**《C#程序设计》课程实验报告**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号：** | **120210510644** |  | **姓名：** | **华心童** |

**实验三、C#的数组及语法**

**一、实验目的**

1、掌握C#的引用数据类型。

2、进一步巩固控制台应用程序的输入和输出操作。

3、将实验报告云盘，文件名是**学号姓名实验3**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-3**”的项目名“P301”。

1. **基本实验**

**任务1、了解枚举类型的值**【项目名P301】

定义一个枚举weekday{Sun，Mon=5，Tue，Wed，Thu=10，Fri，Sat=15}

分别输出每个星期的值。

**源程序代码：**粘贴program.cs中代码

class Program1

{

enum weekday

{

Sun,Mon=5,Tue,Wed,Thu=10,Fri,Sat=15

}

static void Main()

{

int day1= (int)weekday.Sun;

int day2 = (int)weekday.Mon;

int day3 = (int)weekday.Tue;

int day4 = (int)weekday.Wed;

int day5 = (int)weekday.Thu;

int day6 = (int)weekday.Fri;

int day7 = (int)weekday.Sat;

Console.WriteLine("Sun={0}",day1);

Console.WriteLine("Mon={0}",day2);

Console.WriteLine("Tue={0}",day3);

Console.WriteLine("Wed={0}",day4);

Console.WriteLine("Thu={0}",day5);

Console.WriteLine("Fri={0}",day6);

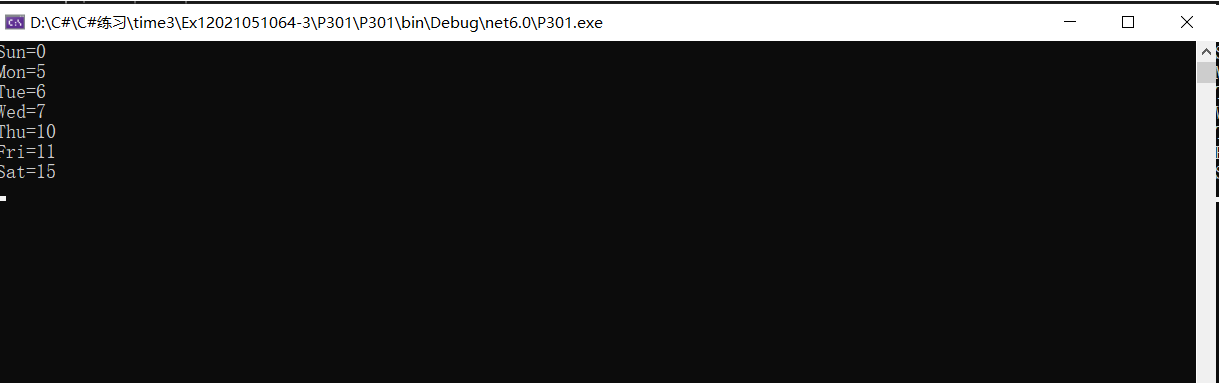
Console.WriteLine("Sat={0}",day7);

Console.ReadKey();

}

}

//粘贴运行结果界面截图



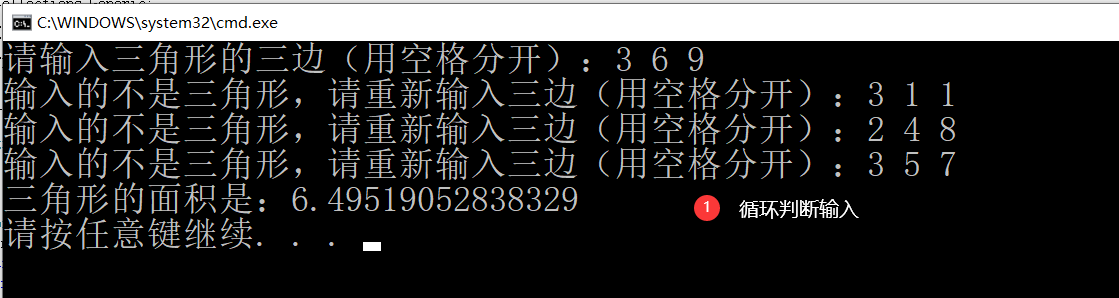
**任务2、利用数组判别三角形【项目名P302】**

要求一行输入用空格隔开的三个值，分别表示三角形的三条边，若三边不能组成三角形，则要求再次输入，直到输入的三条边可以组成三角形为止，最后，计算三角形的面积，并输出三边及面积值（保留两位小数）。

提示： 1）利用循环结构，将空格区分的数的各个数值赋值给数组；



**结果样张**



**源程序代码：**粘贴program.cs中代码

class Program1

{

static void Main()

{

double p, s=0;

