



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

编程实现循环

- ◆ 1、while循环
- ◆ 2、循环三要素

3.5 程序循环结构

- ▶ 1. while语句
- ▶ 2. do语句
- ▶ 3. for语句

3.5.1 while语句

- ▶ while语句
- ▶ while语句的作用是计算给定的表达式，根据结果判定循环执行语句，语句形式为：

```
while (表达式) 语句;
```

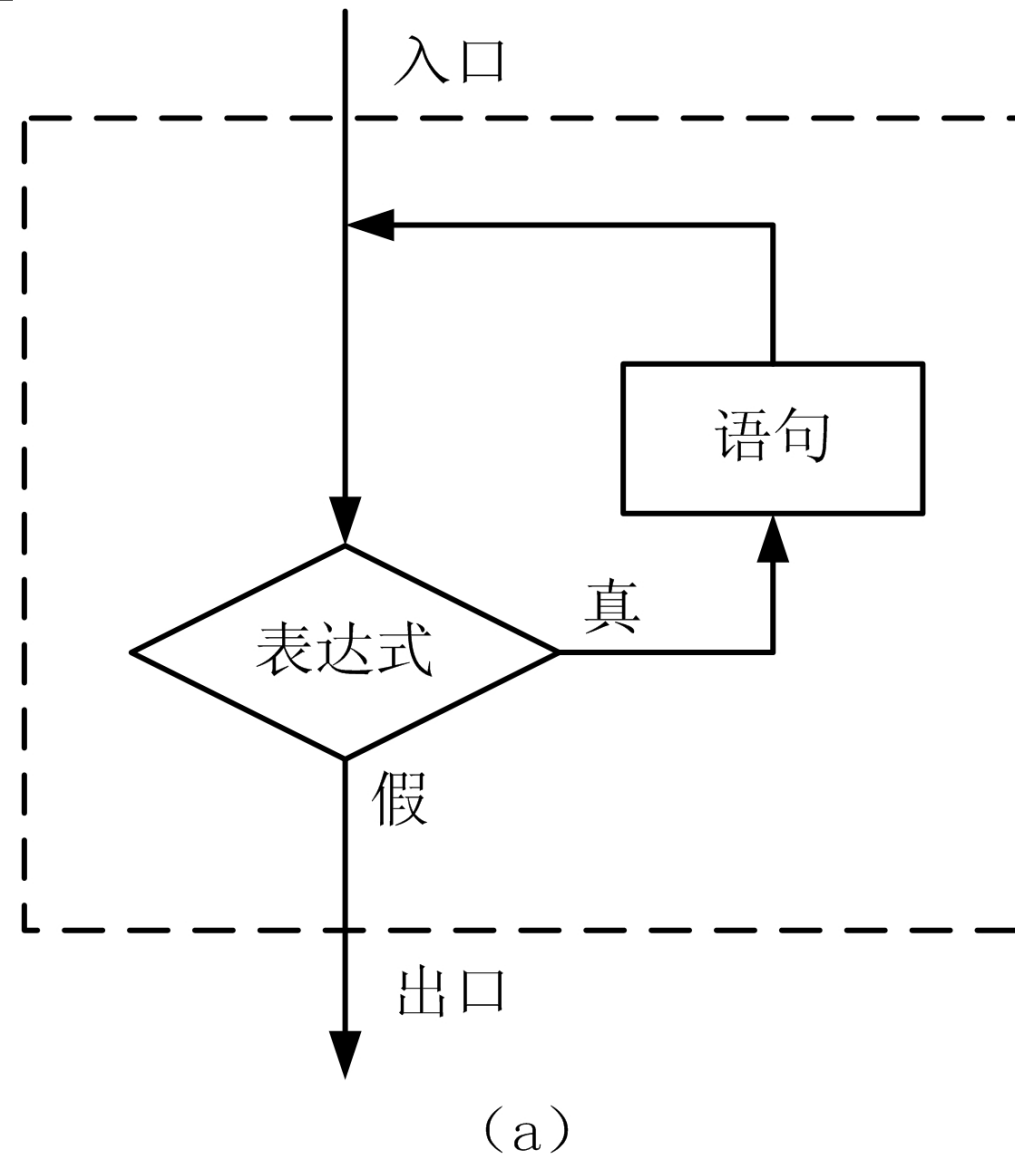
- ▶ 其中的语句称为子语句，又称循环体，圆括号内的表达式称为循环条件。

3.5.1 while语句

- ▶ while语句的执行过程是：
- ▶ ①计算表达式，无论表达式为何种类型均将这个值按逻辑值处理；
- ▶ ②如果值为真，则执行子语句，然后重复①；
- ▶ ③如果值为假，则while语句结束，执行后续语句。

