

05 - 循环结构：while 循环

C++ 程序设计基础

SOJ 信息学竞赛教练组

2024 年 6 月 5 日

目录

1 复习回顾

2 while 循环

3 while 循环的应用

4 do..while 循环

5 总结

顺序结构与分支结构

- 顺序结构：程序按照从上到下的顺序逐条执行各个语句

顺序结构与分支结构

- 顺序结构：程序按照从上到下的顺序逐条执行各个语句
- 分支结构：可以根据不同情况，执行不同的语句

顺序结构与分支结构的局限性

- 只能处理小规模的问题

顺序结构与分支结构的局限性

- 只能处理小规模的问题
 - 输出一行 "Hello world", 如果要求输出 100 行呢?

顺序结构与分支结构的局限性

- 只能处理小规模的问题
 - 输出一行 "Hello world", 如果要求输出 100 行呢?
 - 判断年份是否是闰年, 如果有 1000 个年份要判断呢?

循环结构

- 循环结构：能够反复执行一段指令

循环结构

- 循环结构：能够**反复**执行一段指令
- 循环结构的主要语法
 - while 循环
 - do...while 循环
 - for 循环

目录

1 复习回顾

2 while 循环

3 while 循环的应用

4 do..while 循环

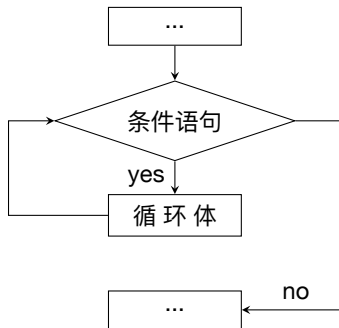
5 总结

while 循环

- while 循环：依照条件语句判定是否继续执行循环

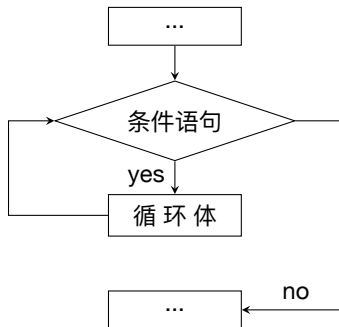
while 循环

- while 循环：依照条件语句判定是否继续执行循环



while 循环

- while 循环：依照条件语句判定是否继续执行循环



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

while 循环

- while 循环：依照条件语句判定是否继续执行循环
- 条件语句通常是布尔类型的表达式

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

while 循环

- while 循环：依照条件语句判定是否继续执行循环
- 条件语句通常是布尔类型的表达式
 - 条件成立 (true) 则执行循环体

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

while 循环

- while 循环：依照条件语句判定是否继续执行循环
- 条件语句通常是布尔类型的表达式
 - 条件成立 (true) 则执行循环体
 - 条件不成立 (false) 则结束循环

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


while 循环

- while 循环：依照条件语句判定是否继续执行循环
- 条件语句通常是布尔类型的表达式
 - 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

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

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中，n 的值都会减 1
10    }
11
12    return 0;
13 }
```

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中，n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 3

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中，n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 3

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中, n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 3
- 输出
hello

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中，n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 2
- 输出
hello

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中, n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 2
- 输出
hello

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中，n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 2
- 输出
hello
hello

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中，n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 1
- 输出
hello
hello

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中，n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 1
- 输出
hello
hello

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中, n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 1
- 输出
hello
hello
hello

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中，n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 0
- 输出
hello
hello
hello

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中, n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n > 0 不成立
- 输出
hello
hello
hello

示例：while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     while (n > 0) {
8         cout << "hello" << endl;
9         n--; // 在每次循环中, n 的值都会减 1
10    }
11
12    return 0;
13 }
```

- 变量
n = 0
- 输出
hello
hello
hello

例 5.1：输出 n 行 hello

编程题

- 编写程序，输入一个整数 n ($1 \leq n \leq 1000$), 输出 n 行 hello。
- 样例输入
2
- 样例输出
hello
hello

