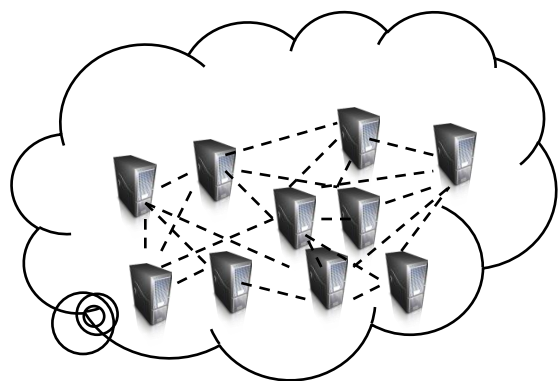


厦门大学非计算机专业本科生公共课 (2012-2013第2学期)



C语言程序设计

林子雨

厦门大学计算机科学系

E-mail: ziyulin@xmu.edu.cn

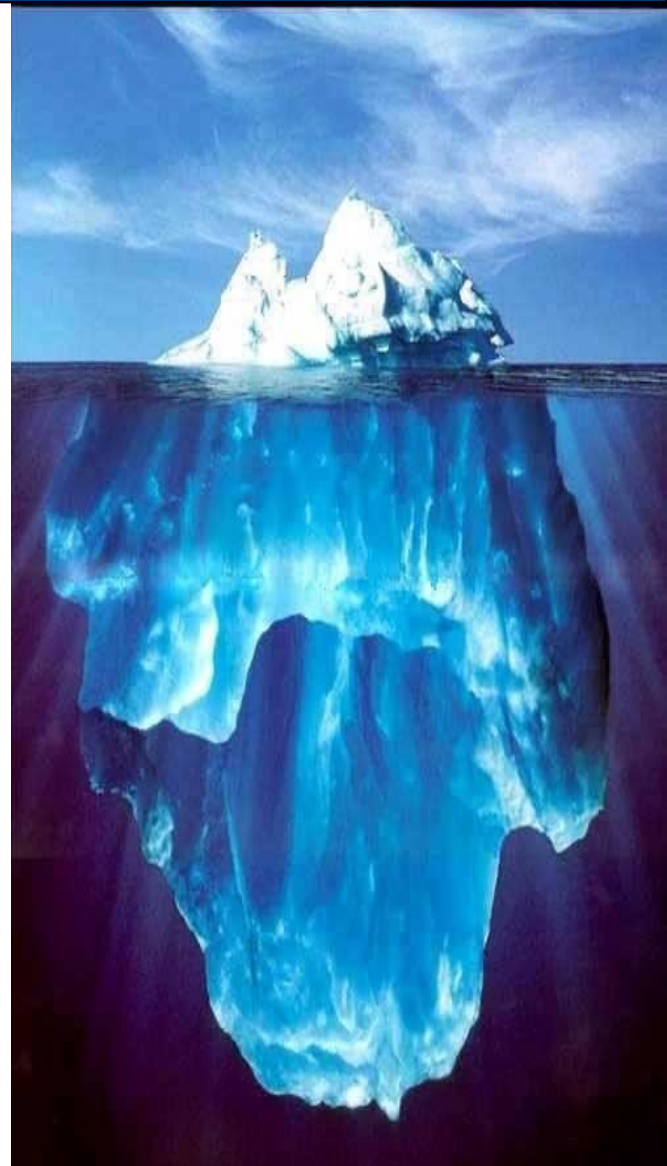
个人主页: <http://www.cs.xmu.edu.cn/linziyu> ▶▶





课程提要

- 第一章 绪论
- 第二章 C语言基础
- 第三章 结构化程序设计
- 第四章 选择结构
- **第五章 循环结构程序设计**
- 第六章 函数
- 第七章 编译预处理
- 第八章 数组
- 第九章 结构体、共用体和枚举类型
- 第十章 指针





第5章 循环结构程序设计

- 5.1 while语句
- 5.2 do...while语句(*)
- 5.3 for语句
- 5.4 break语句和continue语句(*)
- 5.5 循环的嵌套

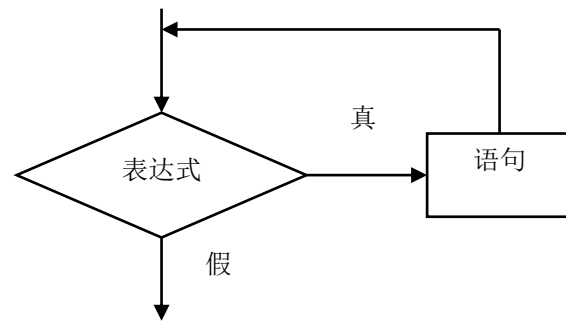




5.1 while语句

语法:

while(表达式)
语句



while循环

语义: 先计算表达式的值, 如果其值为非0 (“真”), 则执行循环体; 否则, 退出循环, 执行该循环结构后的下一条语句。每当执行完一次循环体以后, 再继续计算表达式的值, 如果值为非0, 则再次执行循环体。如此反复, 直到计算出的表达式值为0时, 退出循环。

