

C#程序设计



第3章 控制结构

杨琦

计算机教学实验中心

<http://ctec.xjtu.edu.cn>



第3章 控制结构

- 结构化程序设计方法的基本思想
- C#的基本控制结构
- C#的控制语句
- 结构化程序设计方法的基本思想
- C#的几种基本控制语句
- 熟悉使用伪代码的编程方法



3.1 程序的基本控制结构

- “软件危机” ——>结构化程序设计
- 20 世纪60年代以前，是个人设计、个人使用、个人操作、自给自足的私人化的软件生产方式
- 60年代中期，大容量、高速度计算机的出现
- 1968年北大西洋公约组织的计算机科学家正式提出解决方案——“软件工程”

解决方案：结构化程序设计



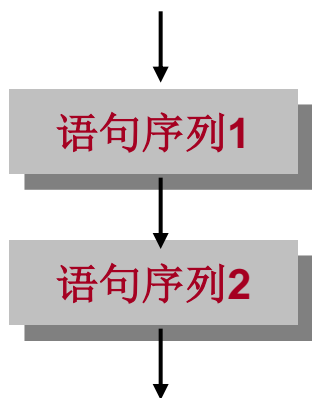
- 模块化
- 就是把程序划分为若干个部分，每个部分独立存放、完成一个特定的功能。
- 没有GOTO语句
- 程序就该仅有一个入口和一个出口
- 自顶向下、逐步求精的分解
- 结构化程序设计的三种基本结构是:顺序结构、选择结构和循环结构

3.1 程序的基本控制结构

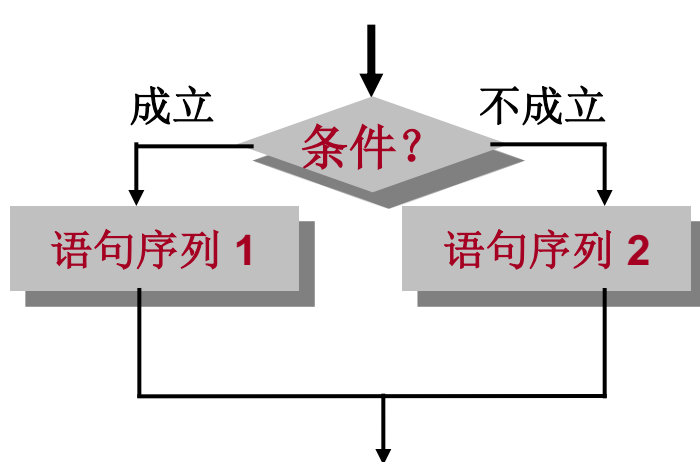


随着人们对程序的结构进行了深入的研究，逐步确定了程序设计的基本技术方法——**结构化程序设计**。

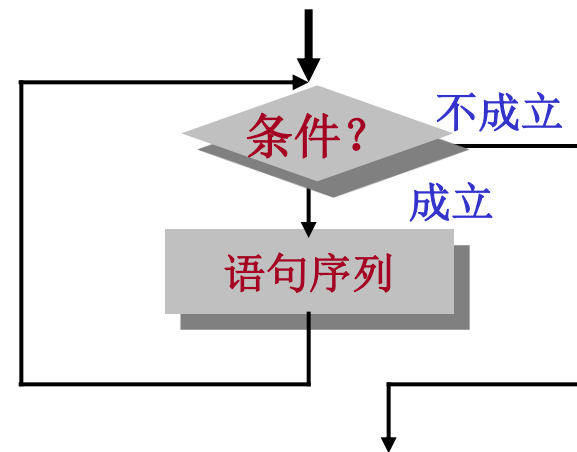
结构化程序设计方法的基本思想是任何程序都可以用三种基本结构表示，即**顺序结构**，**选择结构**，**循环结构**。由这三种基本结构经过反复嵌套构成的程序成为**结构化程序**。而这些结构中的各种程序流程语句就是**流程控制语句**。



