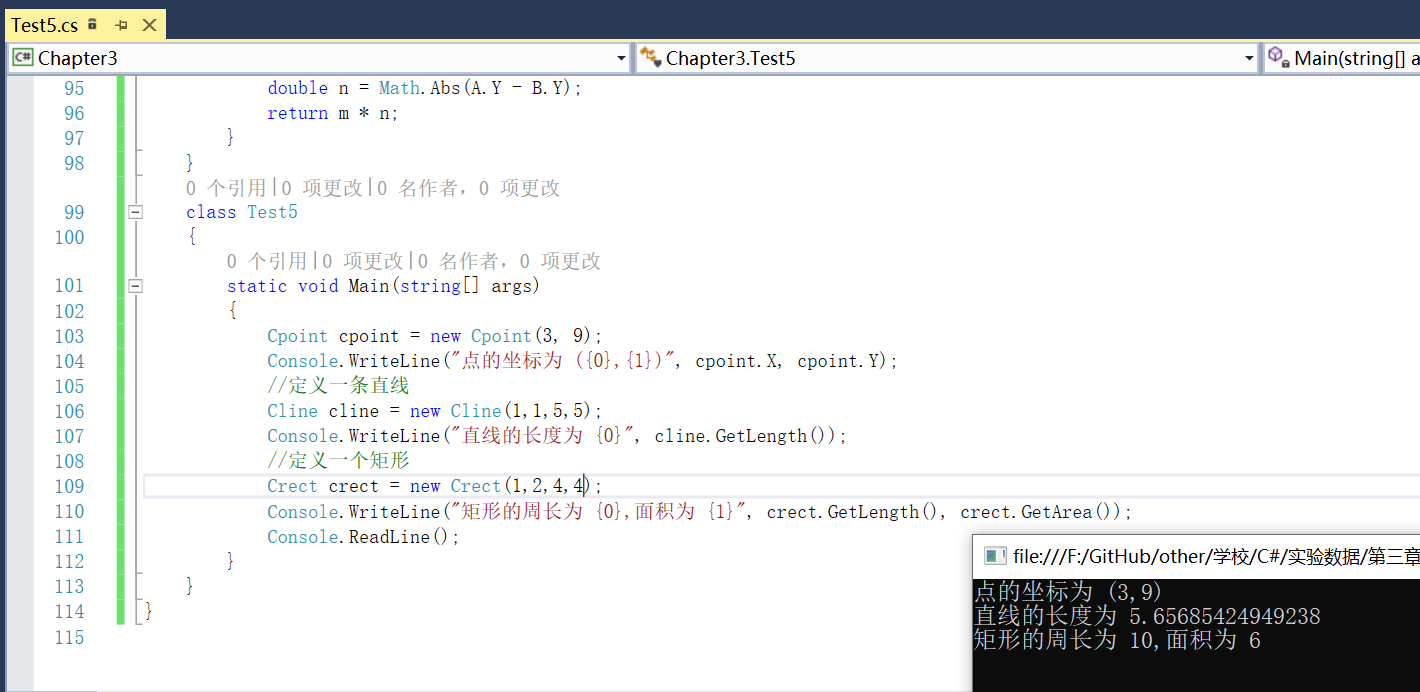
Console.ReadLine();

}

}

}

**6.把定义平面直角坐标系上的一个点的类 CPoint作为基类,派生出描述一条直线的类CLine,再派生出一个矩形类 CRect。要求成员函数能求出两点间的距离、矩形的周长和面积等。设计一个测试程序,并构造完整的程序。**



using System;

namespace Chapter3

{

class Cpoint

{

private double x;//坐标x轴的值

public double X

{

get { return x; }

set { x = value; }

}

private double y;//坐标y轴的值

public double Y

{

get { return y; }

set { y = value; }

}

public Cpoint()

{

this.X = 0;

this.Y = 0;

}

public Cpoint(double x, double y)

{

this.X = x;

this.Y = y;

}

}

class Cline : Cpoint

{

private Cpoint a = new Cpoint();//直线的一个端点

public Cpoint A

{

get { return a; }

set { a = value; }

}

private Cpoint b = new Cpoint();//直线的另一个端点

public Cpoint B

{

get { return b; }

set { b = value; }

}

public Cline(double a1, double a2, double b1, double b2)

{

A.X = a1;

A.Y = a2;

B.X = b1;

B.Y = b2;

}

//求直线长度

public double GetLength()

{

double m = A.X - B.X;

double n = A.Y - B.Y;

return Math.Sqrt(Math.Pow(m, 2) + Math.Pow(n, 2));

}

}

class Crect : Cpoint

{

private Cpoint a = new Cpoint();//矩形的一个端点

public Cpoint A

{

get { return a; }

set { a = value; }

}

private Cpoint b = new Cpoint();//与a为对角的点

public Cpoint B

{

get { return b; }

set { b = value; }

}

public Crect(double a1, double a2, double b1, double b2)

{

A.X = a1;

A.Y = a2;

B.X = b1;

