

# 06 - 循环结构：for 循环

## C++ 程序设计基础

SOJ 信息学竞赛教练组

2024 年 7 月 20 日

# 目录

1 复习回顾

2 for 循环

3 for 循环输出

4 for 循环应用

5 循环结构综合归纳

6 总结

# while 循环

```
1 while (条件语句) {  
2     // 循环体  
3     每次执行的语句块;  
4 }
```

- 不断依照条件语句判定是否继续执行循环

# while 循环

```
1 while (条件语句) {  
2     // 循环体  
3     每次执行的语句块;  
4 }
```

- 不断依照条件语句判定是否继续执行循环
  - 条件成立 (true) 则执行循环体
  - 条件不成立 (false) 则结束循环

# while 循环

```
1 while (条件语句) {  
2     // 循环体  
3     每次执行的语句块;  
4 }
```

- 不断依照条件语句判定是否继续执行循环
  - 条件成立 (true) 则执行循环体
  - 条件不成立 (false) 则结束循环
- 当循环体中只有一个语句时, 可以省略花括号

# do..while 循环

```
1 do {  
2     // 循环体  
3     每次执行的语句块;  
4 } while (条件语句);
```

- 先执行一次循环体，再依照条件语句判定是否继续执行循环

# do..while 循环

```
1 do {  
2     // 循环体  
3     每次执行的语句块;  
4 } while (条件语句);
```

- 先执行一次循环体，再依照条件语句判定是否继续执行循环
  - 条件成立 (true) 则执行循环体
  - 条件不成立 (false) 则结束循环

# do..while 循环

```
1 do {  
2     // 循环体  
3     每次执行的语句块;  
4 } while (条件语句);
```

- 先执行一次循环体，再依照条件语句判定是否继续执行循环
  - 条件成立 (true) 则执行循环体
  - 条件不成立 (false) 则结束循环
- do...while 循环的小括号后必须要有一个分号



# do..while 循环

```
1 do {  
2     // 循环体  
3     每次执行的语句块;  
4 } while (条件语句);
```

- 先执行一次循环体，再依照条件语句判定是否继续执行循环
  - 条件成立 (true) 则执行循环体
  - 条件不成立 (false) 则结束循环
- do...while 循环的小括号后必须要有一个分号
- 当循环体中只有一个语句时，可以省略花括号

# while 和 do..while 的区别

- while 循环：先判断再循环
- do..while 循环：先执行再判断，至少执行一次

# 问题回顾：输出从 $n$ 到 1 之间的整数

## 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出从  $n$  到 1 之间的整数, 每个数字单独占一行。
- 样例输入  
3
- 样例输出  
3  
2  
1

# 问题回顾：输出从 n 到 1 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     while (n > 0) {
9         cout << n << endl; // 输出 n 的值
10        n--;                // n 的值减少 1
11    }
12
13    return 0;
14 }
```

# 问题展开：输出从 1 到 $n$ 之间的整数

## 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出从 1 到  $n$  之间的整数, 每个数字单独占一行。
- 样例输入  
3
- 样例输出  
1  
2  
3

# 问题展开：输出从 1 到 n 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int i = 1;
9     while (i <= n) {
10         cout << i << endl;
11         i++;
12     }
13
14     return 0;
15 }
```

# 问题展开：输出从 1 到 n 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int i = 1;
9     while (i <= n) {
10         cout << i << endl;
11         i++;
12     }
13
14     return 0;
15 }
```

- 倒序计数
  - 从  $n$  开始倒序计数，直到 1 进行最后一次循环
  - 循环条件：  $n > 0$
  - 循环体中对循环变量的修改：  $n--$



# 循环计数

- 倒序计数
  - 从  $n$  开始倒序计数，直到 1 进行最后一次循环
  - 循环条件：  $n > 0$
  - 循环体中对循环变量的修改：  $n--$
- 正序计数
  - 从 1 开始正序计数，直到  $n$  进行最后一次循环
  - 循环条件：  $i \leq n$
  - 循环体中对循环变量的修改：  $i++$

# 目录

1 复习回顾

2 for 循环

3 for 循环输出

4 for 循环应用

5 循环结构综合归纳

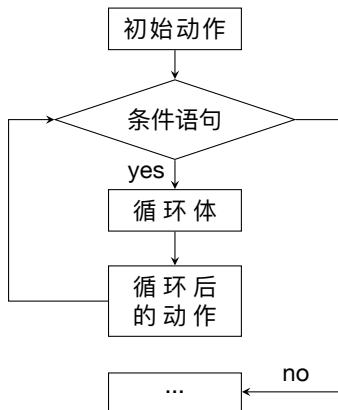
6 总结

# for 循环

- for 循环：通过循环变量控制循环的进程

# for 循环

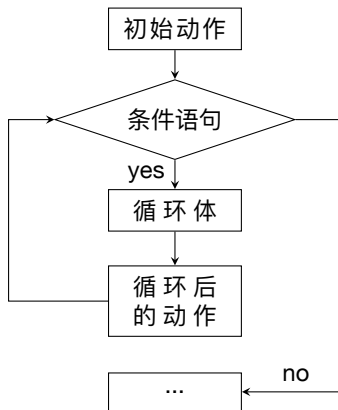
- for 循环：通过循环变量控制循环的进程



# for 循环

- for 循环：通过循环变量控制循环的进程