顺序结构

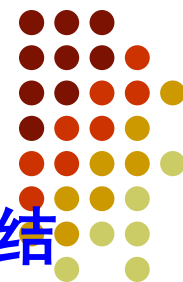


选择结构

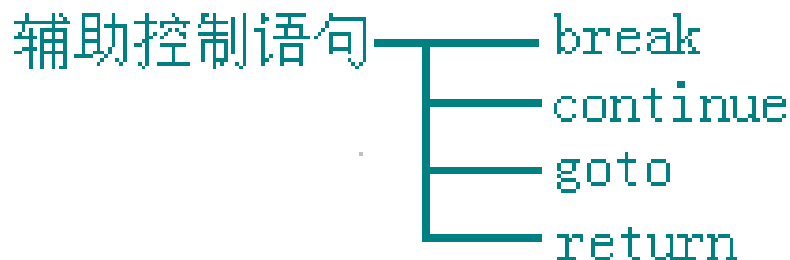
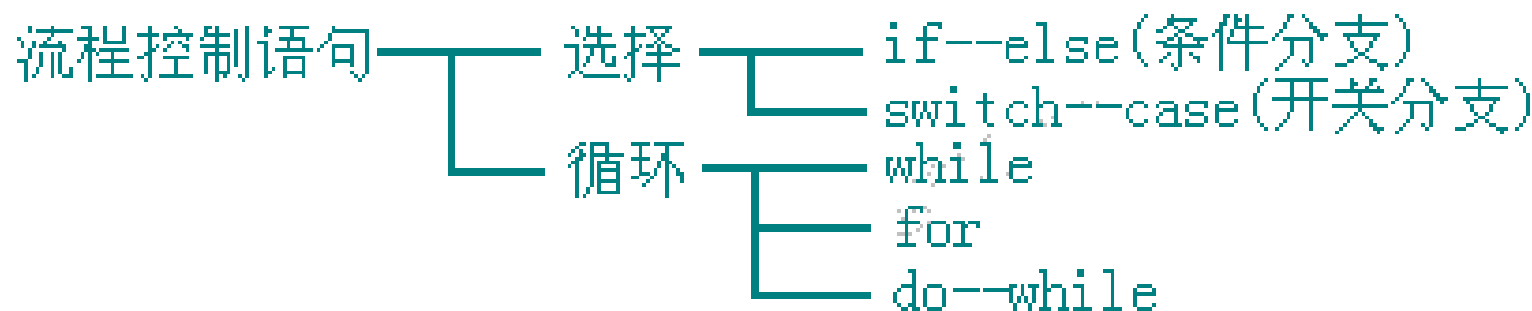


循环结构

3.2 C#语言的控制结构



- C#是一种很好的结构化程序设计语言，它提供了实现结构化程序所需的多种流程控制语句。分类如下：





1、顺序结构

- **顺序结构语句包括：**

- 说明语句
- 赋值语句
- I/O 语句
- 复合语句和空语句

```
{  
    <局部数据说明部分>  
    <执行语句段落>  
    {  
        <局部数据说明部分>  
        <执行语句段落>  
    }  
}
```

用花括号括起来的程序段落又称为分程序或者**复合语句**。

复合语句的一般形式：

```
{  
    <局部数据说明部分>  
    <执行语句段落>  
}
```

复合语句可以**嵌套**，即在复合语句中可以有其它复合语句

空语句：在C语言中经常使用只有一个分号的语句，它程为空语句在语法上占据一个语句的位置，但不具备任何可执行的功能。

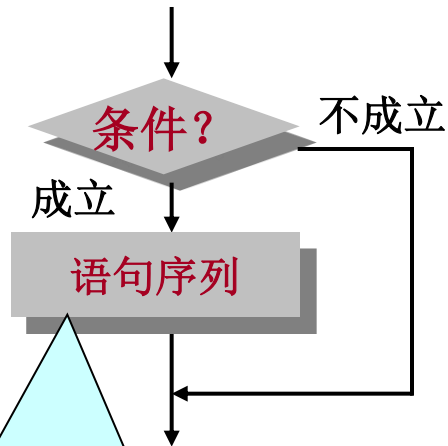
2、选择结构（之一）



- 一路分支

语句格式：

if（表达式）
语句序列

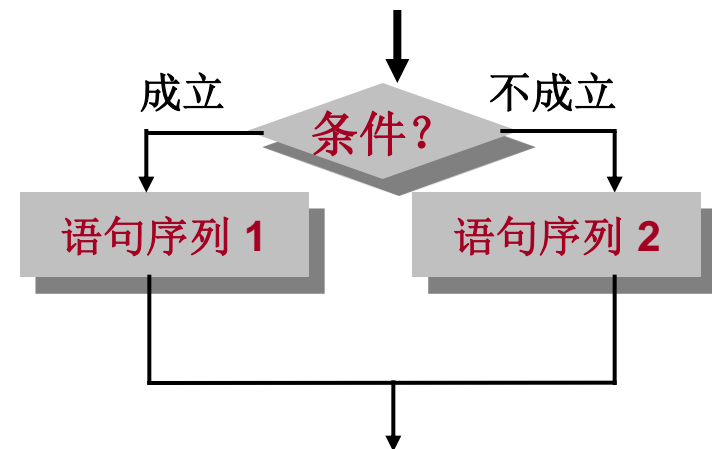


语句序列可以是一个语句，
也可以是复合语句结构。

- 两路分支

语句格式：

if（表达式）
语句序列1
else
语句序列2



【例3-1】 编程实现分段函数



$$y = \begin{cases} x + 1, & x < 0 \\ 1, & 0 \leq x < 1 \\ x^3, & 1 \leq x \end{cases}$$