注意:

- (1) 当循环体由多条语句构成时, 必须加大括号把循环体当做语句块。
- (2) **while**语句是先判断, 后执行。如果循环的条件一开始就不成立, 则循环体一次都不执行。
- (3) 循环体中必须有改变循环条件的语句, 否则循环不能终止, 形成无限循环 (或死循环)。





5.1 while语句

例5.1.1 求1-100中各偶数之和。

```
#include<stdio.h>
void main()
{
    int i,sum=0;
    i=2;
    while( i<=100)
        { sum=sum+i;
          i=i+2;
        } //块语句作为循环体
    printf("sum=%d\n",sum);
}
```





5.1 while语句

例5.1.2 打印出用户输入的每个正数的平方根（用0结束循环的执行）。

```
#include<math.h>
#include<stdio.h>
void main()
{
    float x;
    printf("Enter a positive number:");
    scanf("%f",&x);//输入第一个数，使循环控制变量x有确定值
    while(x>0)
    {
        printf("%f\t%f\n",x,sqrt(x));
        printf("Enter another positive number(or 0 to quit):");
        scanf("%f",&x);
    }
}
```





5.1 while语句

例5.1.3 用 $\pi/4 \approx 1 - 1/3 + 1/5 - 1/7 + \dots$ 公式求 π 的近似值，直到最后一项的绝对值小于 10^{-6} 为止。

```
#include <stdio.h>
#include <math.h>
main ( )
{
int s=1;
float n=1, t=1,pi,sum=0.0;
//t是循环控制变量。执行while语句之前，t必须有确定的值；而sum、n、t也必须有初值
while ( fabs(t) >= 1e-6 )    //某一项t绝对值小于10-6时退出循环
{
    sum=sum+t;
    n=n+2;           //修改分母
    s=-s;           //符号翻转
    t=s/n;          // 求出新的一项
}
pi=sum*4 ;
printf("pi=%f\n",pi);
}
```





5.1 while语句

例5.1.4 输入两个正整数 m 和 n ，求其最大公约数。

解：可采用欧几里德的辗转相除法：

- (1) 对于已知两个数 m 和 n ，使得 $m > n$ ；
- (2) m 除以 n 的余数为 r ；
- (3) 若余数 r 等于0，则转到步骤(5)，否则继续进行下一步；
- (4) 将除数 n 作为新的被除数 m ，把余数 r 作为新的除数 n ，求出新的余数 r ，转到步骤(3)；
- (5) 结束，这时 n 的值即为最大公约数。





5.1 while语句

例5.1.4 输入两个正整数m和n，求其最大公约数。

```
#include<stdio.h>
void main()
{
    int m,n,t,r;
    printf("请输入m,n:");
    scanf("%d,%d",&m,&n);
    if(m<n)
        {t=m;m=n;n=t;}//将大数放在m中，小数放在n中
    r=m%n;
    while(r)
    {
        m=n;//原除数作为新的被除数
        n=r;//原余数作为新的除数
        r=m%n;//求出新的余数r
    }
    printf("最大公约数为%d\n",n);//退出循环时n为最大公约数
}
```





5.1 while语句

例5.1.5 编写一程序，从键盘输入一组整数，以输入0作为结束，查找并显示这组数中的最小数。

```
#include <stdio.h>
main( )
{ int k,min;
  printf("请输入一组整数(以0结束):");
  scanf("%d",&k); //输入第一个数据
  min=k; //假定它是最小数
  while(k!=0) //当输入的数据不等于0，继续循环
  {
    if ( min>k ) //若新输入的数据比原来假定的最
      min=k; //更新最小数
    scanf("%d",&k); //为下一次循环读入新数据
  }
  printf("\nmin=%d\n",min);
}
```





5.1 while语句

【例子】有一楼梯，若每步跨2阶，则最后剩余1阶；若每步跨3阶，则最后剩余2阶；若每步跨4阶，则最后剩余3阶；若每步跨5阶，则最后剩余4阶。问该楼梯共有几阶。

第一种编程方法：

```
#include<stdio.h>
void main()
{
    int i=1;
    while(!(i%2==1 && i%3==2 && i%4==3 && i%5==4))
        i++;
    printf("阶梯共有%d阶\n",i);
}
```