3.5.1 while语句

图3.8 while语句执行流程



3.5.1 while语句



【例3.8】

求

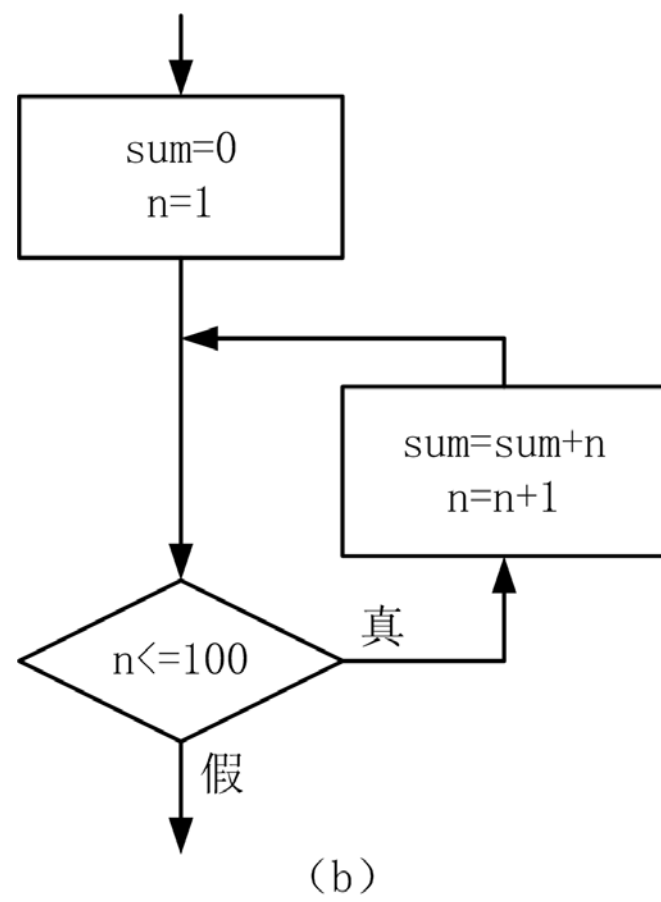
$$s = \sum_{n=1}^{100} n$$

即 $s = 1 + 2 + 3 + \cdots + 100$

3.5.1 while语句



例题分析



3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1,sum=0;
5     while (n<=100) { //循环直到n大于100
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n",sum);
10    return 0;
11 }
```

3.5.1 while语句

例3.8

程序运行屏幕



3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1, sum=0;
5     while (n<=100) { //循环直到n大于100
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n", sum);
10    return 0;
11 }
```

3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1,sum=0;
5     while (n<=100) { //循环直到n大于100
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n",sum);
10    return 0;
11 }
```

3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1,sum=0;
5     while (n<=100) { //循环直到n大于100
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n",sum);
10    return 0;
11 }
```

3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1, sum=0;
5     while (n<=100) { //循环直到n大于100
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n", sum);
10    return 0;
11 }
```

把第4行叫循环初始，即进入循环前的初始计算过程。不给n和sum赋初值可不可以？答案是否定的，如果n没有确定的值那么第5行的“n<=100”不合逻辑。同理，sum=sum+n的累加也就成问题。

3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1, sum=0;
5     while (n<=100) { //循环直到n大于100
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n", sum);
10    return 0;
11 }
```

n和sum随意赋值可不可以？答案是否定的，例如n=2和sum=0，则计算结果为 $0 + 2 + 3 + \dots + 100$ ，例如n=1和sum=1，则计算结果为 $1 + 1 + 2 + \dots + 100$ ，因此n和sum的值是不可随意给的。

3.5.1 while语句

举例

```
1 #include <stdio.h>
2 int main()
3 {
4     while (n<=100) { //循环直到n大于100
5         n=1,sum=0; //在这里赋值, 则每次循环均要执行
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n",sum);
10    return 0;
11 }
```