【例3-1】 编程实现分段函数



```
1.  using System;
2.  class My {
3.      static int Main()
4.      {
5.          double x, y;
6.          Console.WriteLine( "请输入x的值 : ");
7.          x = Convert.ToDouble(Console.ReadLine());
8.          if (x < 0) {
9.              y = x + 1;
10.             Console.WriteLine( "x={0}, y=x+1={1}",x,y);
11.         }
```

【例3-1】 编程实现分段函数



```
1.      else if (x < 1) {                               // 0 ≤ x < 1
2.          y = 1;
3.          Console.WriteLine("x={0}, y={1}", x, y);
4.      }
5.      else                                             // 1 ≤ x
6.      {
7.          y = x * x * x;
8.          Console.WriteLine("x={0}, y=x*x*x={1}", x, y);
9.      }
10.     return 0;
11. }
12. }
```



选择分支的嵌套

if(表达式1) 语句1
else if(表达式2)语句2
else if(表达式3) 语句3

...

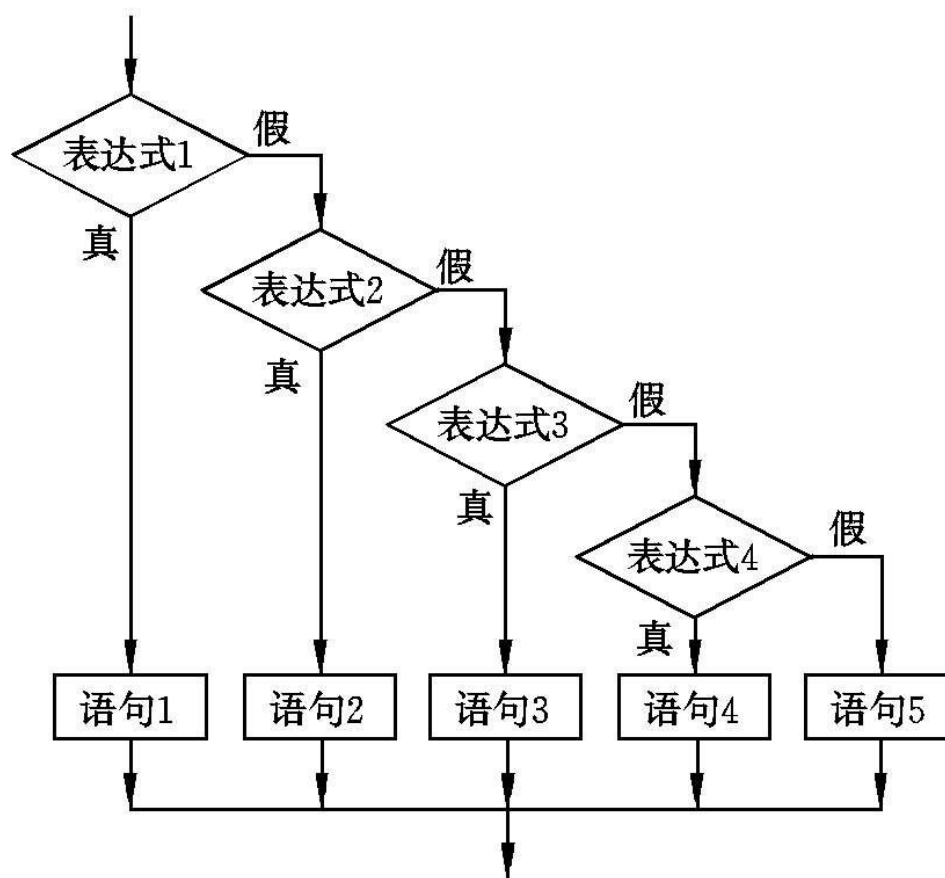
else if(表达式m) 语句m
else语句n

else总是与它上面的最近的if配对。

如果if与else的数目不一样，可以加花括弧来确定配对关系。

例如：

```
if(表达式1)
{ if (表达式2)语句1}
else语句2(内嵌if)
```



选择结构之二



多路（开关）选择语句 语句格式：

switch（整数表达式）

{ case 数值1:

语句序列1;

.....