或第二种编程方法：

```
#include<stdio.h>
void main()
{
    int i=1;
    while(i%2!=1 || i%3!=2 || i%4!=3 || i%5!=4)
        i++;
    printf("阶梯共有%d阶\n",i);
}
```





5.3 for语句

```
for(表达式1;表达式 2;表达式3)
```

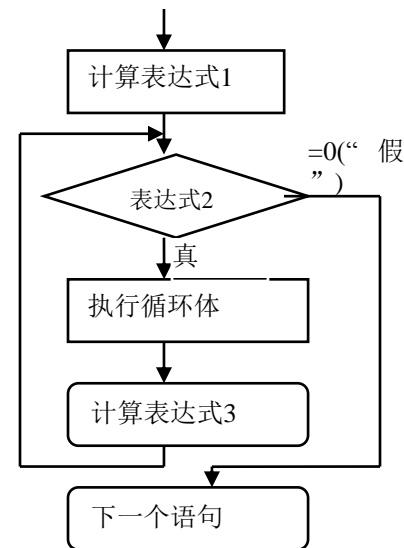
```
{  
    循环体  
}
```

说明:

- (1) 表达式1用来给循环控制变量赋初值，它首先被计算，而且只计算一次；表达式2是循环条件，决定循环是否继续，它在表达式1被计算后立即被计算，如果其值为非零，则执行循环体，否则退出循环；表达式3一般用来对循环控制变量进行增减操作，它在每次循环体被执行后都要立即被计算。
- (2) 循环体可以是一条或多条C语句。当循环体只由一条简单语句构成时，其前后的大括号可以省略。

执行过程:

- (1) 先计算表达式1，给循环控制变量赋初值（仅算一次）；
- (2) 计算表达式2，若值为非0，则执行循环体，然后执行步骤（3）。若值为0，则退出循环，转到步骤（5）。
- (3) 计算表达式3，让循环控制变量增或减；
- (4) 转回步骤（2）继续执行；
- (5) 执行for语句后面的一条语句。





5.3 for语句

例5.3.1 用for语句求 $1^2+2^2+3^2+\dots+100^2$ 的值，并将其结果放在变量sum中。

```
#include<stdio.h>
main()
{
    int i,sum=0; //变量sum为累加器，应初始化为0
    for(i=1;i<=100;i++)
    {
        //循环体开始
        sum=sum+i*i;//累加求和
    }
    //循环体结束
    printf("sum=%d\n",sum);//输出累加器结果
}
```





5.3 for语句

例5.3.2编一程序,从键盘输入某天参加体检的每个教师的体重(公斤),计算出他们的平均体重。

```
#include<stdio.h>
main()
{
    int i,num;                //i是循环控制变量
    float weight,total=0.0,avg; // total为累加器
    printf("请输入今天参加体检的教师人数:");
    scanf("%d",&num);
    for(i=1;i<=num;i++)      //圆括号后面不要随意加分号;
    {
        printf("请输入第%d个老师的体重: ",i);
        scanf("%f",&weight);
        total=total+weight;
    }
    avg=total/num;
    printf("avg=%5.1f\n",avg);
}
```





5.3 for语句

例5.3. 3输出100至9999中所有个位数和百位数之和为9的所有整数，并求它们的和（要求每行输出10项）。

```
#include<stdio.h>
main()
{
    int i,ge,bai,count=0,sum=0;
    for(i=100;i<=9999;i++)
    {
        ge=i%10;           // i%10的值就是i的个位数
        bai=i/100%10;      // i/100%10的值就是i的百位数
        if(ge+bai==9)
        {
            printf("%7d",i);    //按7个字符位、右对齐形式输出
            sum=sum+i;
            count++;           //每输出一项，count就增1
            if(count%10==0) printf("\n"); //控制每行输出10项
        }
    }
    printf("\ncount=%d\tsum=%d\n",count,sum);
}
```





5.3 for语句

例5.3.4 编一程序,求n!的值。

```
#include <stdio.h>
main ( )
{
    int i,n;
    double fac=1; // fac若为整型, 则求13! 时就会出现数据溢出现象;
    另注意fac的初值
    printf("Enter a positive integer: ");
    scanf("%d",&n );
    for(i=1; i<=n; i++ )
        fac=fac*i;
    printf( "%d!=%.0f\n",n,fac); //%. 0f表示不输出小数点后部分
}
```





5.3 for语句

例5.3.5编写一程序,将可显示的**ASCII**字符与它对应的十进制编码值一起输出。每行输出**9**个。

```
#include <stdio.h>
main ( )
{
    int count=0,ascii;
    for (ascii=32; ascii<128; ascii++)
    {
        printf("%c - %d\t", ascii,ascii);
        count++;
        if(count%9==0) printf("\n");
    }
}
```





