第3章 流程控制和数组

3.1 实验目的

- (1) 熟练掌握控制台应用程序的代码编写和调试,以及运行方法。
- (2) 掌握选择结构的一般语法格式和应用。
- (3) 掌握 switch 语句的用法。
- (4) 掌握选择结构的嵌套的用法, 能灵活使用选择结构解决实际问题。
- (5) 掌握 while 循环语句的一般语法格式。
- (6) 掌握 for 循环语句的一般语法格式。
- (7) 掌握循环嵌套的语法格式。
- (8) 掌握一维数组的定义、初始化和数组元素的引用等。
- (9) 掌握遍历数组元素的方法和一维数组的选择法排序方法。
- (10) 掌握一维字符数组的定义、初始化和应用。
- (11) 掌握二维数组的定义和应用方法。
- (12) 掌握 foreach 语句的语法规则,能灵活使用 foreach 语句遍历数组元素。

3.2 实验任务

- (1) 实验1 选择结构应用(一)。
- (2) 实验2 选择结构应用(二)。
- (3) 实验3 选择结构应用(三)。
- (4) 实验4 循环结构应用(一)。
- (5) 实验5 循环结构应用(二)。
- (6) 实验6 循环结构应用(三)。
- (7) 实验7 一维数组。
- (8) 实验8 一维字符数组。
- (9) 实验9 二维数组应用。
- (10) 实验 10 foreach 语句的应用。
- (11) 实验 11 数组类的使用(Array)。

3.2.1 实验1 选择结构应用(一)

1. 实验内容

创建一个控制台应用程序项目,根据输入的学生成绩,显示该成绩相应的等级:优秀、 良好、中等、及格或不及格。

• 12 •

2. 实验目的

(1) 掌握选择结构的一般语法格式和应用。

- (2) 掌握多分支结构的应用。
- (3) 熟练掌握控制台应用程序的代码编写和调试,以及运行方法。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs301, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

```
namespace vs301
{
   class Program
   {
      static void Main(string[] args)
      {
         int score;
         string grade;
         Console.Write("请输入学生的学习成绩");
         score=Int32.Parse(Console.ReadLine());
         if (score>=90)
            grade="优秀!";
         else if (score>=80)
            grade="良好!";
         else if (score>=70)
            grade="中等!";
         else if (score>60)
            grade="及格!";
         else
           grade="不及格!";
        Console.WriteLine ("该学生的学习成绩等级为:{0}",grade);
      }
   }
}
```

(3)单击"调试"菜单中的"开始执行(不调试)"菜单项,或者按 Ctrl+F5 组合键, 在弹出的窗口中输入 97,按 Enter 键,输出结果为该学生的学习成绩等级为:"优秀!",如 图 3.1 所示。

C:\Windows\system32\cmd.exe	×
请输入学生的学习成绩 9?	*
这学生的学习观频寺级为500%。 请按任意键继续	Ε

图 3.1 成绩等级程序运行界面

• 13 •

3.2.2 实验 2 选择结构应用(二)

1. 实验内容

创建一个控制台应用程序项目,使用 switch 语句,根据输入的运算数和运算符,按照 其运算符的不同进行相应的计算并输出运算结果。

2. 实验目的

(1) 掌握 switch 语句的用法。

(2) 掌握选择结构的嵌套的用法, 能灵活使用选择结构解决实际问题。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs302, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

```
namespace vs302
{
   class Program
   {
      static void Main(string[] args)
       {
          double op1, op2,s=0;
         char op='+';
          op1 = double.Parse(Console.ReadLine());
          op =char .Parse ( Console.ReadLine());
          op2 = double.Parse(Console.ReadLine());
          switch (op)
          {
             case '+':
                s = op1 + op2;
                break;
             case '-':
                s = op1 - op2;
                break;
             case '*':
                s = op1 * op2;
                break;
             case '/':
                 if (op2 != 0)
                    s = op1 / op2;
                 else
                    Console.WriteLine("除数不能为 0");
                 break;
```