Console.Write("请输入三角形的三边（用空格分开）：");

string value = Console.ReadLine();

string[] vals = value.Split(' ');

int[] num1= new int[3];

num1[0] = int.Parse(vals[0]);

num1[1] = int.Parse(vals[1]);

num1[2] = int.Parse(vals[2]);

p = (num1[0] + num1[1] + num1[2]) / 2.0;

s = Math.Sqrt(p \* (p - num1[0]) \* (p - num1[2]) \* (p - num1[1]));

if(num1[0]+num1[1]>num1[2]&&num1[0]+num1[2]>num1[1] && num1[2]+num1[1]>num1[0])

{

p = (num1[0] + num1[1] + num1[2]) / 2.0;

s = Math.Sqrt(p \* (p - num1[0]) \* (p - num1[2]) \* (p - num1[1]));

Console.WriteLine("三角形的面积是：{0:F2}",s);

Console.Write("请按任意键继续...");

Console.ReadKey();

}

else{

for (int i = 0; num1[0] + num1[1]<=num1[2] ||num1[0] + num1[2] <= num1[1] ||num1[2] + num1[1] <=num1[0]; i++)

{

Console.Write("输入的不是三角形，请重新输入三边（用空格分开）：");

value = Console.ReadLine();

vals = value.Split(' ');

num1[0] = int.Parse(vals[0]);

num1[1] = int.Parse(vals[1]);

num1[2] = int.Parse(vals[2]);

}

p = (num1[0] + num1[1] + num1[2]) / 2.0;

s = Math.Sqrt(p \* (p - num1[0]) \* (p - num1[2]) \* (p - num1[1]));

Console.WriteLine("三角形的面积是：{0:F2}", s);

Console.Write("请按任意键继续...");

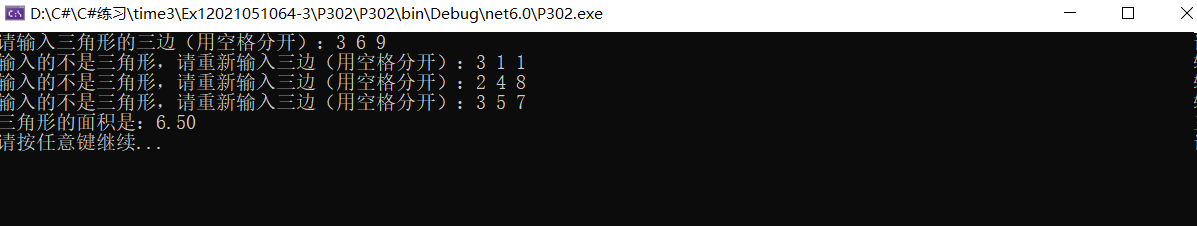
Console.ReadKey();

}

}

}

//粘贴运行结果界面截图



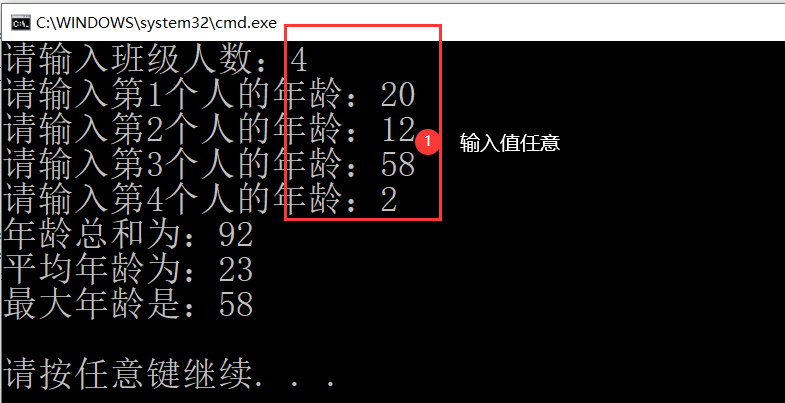
**任务3、利用数组**

从控制台输入班级人数，将每个人的年龄放入数组，并求出所有人的年龄总和、平均年龄及最大年龄。

提示： 1）利用循环结构将输入的各个数值赋值给数组；

2）定义一个最大年龄为0，利用循环结构比较数组的最大年龄；

**结果样张**



**源程序代码：**粘贴program.cs中代码

class Program1

{

static void Main()

{

int sum = 0, average =0, max =0;

Console.Write("请输入班级人数：");

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

int[]num = new int[number];

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

{

Console.Write("请输入第{0}个人的年龄：",i+1);

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

num[i] = old;

if(max < num[i] )

max = num[i];

sum += old;

}

average = sum / number;

Console.WriteLine("年龄总和为：{0}",sum);

Console.WriteLine("平均年龄为：{0}",average);

Console.WriteLine("最大年龄是：{0}\n",max);