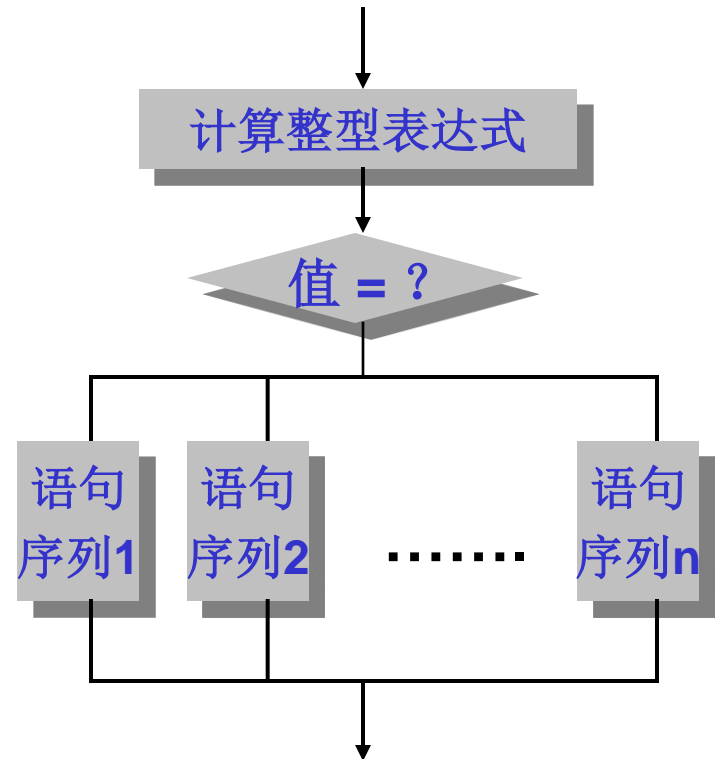
case 数值n:

语句序列n;

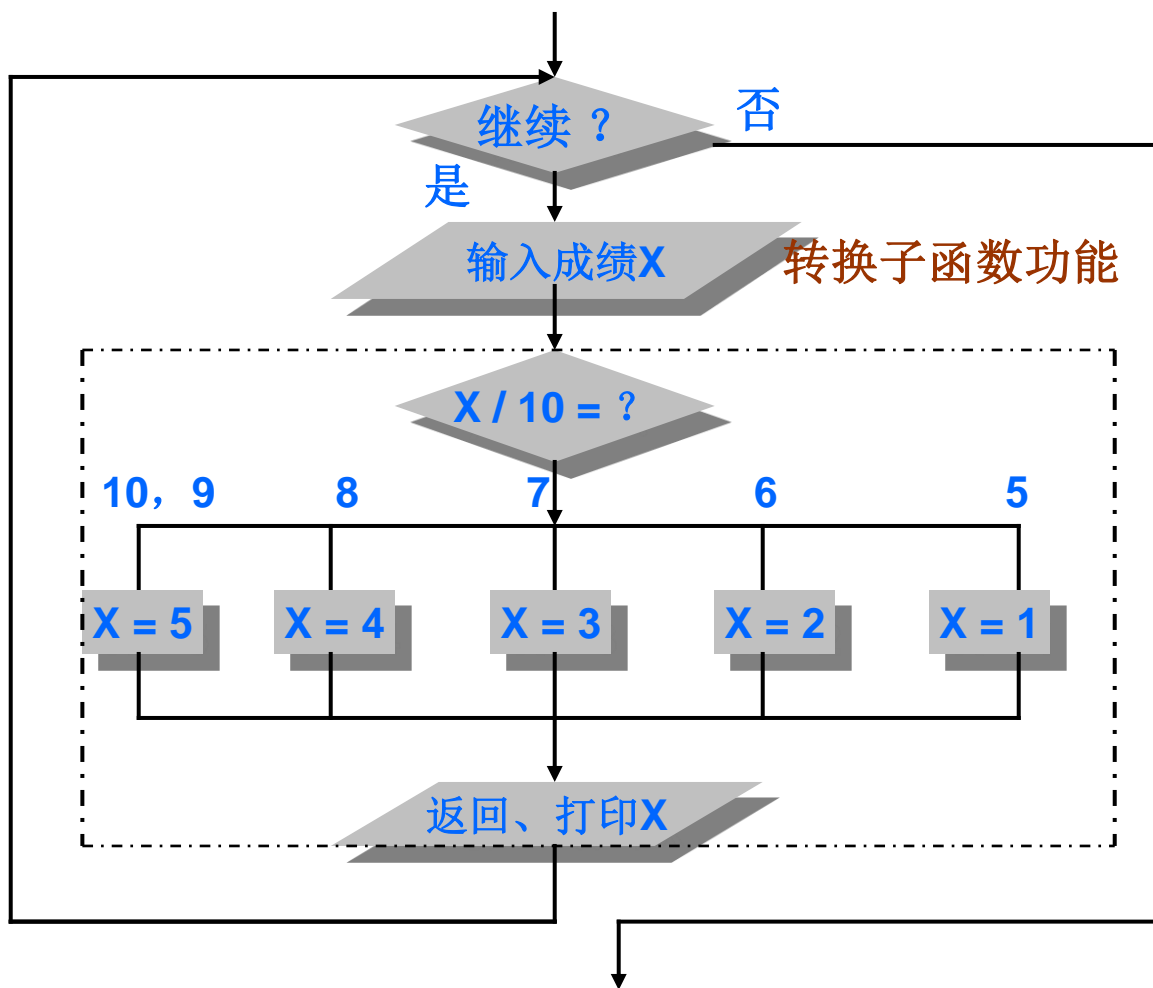
[default :