• 14 •

```
}
Console.WriteLine("{0}{1}{2}={3}",op1,op,op2,s);
}
}
```

(3)选择"调试"→"开始执行(不调试)"命令,或者按 Ctrl+F5 组合键,在弹出的窗口中输入 2.3,按 Enter 键,输入*,按 Enter 键,输入 1.2,按 Enter 键,输出结果为 2.3*1.2=2.76,如图 3.2 所示。



图 3.2 算术题程序的运行界面

3.2.3 实验3 选择结构应用(三)

1. 实验内容

创建一个控制台应用程序项目,实现猜数字游戏,产生一个随机数,输入一个值作为 猜测的值,输出猜的值比该随机数大或者小。

2. 实验目的

(1) 掌握随机数的用法,能使用随机数函数产生任意范围的随机数。

(2) 能熟练使用选择结构解决实际问题。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs303, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

```
namespace vs303
{
    class Program
    {
        static void Main(string[] args)
        {
            int guss = 0,r;
            Random rnd = new Random(); //创建一个存储随机数的变量
        r = (int)rnd.Next(1, 11); //生成一个1~10之间的随机整数
        }
    }
}
```

• 15 •

```
Console.WriteLine("请输入您猜的数是: ");
guss = int.Parse(Console .ReadLine ());
if(guss >r)
    Console .WriteLine ("您猜的数字偏大! ");
else if(guss <r)
    Console .WriteLine ("您猜的数字偏小! ");
else
    Console .WriteLine ("恭喜! 您猜中了! ");
}
</pre>
```

(3)选择"调试"菜单中的"开始执行(不调试)"命令,或者按 Ctrl+F5 组合键,在 弹出的窗口中输入 5,按 Enter 键,输出结果为"您猜的数字偏大!",如图 3.3 所示。

C:\Windows\system32\cmd.exe	- • •
请输入您猜的数是:	<u>^</u>
▶ 您猜的数字偏大! 请按任意键继续	=

图 3.3 猜数程序的运行界面

3.2.4 实验4 循环结构应用(一)

1. 实验内容

创建一个控制台应用程序项目,使用 while 语句计算 1+2+3+…+99 的值。

2. 实验目的

(1) 掌握 while 循环语句的一般语法格式。

(2) 掌握 while 循环的循环原理和运行规则。

(3) 能使用 while 循环语句编写代码解决实际循环问题。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs304, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

```
namespace vs304
{
```

• 16 •

```
class Program
{
    static void Main(string[] args)
    {
        int sum=0,i=1;
        while (i<100)
        {
            sum+=i; //循环变量是i
            i++; //i++用于改变循环变量的值
        }
        Console.WriteLine("sum={0}",sum);
     }
}</pre>
```

(3)选择"调试"菜单中的"开始执行(不调试)"命令,或者按 Ctrl+F5 组合键,在 弹出的窗口中显示输出结果为 sum=4950,如图 3.4 所示。

C:\Windows\system32\cmd.exe	
sum=4950 请按任音键继续	<u>^</u>
	E

图 3.4 求 1~99 的和的程序运行界面

3.2.5 实验 5 循环结构应用(二)

1. 实验内容

创建一个控制台应用程序项目,使用 for 循环语句,计算 1-1/2+1/3-1/4+···-1/99+1/100。 2. 实验目的

(1) 掌握 for 循环语句的一般语法格式。

(2) 掌握 for 循环的循环原理和运行规则。

(3) 能使用 for 循环语句编写代码解决实际问题。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs305, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

```
namespace vs305
{
    class Program
```

```
{
    static void Main(string[] args)
    {
        int i;
        double s=0;
        for (i = 1; i <= 100; i++)
        {
            s += Math .Pow ((-1) , (i + 1)) * 1 / (double)i;
        }
        Console.WriteLine("s={0}", s);
    }
}</pre>
```