5.3 for语句

例5.3.6编一程序。判断从键盘输入的自然数 m （大于1）是不是素数。素数是指除了1和它本身外，不能被其他任何整数整除的正整数。例如2，3，5，7,11，13等均为素数。

```
#include <stdio.h>
void main( )
{
    int i,m,flag=1;
    printf("请输入要判别的正整数m:");
    scanf("%d",&m);
    for(i=2;i<=m-1;i++) //用2、3、...、m-1分别除m，以检测它们是否能整除m
    if ( m%i==0 ) //判断i是否能整除m
    {
        flag=0; //设置不是素数标志
        i=m; //m不是素数,可提前退出循环
    }
    if ( flag==1 )
        printf("%d 是素数\n",m);
    else
        printf("%d 不是素数\n",m);
}
```





5.5 循环的嵌套

```
while()  
{  
  
    .....  
    while()  
    {  
  
        .....  
  
    }  
  
    .....  
  
}
```

```
while()  
{  
  
    .....  
    for(;;)  
    {  
  
        .....  
  
    }  
  
    .....  
  
}
```

```
for(;;)  
{  
  
    .....  
    for(;;)  
    {  
  
        .....  
  
    }  
  
    .....  
  
}
```





5.5 循环的嵌套

【例子】有一张100元钞票，要换成100张5元、1元和5角的零钞，要求各种零钞至少1张。输出各种零钞组合。

第一种编程方法：

```
#include<stdio.h>
void main()
{
    int i,j,k;
    for(i=1;i<=100;i++)
        for(j=1;j<=100;j++)
        {
            k=100-i-j;
            if(5*i+j+0.5*k==100)
                printf("i=%d\tj=%d\tk=%d\n",i,j,k);
        }
}
```





5.5 循环的嵌套

【例子】有一张100元钞票，要换成100张5元、1元和5角的零钞，要求各种零钞至少1张。输出各种零钞组合。

第二种编程方法：

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
    int i,j,k;
```

```
    for(i=1;i<=100;i++)
```

```
        for(j=1;j<=100;j++)
```

```
            for (k=1;k<=100;k++)
```

```
                if(5*i+j+0.5*k==100 && i+j+k==100)
```

```
                    printf("i=%d\tj=%d\tk=%d\n",i,j,k);
```

```
}
```





5.5 循环的嵌套

例5.5.1打印九九乘法口诀表。

```
#include <stdio.h>
void main( )
{
    int i,j;
    printf("\t\t\t乘法口诀表\n");
    for( i=1;i<=9;i++)                //共输出9行
    {
        for( j=1;j<=9;j++)            //每行输出9列
            printf("%d*%d=%d\t",i,j,i*j);
        printf("\n");                 //每输出一行后换行
    }
}
```





5.5 循环的嵌套

例5.5.2 编写一个程序，显示如下图案。

```
*
***
*****
*****
*****

#include<stdio.h>
void main()
{
    int i,j;
    for(i=1;i<=5;i++)                //共输出5行
    {
        for(j=1;j<=5-i;j++) putchar(' ');    //输出每行星号前的空格
        for(j=1;j<=2*i-1;j++) putchar('*');  //输出每行的星号
        putchar('\n');                    //每输出一行后换行
    }
}
```





附件：课程教材（2012-2013第2学期）

- 《C语言程序设计（第2版）》
- 清华大学出版社，黄保和，江弋 编著
- 版次：2011年10月第2版
- ISBN:978-7-302-26972-4
- 定价：35元



附件：课程和班级网站（2012-2013第2学期）

- 课程介绍网站：

<http://dblab.xmu.edu.cn/node/124>

- 班级网站：

<http://dblab.xmu.edu.cn/node/347>



附件：课程教师和助教（2012-2013第2学期）



主讲教师：林子雨

单位：厦门大学信息科学与技术学院计算机科学系
办公地点：福建省厦门市思明区厦门大学海韵园
E-mail: ziyulin@xmu.edu.cn
个人主页： <http://www.cs.xmu.edu.cn/linziyu>

助教：刘颖杰

单位：厦门大学计算机科学系2012级硕士研究生
E-mail: 376339705@qq.com
手机：18020761782

The background of the slide features a blue gradient with several faint, light-blue silhouettes of people. At the top, there are two groups of people standing and holding hands. On the right side, a person is shown in profile, looking towards the center. At the bottom left, two people are shown in profile, facing each other as if in conversation. The overall theme is one of community and collaboration.

Thank You!

Department of Computer Science, Xiamen University, March 26, 2013