

C++ 程序设计 I

徐东

xu.dong.sh@outlook.com

信息与计算科学

数学系

上海师范大学

2018 年 8 月 24 日

内容

- 1 循环语句
- 2 while 语句
- 3 改变循环流程的语句
- 4 for 语句
- 5 do - while 语句
- 6 调试

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

问题

- 输出 100 次 *Hello World*
- 计算 $\sum_{i=1}^{100} i$

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

问题

- 输出 100 次 *Hello World*
- 计算 $\sum_{i=1}^{100} i$
- 重复执行类似的任务

问题

- 输出 100 次 *Hello World*
- 计算 $\sum_{i=1}^{100} i$
- 重复执行类似的任务
- 循环结构
 - 控制语句块重复执行的一种结构

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

循环语句

- 程序设计中的三种基本结构
 - 顺序
 - 选择
 - 循环（迭代）

循环语句

- 程序设计中的三种基本结构
 - 顺序
 - 选择
 - 循环（迭代）
- 循环语句

循环语句

- 程序设计中的三种基本结构
 - 顺序
 - 选择
 - 循环（迭代）
- 循环语句
 - *while* 语句
 - *for* 语句
 - *do - while* 语句

循环语句

- 程序设计中的三种基本结构
 - 顺序
 - 选择
 - 循环（迭代）
- 循环语句
 - *while* 语句
 - *for* 语句
 - *do - while* 语句
- 通过合理、合规地组织这三种语句结构，（几乎）可以完成任何任务。

while 语句

- *while* 语句

```
while(循环条件){
```

当循环条件成立的时候，需要重复执行的（若干）语句；

```
}
```

while 语句

- while 语句

while(循环条件){

当循环条件成立的时候，需要重复执行的（若干）语句；

}

- while 关键字

while 语句

- while 语句

```
while(循环条件){
```

当循环条件成立的时候，需要重复执行的（若干）语句；

```
}
```

- while

关键字

- 循环条件

条件表达式、逻辑表达式

while 语句

- while 语句

```
while(循环条件){
```

当循环条件成立的时候，需要重复执行的（若干）语句；

```
}
```

- while 关键字
- 循环条件 条件表达式、逻辑表达式
- 循环条件放置在小括号内

while 语句

- while 语句

```
while(循环条件){
```

当循环条件成立的时候，需要重复执行的（若干）语句；

```
}
```

- while 关键字
- 循环条件 条件表达式、逻辑表达式
- 循环条件放置在小括号内
- {} 循环体（隔离外部代码）

输出 100 次 *Hello World*

```
1    int number = 0; //计数器
2
3    while(number < 0){
4        cout << "Hello_World" << endl;
5        number = number + 1; // number++; number += 1;
6    }
7
8    cout << "number_=" << number << endl;
```

编写循环语句的步骤

- ① 确定需要重复的语句；
- ② 把需要重复的语句放入一个循环中；
- ③ 编写“循环继续条件”，并为控制循环（结束）添加适当的语句。

编写循环语句的步骤

- ① 确定需要重复的语句；
- ② 把需要重复的语句放入一个循环中；
- ③ 编写“循环继续条件”，并为控制循环（结束）添加适当的语句。

语句组；

编写循环语句的步骤

- ① 确定需要重复的语句；
- ② 把需要重复的语句放入一个循环中；
- ③ 编写“循环继续条件”，并为控制循环（结束）添加适当的语句。

```
while(true){  
    语句组;  
  
}
```

编写循环语句的步骤

- ① 确定需要重复的语句；
- ② 把需要重复的语句放入一个循环中；
- ③ 编写“循环继续条件”，并为控制循环（结束）添加适当的语句。

```
while(循环继续条件){  
    语句组;  
  
}
```

编写循环语句的步骤

- ① 确定需要重复的语句；
- ② 把需要重复的语句放入一个循环中；
- ③ 编写“循环继续条件”，并为控制循环（结束）添加适当的语句。

```
while(循环继续条件){  
    语句组;  
    用于控制循环的附加语句;  
}
```

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

while 循环

- 计算 $\sum_{i=1}^{100} i$

while 循环

- 计算 $\sum_{i=1}^{100} i$
 - 每次添加一个数字 i

while 循环

- 计算 $\sum_{i=1}^{100} i$
 - 每次添加一个数字 i
 - 使用这个数字 i 控制循环

while 循环

- 计算 $\sum_{i=1}^{100} i$
 - 每次添加一个数字 i
 - 使用这个数字 i 控制循环 ($i \leq 100$)

while 循环

- 计算 $\sum_{i=1}^{100} i$
 - 每次添加一个数字 i
 - 使用这个数字 i 控制循环 ($i \leq 100$)
- 按数位分解任意正整数

while 循环

- 计算 $\sum_{i=1}^{100} i$
 - 每次添加一个数字 i
 - 使用这个数字 i 控制循环 ($i \leq 100$)
- 按数位分解任意正整数
 - 234 分解为 2、3、4