(3)选择"调试"→"开始执行(不调试)"命令,或者按 Ctrl+F5 组合键,在弹出的 窗口中显示输出结果为 s=0.688172179310195,如图 3.5 所示。

C:\Windows\system32\cmd.exe	
s=0.688172179310195 请按任意键继续	^

图 3.5 算式程序的运行界面

3.2.6 实验6 循环结构应用(三)

1. 实验内容

创建一个控制台应用程序项目,使用 for 语句计算 1!+2!+3!+…+10! 的值。

2. 实验目的

(1) 掌握循环嵌套的语法格式。

- (2) 掌握多层循环嵌套的语法和应用。
- (3) 能使用多层循环嵌套代码编程解决实际问题。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs306, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

```
namespace vs306
{
    class Program
```

• 18 •

```
{
    static void Main(string[] args)
    {
        int i, k, m = 1, sum = 0;
        for (i = 1; i <= 10; i++)
        {
            for (k = 1; k <= i; k++)
                m = m * k;
                sum = sum + m;
                m = 1;
        }
        Console.WriteLine("1!+2!+3!+...+10!={0}", sum);
    }
}</pre>
```

(3)选择"调试"→"开始执行(不调试)"命令,或者按 Ctrl+F5 组合键,在弹出的 窗口中显示输出结果为 1!+2!+3!+…+10!=4037913,如图 3.6 所示。

C:\Windows\system32\cmd.exe	
1 ;+2;+3;+····+10;=4037913 请按任意键继续■	* W

图 3.6 1~10 阶乘的和程序运行界面

3.2.7 实验7 一维数组

1. 实验内容

创建一个控制台应用程序项目,创建一个一维数组并实现对数组元素的赋值,使用选 择法对数组进行从小到大排序后输出。

2. 实验目的

- (1) 掌握一维数组的定义、初始化,数组元素的引用等。
- (2) 掌握遍历数组元素的方法。
- (3) 掌握一维数组的选择法排序方法。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs307, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

namespace vs307

```
{
   class Program
   {
      static void Main(string[] args)
      {
         int i, j, k, m;
         int[] que = new int[] { 13, 25, 4, 5, 15, 78, 48, 50, 12, 100 };
         //定义数组并动态初始化
         for (i = 0; i < 10; i++)
         {
            k = i;
            for (j = i + 1; j < 10; j++) //从i的下一个元素起开始比较
            {
               if (que[j] < que[k])//比较数组元素
               {
                  k = j; //记录下标值
               }
            }
            if (k != i)
            {
              m = que[i];
               que[i] = que[k];
               que[k] = m;
            }
         }
         Console.WriteLine("输出排序后的结果是: ");
         for (i = 0; i < 10; i++)
         {
            Console.Write("{0}\t", que[i]);
         }
      }
 }
}
```

(3)选择"调试"→"开始执行(不调试)"命令,或者按 Ctrl+F5 组合键,在弹出的 窗口中显示输出结果为"4 5 12 13 15 25 48 50 78 100",如图 3.7 所示。

C:\Windows\system32\cr	nd.exe							×
输出排序后的结果是: 4 5 12 请按任意键继续	13	15	25	48	50	78	100	* III

图 3.7 排序程序运行界面

• 20 •

3.2.8 实验8 一维字符数组

1. 实验内容

创建一个控制台应用程序项目,创建一个一维字符型的数组,对数组元素进行初始化, 并逐个输出元素。

2. 实验目的

(1) 掌握一维字符数组的定义、初始化。

(2) 掌握一维字符数组的元素赋值、遍历等方法。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs308, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

```
namespace vs308
{
   class Program
   {
      static void Main(string[] args)
      {
         //定义并初始化 name 字符数组
         char[] name = new char[] { 'H', 'e', 'l', 'l', 'o', '', 'C', '#',
'!', '\0' };
         int ctr = 0;//建立一个计数器 ctr, 用于遍历数组元素
         //输出数组中的字符,直到遇到一个 ASCII 码为 0 的字符为止,结束循环
         while (name[ctr] != 0)
         {
            Console.WriteLine("{0}", name[ctr]);
            ctr++;
         }
     }
   }
}
```