```
1 for (初始动作； 条件语句； 每次循环后的动作) {  
2   // 每次循环执行的语句块  
3   循环体；  
4 }
```



# for 循环

- for 循环：通过循环变量控制循环的进程

```
1 for (初始动作； 条件语句； 每次循环后的动作) {  
2     // 每次循环执行的语句块  
3     循环体；  
4 }
```

- 小括号中通过两个分号，隔开三种语句（可以留空）

# for 循环

- for 循环：通过循环变量控制循环的进程

```
1 for (初始动作； 条件语句； 每次循环后的动作) {  
2     // 每次循环执行的语句块  
3     循环体；  
4 }
```

- 小括号中通过两个分号，隔开三种语句（可以留空）
  - 初始动作只执行一次

# for 循环

- for 循环：通过循环变量控制循环的进程

```
1 for (初始动作； 条件语句； 每次循环后的动作) {  
2     // 每次循环执行的语句块  
3     循环体；  
4 }
```

- 小括号中通过两个分号，隔开三种语句（可以留空）
  - 初始动作只执行一次
  - 条件语句为循环继续的条件（留空视作条件成立）
    - 条件成立 (true) 则执行循环体
    - 条件不成立 (false) 则结束循环



# for 循环

- for 循环：通过循环变量控制循环的进程

```
1 for (初始动作； 条件语句； 每次循环后的动作) {  
2     // 每次循环执行的语句块  
3     循环体；  
4 }
```

- 小括号中通过两个分号，隔开三种语句（可以留空）
  - 初始动作只执行一次
  - 条件语句为循环继续的条件（留空视作条件成立）
    - 条件成立（true）则执行循环体
    - 条件不成立（false）则结束循环
  - 每次循环后，必定执行循环后的动作

# for 循环

- for 循环：通过循环变量控制循环的进程

```
1 for (初始动作； 条件语句； 每次循环后的动作) {  
2     // 每次循环执行的语句块  
3     循环体；  
4 }
```

- 小括号中通过两个分号，隔开三种语句（可以留空）
  - 初始动作只执行一次
  - 条件语句为循环继续的条件（留空视作条件成立）
    - 条件成立（true）则执行循环体
    - 条件不成立（false）则结束循环
  - 每次循环后，必定执行循环后的动作
- 当循环体中只有一个语句时，可以省略花括号

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量  
i = 1

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量  
i = 1

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量  
i = 1
- 输出  
1

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量  
 $i = 2$
- 输出  
1

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量  
i = 2
- 输出  
1



# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量  
i = 2
- 输出  
1  
2

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量

$i = 3$

- 输出

1

2

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量  
i = 3
- 输出  
1  
2

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量  
i = 3
- 输出  
1  
2  
3

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量

*i* = 4

- 输出

1

2

3

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量  
 $i \leq 3$  不成立
- 输出  
1  
2  
3

# 示例：for 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     for (int i = 1; i <= 3; i++) {
7         cout << i << endl;
8     }
9
10    return 0;
11 }
```

- 变量  
i = 4
- 输出  
1  
2  
3

## 例 6.1：输出从 1 到 $n$ 之间的整数

### 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出从 1 到  $n$  之间的整数, 每个数字单独占一行。
- 样例输入  
3
- 样例输出  
1  
2  
3



## 例 6.1：输出从 1 到 n 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     for (int i = 1; i <= n; i++) {
9         cout << i << endl;
10    }
11
12    return 0;
13 }
```

# 目录

1 复习回顾

2 for 循环

**3 for 循环输出**

4 for 循环应用

5 循环结构综合归纳

6 总结

## 例 6.2：输出从 0 到 $n - 1$ 之间的整数

### 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 输出从 0 到  $n - 1$  之间的整数, 每个数字单独占一行。
- 样例输入  
3
- 样例输出  
0  
1  
2

## 例 6.2：输出从 0 到 $n - 1$ 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     for (int i = 0; i <= n - 1; i++) {
9         cout << i << endl;
10    }
11
12    return 0;
13 }
```