例 5.1：输出 n 行 hello

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     while (n > 0) {
9         cout << "hello" << endl;
10        n--; // 在每次循环中, n 的值都会减 1
11    }
12
13    return 0;
14 }
```


目录

- 1 复习回顾
- 2 while 循环
- 3 while 循环的应用**
- 4 do..while 循环
- 5 总结

例 5.2：输出从 n 到 1 之间的整数

编程题

- 编写程序，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

例 5.2：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 }
```

例 5.3：输出从 n 到 1 的整数之和

编程题

- 编写程序，输入一个整数 n ($1 \leq n \leq 1000$)，输出从 n 到 1 之间的整数之和。
- 样例输入
5
- 样例输出
15
- 样例说明
 $5 + 4 + 3 + 2 + 1 = 15$

例 5.3：输出从 n 到 1 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     while (n > 0) {
10         sum += n; // 相当于 sum = sum + n
11         n--;
12     }
13     cout << sum << endl;
14
15     return 0;
16 }
```

例 5.3：输出从 n 到 1 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     while (n > 0) {
10         sum += n; // 相当于 sum = sum + n
11         n--;
12     }
13     cout << sum << endl;
14
15     return 0;
16 }
```

例 5.4：对输入的 n 个整数求和

编程题

- 编写程序，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 ($1 \leq x \leq 10^6$)，输出 x 的和。
- 样例输入
3
2 5 8
- 样例输出
15

例 5.4：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     while (n > 0) {
10         cin >> x; // 输入 x
11         sum += x; // 相当于 sum = sum + x
12         n--;
13     }
14     cout << sum << endl;
15
16     return 0;
17 }
```


例 5.4：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     while (n > 0) {
10         cin >> x; // 输入 x
11         sum += x; // 相当于 sum = sum + x
12         n--;
13     }
14     cout << sum << endl;
15
16     return 0;
17 }
```

例 5.5：输出 n 天的学习时长

编程题

- 编写程序，输入一个整数 n ($1 \leq n \leq 1000$)，表示有 n 天。
接下来 n 行，每行输入两个整数 a, b ($0 \leq a, b \leq 6$)，分别表示每天校内的学习时长和校外的学习时长。
对于每一天，输出一行，表示这天的总学习时长。
- 样例输入
2
5 3
3 2
- 样例输出
8
5

例 5.5：输出 n 天的学习时长

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int a, b;
9     while (n > 0) {
10         cin >> a >> b;           // 输入校内学习时长 a，校外学习时长 b
11         cout << a + b << endl;   // 输出一天的总学习时长
12         n--;
13     }
14
15     return 0;
16 }
```

前面输入的数据的数量都是确定的，
如果输入的数据的数量不确定该怎么做呢？

例 5.6：求出若干个数的总和

编程题

- 编写程序，输入若干个整数 x ($1 \leq x \leq 1000$, 不超过 100 个), 当输入为 0 时结束，求输入的数的总和。
- 样例输入
5 1 3 4 0
- 样例输出
13
- 样例说明
 $5 + 1 + 3 + 4 = 13$

例 5.6：求出若干个数的总和

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

例 5.6：求出若干个数的总和

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

目录

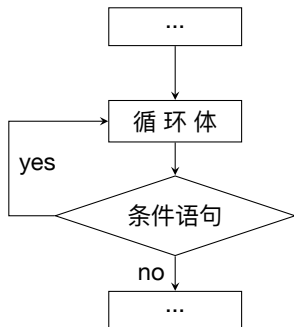
- 1 复习回顾
- 2 while 循环
- 3 while 循环的应用
- 4 do..while 循环**
- 5 总结

do..while 循环

- do..while 循环：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

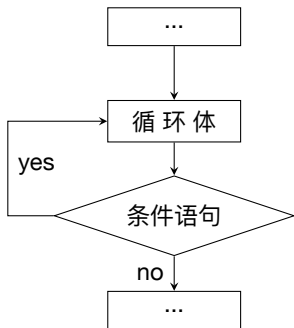
do..while 循环

- do..while 循环：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

- do..while 循环：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



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

do..while 循环

- do..while 循环：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
- 条件语句通常是布尔类型的表达式

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

do..while 循环

- do..while 循环：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
- 条件语句通常是布尔类型的表达式
 - 条件成立（true）则执行循环体

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

do..while 循环

- do..while 循环：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
- 条件语句通常是布尔类型的表达式
 - 条件成立（true）则执行循环体
 - 条件不成立（false）则结束循环

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

do..while 循环

- do..while 循环：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
- 条件语句通常是布尔类型的表达式
 - 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

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

do..while 循环

- do..while 循环：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
- 条件语句通常是布尔类型的表达式
 - 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
- 当语句块中只有一个语句时，可以省略花括号

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


示例：do..while 循环

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n = 3;
7     do {
8         cout << "hello" << endl;
9         n--; // 在每次循环中, n 的值都会减 1
10    } while (n > 0);
11
12    return 0;
13 }
```