将n和sum赋值放到while语句中可不可以？答案是否定的
每次进入循环体，n或sum就被重新赋值，根本无法累加。
将n和sum赋值放到while语句后去做可不可以？答案更是否定的。

3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1, sum=0;
5     while (n<=100) { //循环直到n大于100
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n", sum);
10    return 0;
11 }
```

把第5行叫循环条件，即判断是否继续循环的条件或循环终止的条件。
如果把“while (n<=100)”写成“while (n<=200)”，那么计算结果就是 $0+1+2+\cdots+200$ ；

3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1, sum=0;
5     while (n<=100) { //循环直到n大于100    n=100
6         sum=sum+n; //累加和                "死循环"
7         n=n+1;
8     }
9     printf("sum=%d\n", sum);
10    return 0;
11 }
```

如果写成“while (n=100)”，由于是把100赋给n且按逻辑值来理解恒为真，则while语句的循环条件永远为真，循环不能结束了，我们称这样的循环为“死循环”。显然，循环条件也不是随意设定的。

3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1, sum=0;
5     while (n<=100) { //循环直到n大于100
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n", sum);
10    return 0;
11 }
```

n=n+2
0+1+3+5+...+99

把第7行叫循环控制，即让循环条件趋向结束的计算过程。如果没有“n=n+1”，那么“while (n<=100)”就恒为真，循环同样为死循环；如果改成“n=n+2”，那么结果就是显然，必须有符合算法要求的循环控制。

3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1,sum=0;
5     while (n<=100) { //循环直到n大于100
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n",sum);
10    return 0;
11 }
```

第6行是逻辑意义上的循环体，即循环目的的程序步骤。

3.5.1 while语句

例3.8

```
1 #include <stdio.h>
2 int main()
3 {
4     int n=1, sum=0;
5     while (n<=100) { //循环直到n大于100
6         sum=sum+n; //累加和
7         n=n+1;
8     }
9     printf("sum=%d\n", sum);
10    return 0;
11 }
```

循环初始

循环条件

循环目的

循环控制

从中发现循环结构有三要素。

3.5.1 while语句

- ▶ 循环结构有三要素：
- ▶ 循环初始、循环条件和循环控制。
- ▶ 编写循环程序，就要精确设计三要素。循环初始发生在循环之前，使得循环“就绪”；循环条件是循环得以继续或终止的判定，而循环控制是在循环内部实现循环条件的关键过程。循环体可以直接或间接利用三要素来达到计算目的，也可以与三要素无关。

3.5.1 while语句

- ▶ while语句的说明。
- ▶ (1) while语句的循环体既可以是简单语句，又可以是复合语句或控制语句，但必须是“一个语句”的语法形式。在实际编程中，当循环体有多条语句时使用复合语句。
- ▶ (2) 从while语句的执行流程图可知，while语句有一个入口和出口。从流程图的虚框上来看，while语句所形成的循环结构是可以抽象为顺序结构的一步。反过来提示我们在编程时可以先抽象设计顺序结构的一步，再使用循环结构细化。

3.5.1 while语句

- ▶ (3) 在循环中应该有使while表达式趋向假的操作，否则表达式恒为真，循环永不结束，成为死循环。
 - 有时在while条件后面不小心额外添加分号（；），往往会彻底改变循环的意图，例如：

```
1 while (i != fun()) ; //循环子语句是空语句
2     i++; //循环不结束，是执行不到这里的
```

- 这个程序将会无限次循环。由于循环条件后面多了一个分号，因此循环体为空语句，i++自增并不是循环的一部分。

3.5.1 while语句

- ▶ (4) 由于while语句先计算表达式的值，再判断是否循环，所以如果表达式的值始终为假，则循环一次也不执行，失去了循环的意义。
- ▶ (5) while语句后面的圆括号是语法规定必须有的，循环条件可以是C语言的任意表达式。但由于结果是按逻辑值来处理的，通常情况下，循环条件是关系表达式或逻辑表达式，应该谨慎出现别的表达式。

3.5.1 while语句

- ▶ (6) 从循环结构来看，while语句前应有循环初始，循环体内应有循环控制。

CP 程序设计