## 例 6.2：输出从 0 到 $n - 1$ 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     for (int i = 0; i <= n - 1; i++) {
9         cout << i << endl;
10    }
11
12    return 0;
13 }
```

## 例 6.2：输出从 0 到 $n - 1$ 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     for (int i = 0; i < n; i++) {
9         cout << i << endl;
10    }
11
12    return 0;
13 }
```

## 例 6.2：输出从 0 到 $n - 1$ 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     for (int i = 0; i < n; i++) {
9         cout << i << endl;
10    }
11
12    return 0;
13 }
```

## 例 6.3：输出从 $L$ 到 $R$ 之间的整数

### 编程题

- 输入两个整数  $L$  和  $R$  ( $1 \leq L < R \leq 1000$ ), 输出从  $L$  到  $R$  之间的整数, 每个数字之间用空格间隔。
- 样例输入  
3 7
- 样例输出  
3 4 5 6 7



## 例 6.3：输出从 L 到 R 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int l, r;
7     cin >> l >> r;
8     for (int i = l; i <= r; i++) {
9         cout << i << " ";
10    }
11    cout << endl;
12
13    return 0;
14 }
```

## 例 6.3：输出从 L 到 R 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int l, r;
7     cin >> l >> r;
8     for (int i = l; i <= r; i++) {
9         cout << i << " ";
10    }
11    cout << endl;
12
13    return 0;
14 }
```

## 例 6.4：输出从 1 到 $n$ 之间的奇数

### 编程题

- 输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 保证  $n$  为奇数, 输出从 1 到  $n$  之间的奇数, 每个数字之间用空格间隔。
- 样例输入  
7
- 样例输出  
1 3 5 7

## 例 6.4：输出从 1 到 n 之间的奇数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     for (int i = 1; i <= n; i += 2) {
9         cout << i << " ";
10    }
11    cout << endl;
12
13    return 0;
14 }
```

## 例 6.4：输出从 1 到 n 之间的奇数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     for (int i = 1; i <= n; i += 2) {
9         cout << i << " ";
10    }
11    cout << endl;
12
13    return 0;
14 }
```

## 填空题

### 1. 阅读程序写结果

```
1 int main() {  
2     int n;  
3     cin >> n;  
4     for (int i = n; i >= 1; i--) {  
5         cout << i << " ";  
6     }  
7     cout << endl;  
8  
9     return 0;  
10 }
```

输入: 5

输出:

## 填空题

### 1. 阅读程序写结果

```
1 int main() {  
2     int n;  
3     cin >> n;  
4     for (int i = n; i >= 1; i--) {  
5         cout << i << " ";  
6     }  
7     cout << endl;  
8  
9     return 0;  
10 }
```

输入: 5

输出: 5 4 3 2 1

## 填空题

### 2. 阅读程序写结果

```
1 int main() {  
2     int l, r;  
3     cin >> l >> r;  
4     for (int i = r; i >= l; i--) {  
5         cout << i << " ";  
6     }  
7     cout << endl;  
8  
9     return 0;  
10 }
```

输入: 15 20

输出:



## 填空题

### 2. 阅读程序写结果

```
1 int main() {  
2     int l, r;  
3     cin >> l >> r;  
4     for (int i = r; i >= l; i--) {  
5         cout << i << " ";  
6     }  
7     cout << endl;  
8  
9     return 0;  
10 }
```

输入: 15 20

输出: 20 19 18 17 16 15

# 目录

1 复习回顾

2 for 循环

3 for 循环输出

**4 for 循环应用**

5 循环结构综合归纳

6 总结

# 示例：输出从 1 到 n 的整数之和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int sum = 0;
9     for (int i = 1; i <= n; i++) {
10         sum += i;
11     }
12     cout << sum << endl;
13
14     return 0;
15 }
```

# 示例：输出从 1 到 n 的整数之和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int sum = 0;
9     for (int i = 1; i <= n; i++) {
10         sum += i;
11     }
12     cout << sum << endl;
13
14     return 0;
15 }
```

## 例 6.5：分数数列求和

### 编程题

- 已知  $S_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$ 。  
输入整数  $n$ ，输出  $S_n$ ，结果保留两位小数。
- 样例输入  
3
- 样例输出  
1.83

## 例 6.5：分数数列求和