例 5.7：判断非负整数的位数

编程题

- 编写程序，输入一个非负整数 n ($1 \leq n \leq 10^9$)，输出 n 的位数。
- 样例输入
16
- 样例输出
2

例 5.7：判断非负整数的位数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int cnt = 0; // 记录位数
9     while (n != 0) {
10         cnt++; // 位数加 1
11         n /= 10; // 相当于 n = n / 10, 去掉 n 的最后一位
12     }
13     cout << cnt << endl;
14
15     return 0;
16 }
```

例 5.7：判断非负整数的位数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int cnt = 0; // 记录位数
9     while (n != 0) {
10         cnt++; // 位数加 1
11         n /= 10; // 相当于 n = n / 10, 去掉 n 的最后一位
12     }
13     cout << cnt << endl;
14
15     return 0;
16 }
```

例 5.7：判断非负整数的位数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int cnt = 0; // 记录位数
9     while (n != 0) {
10         cnt++; // 位数加 1
11         n /= 10; // 相当于 n = n / 10, 去掉 n 的最后一位
12     }
13     cout << cnt << endl;
14
15     return 0;
16 }
```

- 如果输入为 0 呢？

例 5.7：判断非负整数的位数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int cnt = 0; // 记录位数
9     do {
10         cnt++;    // 位数加 1
11         n /= 10;  // 相当于 n = n / 10, 去掉 n 的最后一位
12     } while (n != 0);
13     cout << cnt << endl;
14
15     return 0;
16 }
```

例 5.7：判断非负整数的位数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     int cnt = 0; // 记录位数
9     do {
10         cnt++; // 位数加 1
11         n /= 10; // 相当于 n = n / 10, 去掉 n 的最后一位
12     } while (n != 0);
13     cout << cnt << endl;
14
15     return 0;
16 }
```

例 5.8：反向输出非负整数

编程题

- 编写程序，输入一个整数 n ($0 \leq n \leq 10^9$)，对 n 进行拆分数位并反向输出每一位数，数字之间以空格间隔。
- 样例输入
856
- 样例输出
6 5 8

例 5.8：反向输出非负整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     do {
9         cout << n % 10 << " "; // 输出最低位
10        n /= 10;                // 抹去最低位，其余数位右移一位
11    } while (n != 0);
12    cout << endl;
13
14    return 0;
15 }
```

例 5.8：反向输出非负整数

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     do {
9         cout << n % 10 << " "; // 输出最低位
10        n /= 10;                // 抹去最低位，其余数位右移一位
11    } while (n != 0);
12    cout << endl;
13
14    return 0;
15 }
```

while 和 do..while 的区别

- while 循环：先循环再判断
- do..while 循环：先执行再判断，至少执行一次

目录

- 1 复习回顾
- 2 while 循环
- 3 while 循环的应用
- 4 do..while 循环
- 5 总结**

总结

- while 循环
- do...while 循环
- 累加器 `sum` 和计数器 `cnt` 需要初始化为 0

Thank you!