(3)选择"调试"→"开始执行(不调试)"命令,或者按 Ctrl+F5 组合键,在弹出的 窗口中显示输出结果,如图 3.8 所示。

C:\Windows\system32\cmd.exe	
H	~
e	=
1	
c	
#	
请按仕意键继续。。。	

图 3.8 字符数组应用程序运行界面

• 21 •

3.2.9 实验9 二维数组应用

1. 实验内容

创建一个控制台应用程序项目,定义一个 3×4 的整型二维数组,创建数组,通过控制 台对数组元素赋值,求出其中的最大值并输出。

2. 实验目的

(1) 掌握二维数组的定义,数组元素赋值的方法。

(2) 掌握对二维数组元素进行遍历的方法。

(3) 能灵活掌握二维数组进行数据存储和数据处理的方法。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs309, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

```
namespace vs309
{
   class Program
   {
       static void Main(string[] args)
       {
          int [,] arr=new int[3,4];
          int i,j,max;
          for(i=0;i<3;i++)</pre>
              for (j=0; j<4; j++)</pre>
                 arr[i,j]=int.Parse (Console.ReadLine());
              max=arr[0,0];
          foreach(int x in arr)
          {
              if(x>max)
                 max=x;
          }
          Console.WriteLine("最大值是{0}",max);
       }
   }
}
```

(3)选择"调试"→"开始执行(不调试)"命令,或者按Ctrl+F5组合键,在弹出的 窗口中输入任意数组元素,输出结果如图 3.9 所示。

• 22 •



图 3.9 二维数组应用程序的运行界面

3.2.10 实验 10 foreach 语句的应用

1. 实验内容

创建一个控制台应用程序项目,使用 foreach 语句按照元素的顺序打印出一个二维数 组中的各个值。

2. 实验目的

(1) 掌握 foreach 语句的语法规则。

(2) 能够灵活使用 foreach 语句遍历数组元素。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs310, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

```
namespace vs310
{
    class Program
    {
        static void Main(string[] args)
        {
            double[,] values = { { 1.2, 2.3, 3.4, 4.5 }, { 5.6, 6.7, 7.8, 8.9 } };
            foreach (double elementValue in values)
                Console.Write("{0}?", elementValue);
    }
}
```

```
Console.WriteLine();
}
}
```

(3)选择"调试"→"开始执行(不调试)"命令,或者按Ctrl+F5组合键,在弹出的 窗口中输出结果,如图 3.10 所示。



图 3.10 foreach 程序的运行界面

3.2.11 实验 11 数组类的使用(Array)

1. 实验内容

定义两个数组,实现数组间的复制(使用数组类 Array 实现)。

2. 实验目的

掌握数组类 Array 中的方法并灵活运用。

3. 实验步骤

(1)运行 Visual Studio .NET 2013,选择"文件"→"新建"→"项目"命令,进入"新 建项目"对话框,在项目模板中选择"控制台应用程序",并在下方输入项目名称为 vs311, 设置保存路径为 E:\C#\ch03,其中路径可根据实际情况更换,单击"确定"按钮,进入项 目设计界面。

(2) 在 Program.cs 中添加如下代码:

```
namespace vs311
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] a=new int[5]{1,2,3,4,5};
            int[] b = new int[5];
            Array.Copy(a, b,5);
            foreach (int s in b)
                Console.WriteLine(s);
        }
    }
    24 •
```

}

(3)选择"调试"→"开始执行(不调试)"命令,或者按Ctrl+F5组合键,在弹出的窗口中输出结果,如图3.11所示。



图 3.11 数组类应用程序的运行界面