语句序列n+1;]

}



程序逻辑功能框图





【例3-2】将百分制转换为5级制

优秀：100-90分；
良好：80-89分；
中等：70-79分；
及格：60-69分；
不及格：60分以下。

输入和输出
请输入学生成绩：
85
转换前成绩是 85
转换后成绩是 良好

【例3-2】 将百分制转换为5级制



```
1.  class My{
2.      static int Main()  {
3.          int old_grade;
4.          string new_grade;
5.          Console.WriteLine("请输入学生成绩: ");
6.          old_grade = Convert.ToInt32(Console.ReadLine());
7.          switch (old_grade / 10)      {
8.              case 10:
9.              case 9:
10.                  new_grade = "优秀"; break;
11.              case 8:
12.                  new_grade = "良好"; break;
```


【例3-2】 将百分制转换为5级制



```
1.         case 7:
2.             new_grade = "中等"; break;
3.         case 6:
4.             new_grade = "及格"; break;
5.         default:
6.             new_grade = "不及格"; break;
7.     }
8.     Console.WriteLine("转换前成绩是 {0}", old_grade);
9.     Console.WriteLine("转换后成绩是 {0}", new_grade);
10.    return 0;
11. }
12. }
```

3、循环结构（之一）



- 当型循环

语句格式：

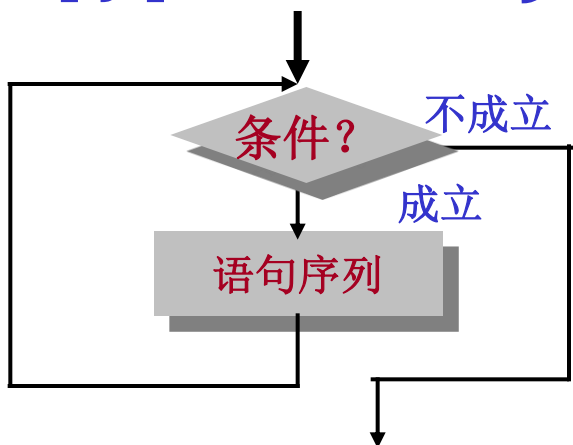
while（表达式）

[{

语句序列

[}]

} 循环体



- 直到型循环

语句格式：

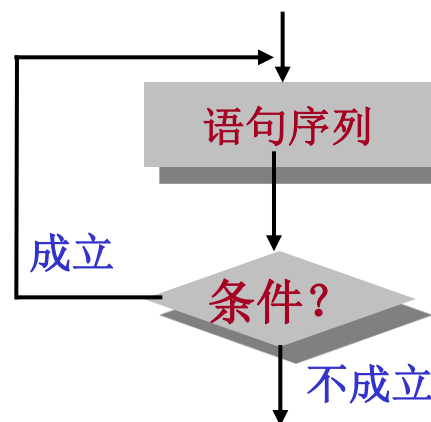
do

{

语句序列

} while（表达式）;

} 循环体



伪代码



- C语言的控制结构语句和自然语言结合起来描述算法
- 比画流程图省时、省力，且更容易转化为程序
- 不能运行，例如：

```
e = 1.0; n = 1; u = 1.0;  
while (通项u大于等于10-7)  
{  
    计算新的通项值 u = u/n;  
    将新通项值加到结果近似值上;  
    准备处理下一项 n = n+1;  
}
```