B.Y = b2;

}

//求周长

public double GetLength()

{

double m = Math.Abs(A.X - B.X);

double n = Math.Abs(A.Y - B.Y);

return 2.0 \* (m + n);

}

//求面积

public double GetArea()

{

double m = Math.Abs(A.X - B.X);

double n = Math.Abs(A.Y - B.Y);

return m \* n;

}

}

class Test5

{

static void Main(string[] args)

{

Cpoint cpoint = new Cpoint(3, 9);

Console.WriteLine("点的坐标为 ({0},{1})", cpoint.X, cpoint.Y);

Cline cline = new Cline(1,1,5,5);

Console.WriteLine("直线的长度为 {0}", cline.GetLength());

Crect crect = new Crect(1,2,4,4);

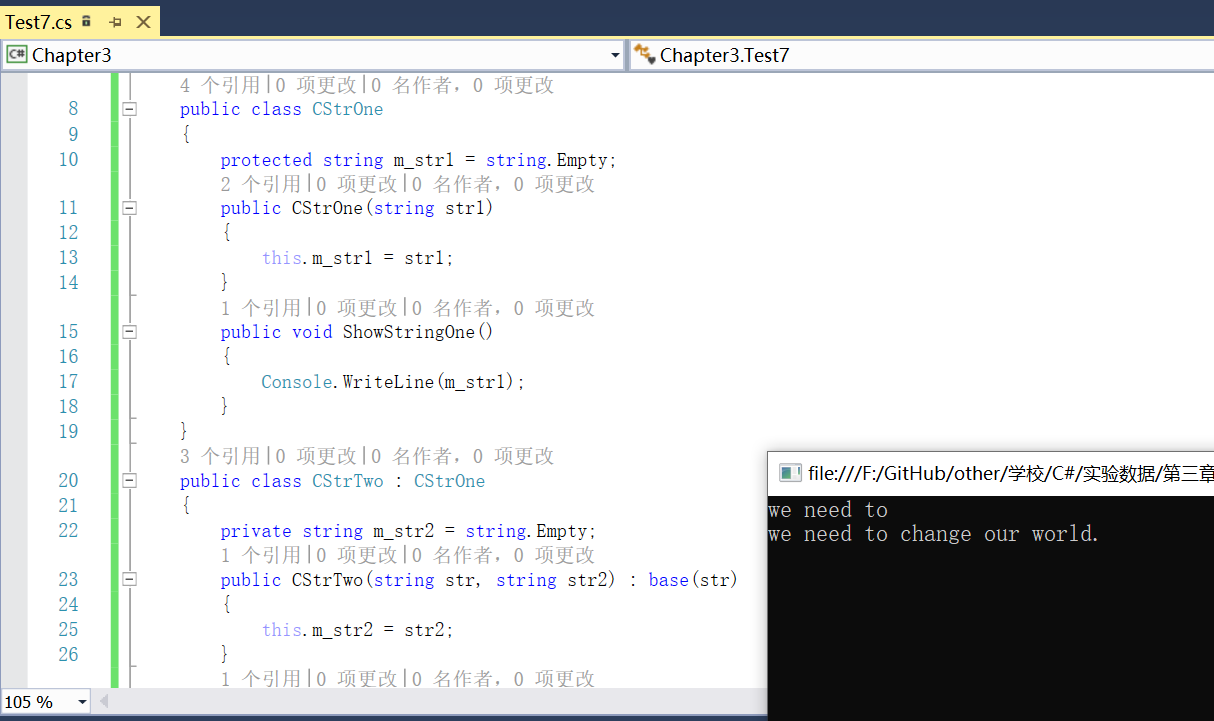
Console.WriteLine("矩形的周长为 {0},面积为 {1}", crect.GetLength(), crect.GetArea());

Console.ReadLine(); }

}

}

**7. 定义一个字符串类 CStrOne,它包含一个存放字符串的数据成员,能够通过构造函数初始化字符串,通过成员函数显示字符串的内容。在此基础上派生出 CStrTwo类,增加一个存放字符串的数据成员,并能通过派生类的构造函数传递参数,初始化两个字符串,通过成员函数进行两个字符串的合并及输出。**



using System;

namespace Chapter3

{

public class CStrOne

{

protected string m\_str1 = string.Empty;

public CStrOne(string str1)

{

this.m\_str1 = str1;

}

public void ShowStringOne()

{

Console.WriteLine(m\_str1);

}

}

public class CStrTwo : CStrOne

{

private string m\_str2 = string.Empty;

public CStrTwo(string str, string str2) : base(str)

{

this.m\_str2 = str2;

}

public void ShowStringTwo()

{

Console.WriteLine("{0}{1}", m\_str1, m\_str2);

}

}

class Test7

{

static void Main(string[] args)

{

CStrOne cs1 = new CStrOne("we need to ");

cs1.ShowStringOne();

CStrTwo cs2 = new CStrTwo("we need to ", "change our world."); cs2.ShowStringTwo();

Console.Read();

}

}

}