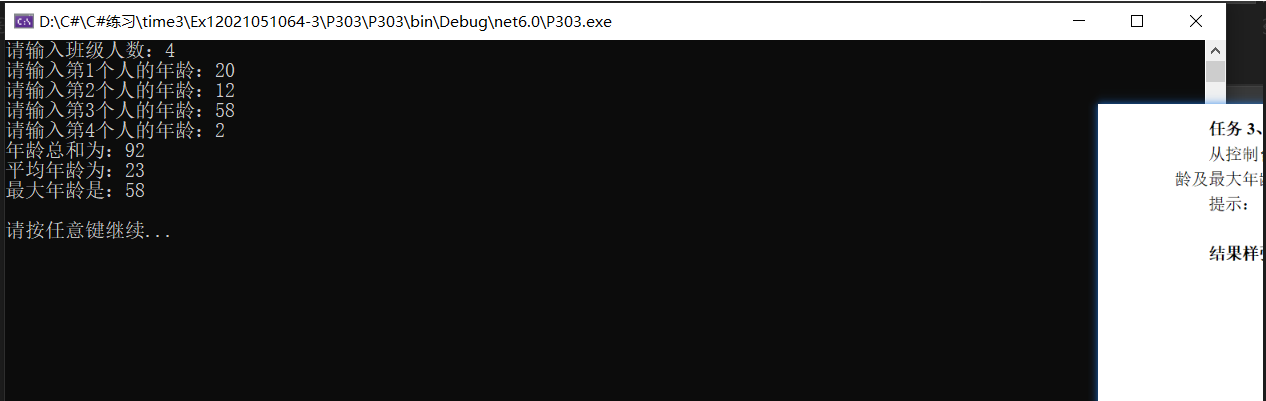
Console.WriteLine("请按任意键继续...");

Console.ReadKey();

}

}

//粘贴运行结果界面截图

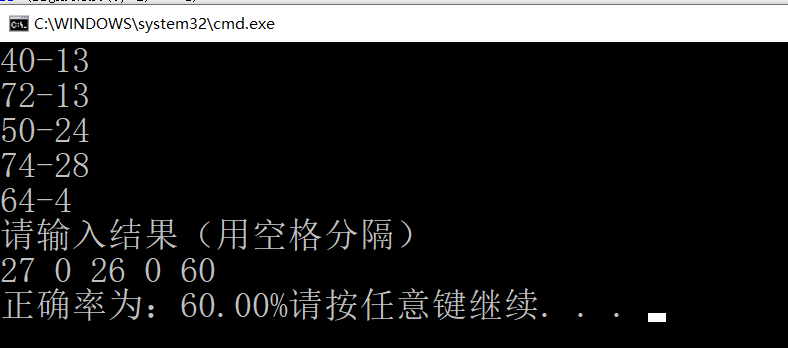


1. **扩展实验**

**1、实验一**

设计一个加减法自动生成器。利用随机数生成100以内操作数构成的两位数加减法算式5个，用户依次输入计算结果后，统计正确率。（35分钟）

运行结果如下图所示：



提示：

1. Random a = new Random();//生成随机数
2. b[i] = a.Next(0, 100);//把0-99的随机数赋值给数组
3. 两个操作数均由随机数生成，运算符也可用随机数做对应转换而成，例如随机生成0或1，1则设置为加运算，0则设置为减运算。
4. 先依次生成五个算式，同时计算运算结果放入一维数组，然后接收用户输入，分隔后，转换类型与数组元素依次对比，统计正确算式个数，计算正确率。

**源程序代码：**粘贴program.cs中代码

class Program

{

static void Main()

{

int[] num = new int[5];

int[] nums = new int[5];

int[] num1 = new int[5];

int[] result = new int[5];

for (int i = 0; i < 5; i++)

{

Random a = new Random();

num[i] = a.Next(0,100);

nums[i] = a.Next(0,100);

Random b = new Random();

if (b.Next(0, 2) == 0)

{

Console.WriteLine("{0}-{1}", num[i], nums[i]);

result[i] = num[i] - nums[i];

}

else

{

Console.WriteLine("{0}+{1}", num[i], nums[i]);

result[i] = num[i] + nums[i];

}

}

Console.WriteLine("请输入结果(用空格分开)");

string value = Console.ReadLine();

string[] vals = value.Split(' ');

for (int i = 0; i < vals.Length; i++)

{

num1[i] = int.Parse(vals[i]);

}

int Count = 0;

for (int i = 0; i < 5; i++)

{

if (result[i]== num1[i])

{

Count++;

}

}

double j = 5;

double Accuracy =(double) Count/j;

Console.WriteLine("正确率为：{0:P2}请按任意键继续...", Accuracy);

Console.ReadKey();

}

}

//粘贴运行结果界面截图