【例3-3】求e值

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!} + \dots$$



1、循环终止的条件是

$$\frac{1}{n!} < 10^{-7}$$

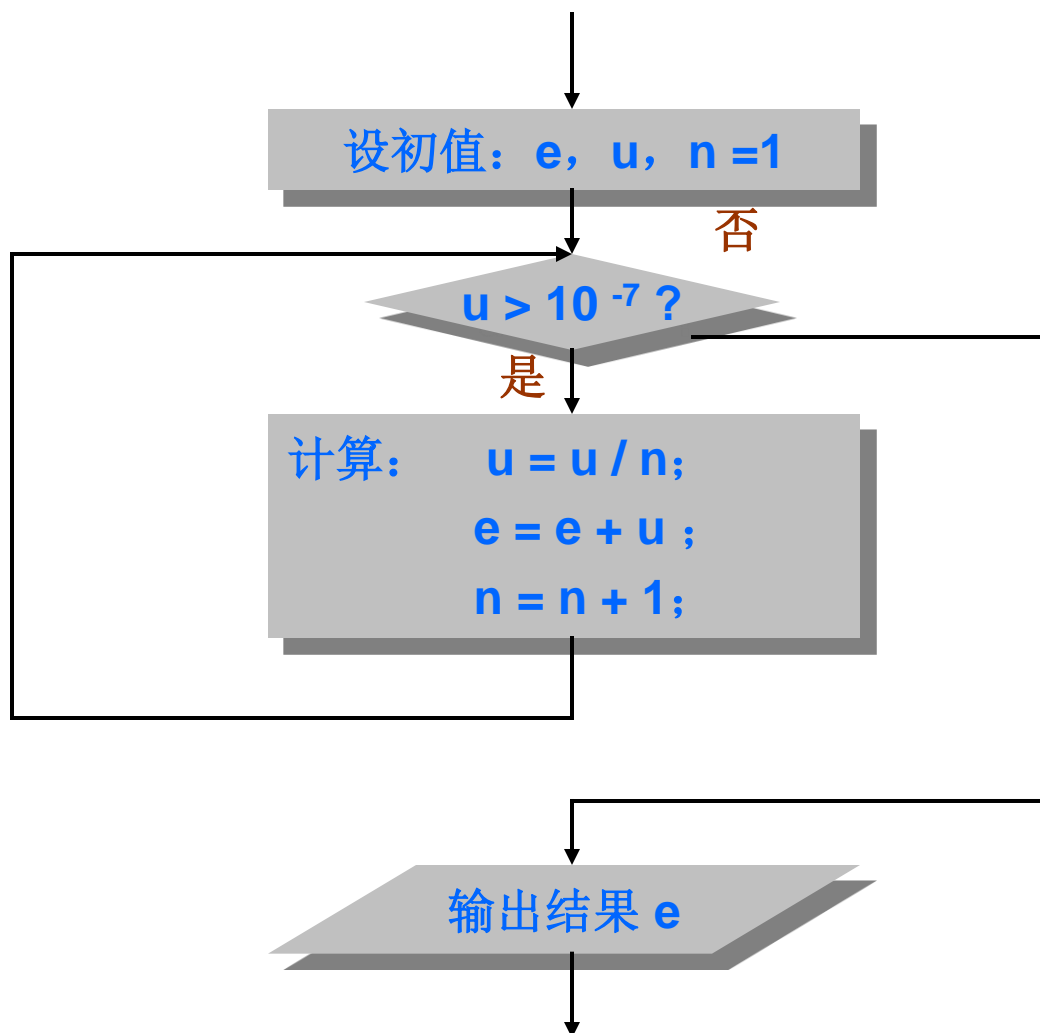
2、第i项 ($1 \leq i \leq n$) 可以写成 $u = u / i$ 。

$$\frac{1}{n!} = \frac{1}{(n-1)! * n}$$

3、累加和放在e中，可以写成 $e = e + u$;

4、要设一个计数器n，每循环一次， $n = n + 1$ 。

程序逻辑功能框图



【例3-3】求e值



```
1.  using System;
2.  class My {
3.      static int Main()  {
4.          double e = 1.0;
5.          double u = 1.0;
6.          int n = 1;
7.          while (u >= 1.0E-7)      {
8.              u = u / n;
9.              e = e + u;
10.             n = n + 1;
11.         }
12.         Console.WriteLine( "e = {0} ( n = {1} )",e,n);
13.         return 0;
```



【例3.4】 用do-while结构求e值

```
1.  static int Main()  {  
2.      double e = 1.0;  
3.      double u = 1.0;  
4.      int n = 1;  
5.      do      {  
6.          u = u / n;  
7.          e = e + u;  
8.          n = n + 1;  
9.      } while (u >= 1.0E-7);  
10.     Console.WriteLine("e = {0} ( n = {1} )", e, n);  
11.     return 0;  
12. }
```

