

# 循环控制

Xuehui Song

ECE, PKUSZ

March 30, 2013



## Definition

循环结构是程序中一种很重要的结构。其特点是,在给定条件成立时,反复执行某程序段,直到条件不成立为止。给定的条件称为循环条件,反复执行的程序段称为循环体。C语言提供了多种循环语句,可以组成各种不同形式的循环结构。

- 用goto 语句和if 语句构成循环;
- 用while 语句;
- 用do-while 语句;
- 用for 语句;

# while语句

## Definition

while 语句的一般形式为:

**while(表达式)**  
语句

其中表达式是循环条件,语句为循环体。while 语句的语义是:计算表达式的值,当值为真(非0)时,执行循环体。

while.pdf

# while例子

用while语句求 $\sum_{i=1}^n i^3$

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n, i = 0, sum = 0;
    scanf("%d", &n);
    while(i <= n) {
        sum += i * i * i;
        ++ i;
    }
    printf("sum=%d\n", sum);
    return 0;
}
```

while\_e.pdf

## Definition

do-while 语句的一般形式为:

**do**

语句

**while(表达式);**

do-while先执行循环中的语句,然后再判断表达式是否为真,如果为真则继续循环;如果为假,则终止循环。因此, do-while 循环至少要执行一次循环语句。其执行过程可用右图表示。

do\_while.pdf

# do-while例子

用do-while语句求 $\sum_{i=1}^n i^3$

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n, i = 0, sum = 0;
    scanf("%d", &n);
    do {
        sum += i * i * i;
        ++i;
    } while(i <= n);
    printf("sum=%d\n", sum);
    return 0;
}
```

## Definition

在C语言中,for语句使用最为灵活,可以取代while语句。它的一般形式为:

**for(表达式1;表达式2;表达式3)  
语句**

它的执行过程如下:

- ① 先求解表达式1。
- ② 求解表达式2,若为真(非0),则执行指定的内嵌语句,然后执行第3步;若为假(0),则结束循环
- ③ 求解表达式3。
- ④ 转回上面第2步继续执行。

for.pdf



# for例子

用for语句求 $\sum_{i=1}^n i^3$

```
#include <stdio.h>
int main() {
    int n, i, sum;
    scanf("%d", &n);
    for (i = 0, sum = 0; i <= n; ++i) {
        sum += i * i * i;
    }
    printf("sum=%d\n", sum);
}
```

**for(循环变量赋初值;循环条件;循环变量增量)  
语句**

# for 注意事项 I

- ① for 循环中的“表达式1(循环变量赋初值)”、“表达式2(循环条件)”和“表达式3(循环变量增量)”都是选择项, 即可以缺省, 但“;”不能缺省
- ② 省略了“表达式1(循环变量赋初值)”, 应在for语句之前给循环变量赋初值。

```
int n, i = 0, sum = 0;  
for (; i <= n; ++i) {  
    sum += i * i * i;  
}
```

- ③ 省略了“表达式2(循环条件)”, 默认循环条件始终为真, 则不做其它处理时便成为死循环。

```
for (i = 0; ; ++i) {  
    sum += i * i * i;  
}
```

修改为:

## for 注意事项 II

```
for (i = 0; ; ++i) {  
    if (i > n) break;  
    sum += i * i * i;  
}
```

- ④ 省略了“表达式3(循环变量增量)”，则不对循环控制变量进行操作,这时可在语句体中加入修改循环控制变量的语句。

```
for (i = 0; i <= n;) {  
    sum += i * i * i;  
    i = i + 1;  
}
```

- ⑤ 省略了“表达式1(循环变量赋初值)”和“表达式3(循环变量增量)”，只保留“表达式2(循环条件)”。

```
for (; i <= n;) {  
    sum += i * i * i;  
    i = i + 1;  
}
```

## for 注意事项 III

相当于：

```
while (i <= n) {  
    sum += i * i * i;  
    i = i + 1;  
}
```

⑥ 3 个表达式都可以省略。

```
for ( ; ; ) {  
    sum += i * i * i;  
}
```

相当于：

```
while (1) {  
    sum += i * i * i;  
}
```

# 循环嵌套

## Definition

一个循环体内又包含一个完整的循环结构，称为循环的嵌套。内嵌的循环中还可以嵌套循环，这就是多层循环。

输出99乘法表

```
#include <stdio.h>
int main() {
    int i, j;
    for (i = 1; i <= 9; ++i) {
        for (j = 1; j <= 9; ++j) {
            printf("%5d", i * j);
        }
        printf("\n");
    }
}
```

# 几种循环的比较

- 四种循环都可以用来处理同一个问题,一般可以互相代替。但一般不提倡用goto 型循环。
- while 和do-while 循环,循环体中应包括使循环趋于结束的语句。for 语句功能最强。
- 用while 和do-while 循环时,循环变量初始化的操作应在while 和do-while 语句之前完成,而for 语句可以在表达式1 中实现循环变量的初始化。

## Definition

break 语句通常用在循环语句和开关语句中。

- ① 当break 用于开关语句switch 中时,可使程序跳出switch 而执行switch 以后的语句
- ② 当break 语句用于do-while、for、while 循环语句中时,可使程序终止循环而执行循环后面的语句,通常break 语句总是与if 语句联在一起。即满足条件时便跳出循环。

# continue语句

## Definition

continue 语句的作用是跳过循环本中剩余的语句而强行执行下一次循环。continue 语句只用在for、while、do-while 等循环体中,常与if 条件语句一起使用。

求1到10中, 不是3倍数的数的阶乘的和, 也就是 $1! + 2! + 4! + \dots + 10!$

```
#include <stdio.h>
int main() {
    int sum = 0, fac, i, j;
    for(i = 1; i <= 10; ++i) {
        if (i % 3 == 0) continue;
        fac = 1;
        for(j = 1; j <= i; ++j) fac = fac * j;
        sum += fac;
    }
}
```



# 例子

- ① 用  $\pi \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$  公式求  $\pi$  的近似值，直到某一项的绝对值小于  $10^{-8}$  为止
- ② 输入  $n$ ，统计1到  $n$  中回文数字的个数，所谓回文数就是从左往右和从右往左看是一样的数，如33，151，2442等
- ③ 输入  $n$ ，统计1到  $n$  中素数的个数
- ④ NOIP提高组2005 谁拿了最多奖学金  
<https://vijos.org/p/1001>
- ⑤ NOIP提高组2004 津津的储蓄计划  
<https://vijos.org/p/1096>

faq.jpg