```
1 #include <iostream>
2 #include <iomanip>
3
4 using namespace std;
5
6 int main() {
7     int n;
8     cin >> n;
9     double sum = 0;
10    for (int i = 1; i <= n; i++) {
11        sum += 1.0 / i;
12    }
13    cout << fixed << setprecision(2) << sum << endl;
14
15    return 0;
16 }
```

## 例 6.5：分数数列求和

```
1 #include <iostream>
2 #include <iomanip>
3
4 using namespace std;
5
6 int main() {
7     int n;
8     cin >> n;
9     double sum = 0;
10    for (int i = 1; i <= n; i++) {
11        sum += 1.0 / i;
12    }
13    cout << fixed << setprecision(2) << sum << endl;
14
15    return 0;
16 }
```

## 例 6.6: 对输入的 $n$ 个整数求和

### 编程题

- 第一行输入一个整数  $n$  ( $1 \leq n \leq 1000$ ), 第二行输入  $n$  个整数  $x$  ( $-10^6 \leq x \leq 10^6$ ), 输出  $x$  的和。
- 样例输入  
3  
2 5 8
- 样例输出  
15



## 例 6.6：对输入的 n 个整数求和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int sum = 0, x;
9     for (int i = 1; i <= n; i++) {
10         cin >> x;
11         sum += x;
12     }
13     cout << sum << endl;
14
15     return 0;
16 }
```

## 例 6.6：对输入的 n 个整数求和

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int sum = 0, x;
9     for (int i = 1; i <= n; i++) {
10         cin >> x;
11         sum += x;
12     }
13     cout << sum << endl;
14
15     return 0;
16 }
```

## 例 6.7: 对输入的 $n$ 个整数求乘积

### 编程题

- 第一行输入一个整数  $n$  ( $1 \leq n \leq 10$ ), 第二行输入  $n$  个整数  $x$  ( $0 \leq x \leq 20$ ), 输出  $x$  的乘积。
- 样例输入  
3  
2 5 8
- 样例输出  
80

## 例 6.7：对输入的 n 个整数求乘积

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     long long mul = 1, x;
9     for (int i = 1; i <= n; i++) {
10         cin >> x;
11         mul *= x;
12     }
13     cout << mul << endl;
14
15     return 0;
16 }
```

## 例 6.7：对输入的 n 个整数求乘积

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     long long mul = 1, x;
9     for (int i = 1; i <= n; i++) {
10         cin >> x;
11         mul *= x;
12     }
13     cout << mul << endl;
14
15     return 0;
16 }
```

# 目录

1 复习回顾

2 for 循环

3 for 循环输出

4 for 循环应用

5 循环结构综合归纳

6 总结

# 不同循环结构的对比

- while 循环：先判断再执行

```
1 while (条件语句) {  
2     // 循环体  
3     每次执行的语句块;  
4 }
```

# 不同循环结构的对比

- while 循环：先判断再执行
- do...while 循环：先执行再判断，至少执行一次

```
1 do {  
2     // 循环体  
3     每次执行的语句块;  
4 } while (条件语句);
```



# 不同循环结构的对比

- while 循环：先判断再执行
- do...while 循环：先执行再判断，至少执行一次
- for 循环：执行初始动作后，先判断再执行

```
1 for (初始动作； 条件语句； 每次循环后的动作) {  
2     // 每次循环执行的语句块  
3     循环体；  
4 }
```

# 不同循环结构的适用场景

- 循环次数未知
  - while 循环
  - do...while 循环：至少执行一次

# 不同循环结构的适用场景

- 循环次数未知
  - while 循环
  - do...while 循环：至少执行一次
- 循环次数已知
  - for 循环

# 示例：输出从 L 到 R 之间的整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int l, r;
7     cin >> l >> r;
8     int i = l;
9     while (i <= r) {
10         cout << i << " ";
11         i++;
12     }
13     cout << endl;
14
15     return 0;
16 }
```

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int l, r;
7     cin >> l >> r;
8     for (int i = l; i <= r; i++) {
9         cout << i << " ";
10     }
11     cout << endl;
12
13     return 0;
14 }
```

# 目录

1 复习回顾

2 for 循环

3 for 循环输出

4 for 循环应用

5 循环结构综合归纳

**6 总结**

- for 循环的语法语义
- for 循环的应用
  - 循环输出
  - 循环累加
  - 循环累乘
- 循环结构综合归纳

# Thank you!