For循环语句



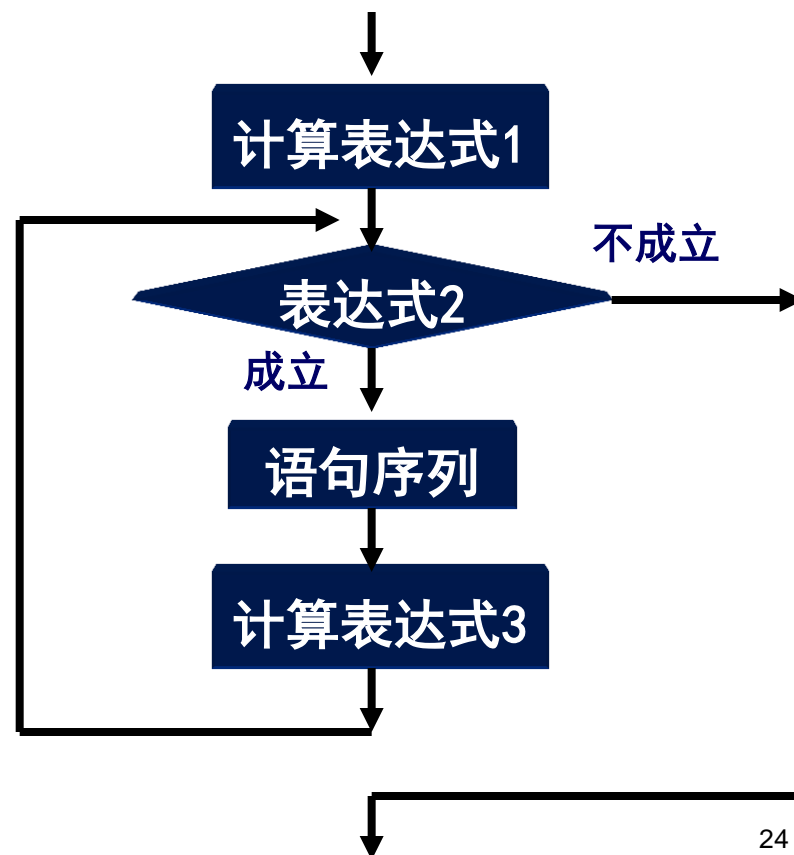
- For语句格式:

```
for (表达式1; 表达式2; 表达式3)
{
    语句序列
}
```

- For语句执行功能

- 问题:

用for结构求e值



【例3-5】 计算 $1+2+3+\dots+100$



```
1.  using System;
2.  class My{
3.      static int Main()  {
4.          int sum = 0;
5.          for (int i = 1; i <= 100; i++)
6.              sum = sum + i;
7.          Console.WriteLine( "1+2+3+...+100={0}", sum);
8.          return 0;
9.      }
10. }
```

【例3-6】 按照欧几里德算法

编写一个程序来求解任意两个正整数的最大公因数。



输入和输出

12

18

The maximum common divisor is 6

【例3-6】按照欧几里德算法



```
1.  class My{
2.      static int Main()  {
3.          int p = 12, q = 18, r;
4.          p = Convert.ToInt32(Console.ReadLine());
5.          q = Convert.ToInt32(Console.ReadLine());
6.          // 如果p < q, 交换p 和q
7.          if (p < q)      {
8.              r = p; p = q; q = r;
9.          }
10.         // 计算p 除以q 的余数r
11.         r = p % q;
```



【例3-6】按照欧几里德算法

```
1.      // 只要r 不等于0, 重复进行下列计算
2.      while (r != 0)      {
3.          p = q; q = r; r = p % q;
4.      }
5.      // 输出结果
6.      Console.WriteLine("The maximum common
7. divisor is " + q);
8.      return 0;
9.  }
10. }
```



【例3-7】利用迭代公式求平方根

设 $x = \sqrt{a}$ ，则迭代公式为 $x_0=1$;

$$x_{n+1} = \frac{(x_n + a/x_n)}{2}$$

迭代结束条件取相对误差

$$\left| \frac{x_{n+1} - x_n}{x_{n+1}} \right| < \varepsilon$$

输入和输出

Please input the value :

2

The square root of 2 is 1.41421356237309

【例3-7】 利用迭代公式求平方根



```
1.  using System;
2.  class My {
3.      const double EPS=1.0e-10;
4.      static int Main()
5.      {
6.          double a, x;
7.          Console.WriteLine("Please input the value : ");
8.          a=Convert.ToDouble(Console.ReadLine());
9.          double x0, x1;
10.         x1 = 1.0;
```

【例3-7】 利用迭代公式求平方根



```
1.      if (a > 0.0)
2.      {
3.          do
4.          {
5.              x0 = x1;
6.              x1 = (x0 + a / x0) / 2;
7.          } while (Math.Abs((x0 - x1) / x1) >= EPS);
8.          x = x1;
9.      }
10.     else
11.         x = a;
```

【例3-7】 利用迭代公式求平方根



```
1.         if (x < 0)
2.             Console.WriteLine( " The negative does not have square root" );
3.         else
4.             Console.WriteLine( "The square root of {0} is {1}",a,x );
5.         {0} is {1}",a,x );
6.         return 0;
7.     }
8. }
```


【例3-8】求 π 的近似值



- 计算公式：

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} +$$

【例3-8】求 π 的近似值



```
1.  using System;
2.  class My{
3.      static int Main()  {
4.          int s = 1;
5.          double n = 1.0, u = 1.0, pi = 0.0;
6.          while (Math.Abs(u) >= 1.0e-7)    {
7.              u = s / n;
8.              pi = pi + u;
9.              n = n + 2;
10.             s = -s;
11.         }
12.         Console.WriteLine( "pi = {0}", 4 * pi);
13.         return 0;
```



【例3-9】求水仙花数

- 算法分析：

1、用穷举法对100~999之间的每个数进行验证。验证公式为：

$$\text{hdn} = h * h * h + d * d * d + n * n * n$$

2、如何分解一个3位数的百位、十位和个位？是关键！

$$\text{百位} = n / 100 \quad [\text{整除100, 丢弃小数}]$$

$$\text{十位} = (n / 10) \% 10 \quad [\text{整除10, 得百十位。再对10取余数, 得十位数}]$$

$$\text{个位} = n \% 10 \quad [n \text{ 对10取余数, 得个位}]$$

例：371 $h = 371 / 100 = 3$

$$d = (371 / 10) \% 10 = 37 \% 10 = 7$$

$$n = 371 \% 10 = 1$$

【例3-9】求水仙花数



```
1.  using System;
2.  class My{
3.      static int Main()  {
4.          int n, i, j, k;
5.          for (n = 100; n <= 999; n++)  {
6.              i = n / 100;      // 取出n的百位数
7.              j = (n / 10) % 10;  // 取数n的十位数
8.              k = n % 10;      // 取出n的个位数
9.              if (n == i * i * i + j * j * j + k * k * k)
10.                  Console.WriteLine("{0} = {1}^3 + {2}^3 +
11. {3}^3", n, i, j, k);
12.          }
13.          return 0;
```

【例3-10】制作九九乘法表



$$1*1=1$$

$$1*2=2 \quad 2*2=4$$

$$1*3=3 \quad 2*3=6 \quad 3*3=9$$

$$1*4=4 \quad 2*4=8 \quad 3*4=12 \quad 4*4=16$$

$$1*5=5 \quad 2*5=10 \quad 3*5=15 \quad 4*5=20 \quad 5*5=25$$

$$1*6=6 \quad 2*6=12 \quad 3*6=18 \quad 4*6=24 \quad 5*6=30 \quad 6*6=36$$

$$1*7=7 \quad 2*7=14 \quad 3*7=21 \quad 4*7=28 \quad 5*7=35 \quad 6*7=42 \quad 7*7=49$$

$$1*8=8 \quad 2*8=16 \quad 3*8=24 \quad 4*8=32 \quad 5*8=40 \quad 6*8=48 \quad 7*8=56 \quad 8*8=64$$

$$1*9=9 \quad 2*9=18 \quad 3*9=27 \quad 4*9=36 \quad 5*9=45 \quad 6*9=54 \quad 7*9=63 \quad 8*9=72 \quad 9*9=81$$



【例3-10】制作九九乘法表

```
1.  using System;
2.  class My{
3.      static int Main()  {
4.          int i, j;
5.          for (i = 1; i < 10; i++)      {
6.              for (j = 1; j <= i; j++)
7.                  Console.Write("{0}*{1}={2}\t", j, i, i * j);
8.                  Console.WriteLine();
9.          }
10.         return 0;
11.     }
12. }
```

【例3-11】 $n!$ 的末尾中0的个数

- Please input a positive number: 50
- The number of zero in 50! is: 12



【例3-11】 $n!$ 的末尾中0的个数



```
1.  using System;
2.  class My{
3.      static int Main()  {
4.          int n;      int sum = 0;      int i, k;
5.          Console.WriteLine( "Pleasant input a positive number.");
6.          n=Convert.ToInt32(Console.ReadLine());
7.          for (i = 5; i <= n; i = i + 5)    //只有的倍数才含的因子
8.          {  int m = i;
9.              for (k = 0; m % 5 == 0; k++)
10.                  m = m / 5;
11.                  sum = sum + k;
12.          }
```


【例3-11】 $n!$ 的末尾中0的个数



```
1.      Console.WriteLine( "The number of zero in
2.  {0}! is: {1}", n, sum);
3.      return 0;
4.  }
5. }
```

结 束 语



- 学好程序设计语言的唯一途径是

上机练习

- 你的编程能力与你在计算机上投入的时间成

正比