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

编写循环时需要注意的两个问题

- 无限循环

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

编写循环时需要注意的两个问题

- 无限循环
 - 循环条件始终为 **true**

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

编写循环时需要注意的两个问题

- 无限循环

- 循环条件始终为 **true**

- Console *ctrl* + *c* 强制结束循环

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

编写循环时需要注意的两个问题

- 无限循环
 - 循环条件始终为 **true**
 - Console *ctrl + c* 强制结束循环
 - 保证循环条件最终可以变为 *false*

编写循环时需要注意的两个问题

- 无限循环
 - 循环条件始终为 **true**
 - Console *ctrl + c* 强制结束循环
 - 保证循环条件最终可以变为 *false*
- 差一错误 (*off-by-one error*)

编写循环时需要注意的两个问题

- 无限循环
 - 循环条件始终为 **true**
 - Console *ctrl + c* 强制结束循环
 - 保证循环条件最终可以变为 *false*
- 差一错误 (*off-by-one error*)
 - 循环多执行一次或少执行一次

编写循环时需要注意的两个问题

- 无限循环
 - 循环条件始终为 **true**
 - Console *ctrl + c* 强制结束循环
 - 保证循环条件最终可以变为 *false*
- 差一错误 (*off-by-one error*)
 - 循环多执行一次或少执行一次
 - **number < 100**

编写循环时需要注意的两个问题

- 无限循环

- 循环条件始终为 **true**
- Console *ctrl + c* 强制结束循环
- 保证循环条件最终可以变为 *false*

- 差一错误 (*off-by-one error*)

- 循环多执行一次或少执行一次
- $\text{number} < 100 \rightsquigarrow \text{number} \leq 100$

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

改变循环的流程

- C++ 对循环提供了两种额外的（流程）控制
 - *break* 语句
 - *continue* 语句

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

改变循环的流程

- C++ 对循环提供了两种额外的（流程）控制
 - *break* 语句
 - *continue* 语句
- *break* 结束整个循环（提前跳出整个循环）

改变循环的流程

- C++ 对循环提供了两种额外的（流程）控制
 - *break* 语句
 - *continue* 语句
- *break* 结束整个循环（提前跳出整个循环）
- *continue* 结束当前迭代（提前进入下一轮迭代）

改变循环的流程

- C++ 对循环提供了两种额外的（流程）控制
 - *break* 语句
 - *continue* 语句
- *break* 结束整个循环（提前跳出整个循环）
- *continue* 结束当前迭代（提前进入下一轮迭代）
- *continue* 可能导致无限循环（跳过循环体内的后续代码）

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

问题

- 对任意一组数字求和

-

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

问题

- 对任意一组数字求和

- 提示

问题

- 对任意一组数字求和 -
- 提示
 - 只计算正整数（简化问题）

问题

- 对任意一组数字求和
- 提示
 - 只计算正整数（简化问题）
 - 把 0 或者 负数 设置为
”结束标志”

问题

- 对任意一组数字求和

- 提示

- 只计算正整数（简化问题）
- 把 0 或者 负数 设置为

”结束标志”

-

```
int total = 0, x = 0;    1
while(true){            2
    cin >> x;            3
    if(x == 0) break;    4
    total += x;          5
}                        6
```

问题

- 对任意一组数字求和

- 提示

- 只计算正整数（简化问题）
- 把 0 或者 负数 设置为

“结束标志”

- “结束标志”可能是有意义的数

```
int total = 0, x = 0; 1
while(true){          2
    cin >> x;          3
    if(x == 0) break;  4
    total += x;        5
}                      6
```

对任意一组数字求和

```
int total = 0, x = 0;
while(cin >> x){
    total += x;
}
```

- 按 *CTRL + D* 或 *CTRL + Z* 结束输入 (循环停止)

对任意一组数字求和

```
int total = 0, x = 0;
while(cin >> x){
    total += x;
}
```

- 按 *CTRL + D* 或 *CTRL + Z* 结束输入 (循环停止)

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

问题

- 对任意一组数字求和
- 修改
 - 不计入数字 6

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

计算 $\sum_{i=1}^{100} i$

```
1      int total = 0, i = 1;
2      while(i <= 100){
3          total = total + i;
4          i = i + 1; // ++i
5      }
```


循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

计算 $\sum_{i=1}^{100} i$

```
1    int total = 0, i = 1;
2    while(i <= 100){
3        total = total + i;
4        i = i + 1; // ++i
5    }
```

● 更习惯的写法 (for 语句)

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

计算 $\sum_{i=1}^{100} i$

```
1    int total = 0, i = 1;
2    while(i <= 100){
3        total = total + i;
4        i = i + 1; // ++i
5    }
```

● 更习惯的写法 (for 语句)

```
1    int total = 0;
2    for(int i = 1; i <= 100; ++i){
3        total = total + i;
4    }
```

for 循环

- for 语句

```
for(循环控制变量初始化; 循环条件; 修改控制变量){  
    需要重复执行的若干语句;  
}
```

- for 关键字

- 不能省略小括号 for(;;) 内的两个分号 (;)

- for 语句适合循环次数已知的循环

for 循环

- for 语句

```
for(exp_1; exp_2; exp_3){  
    需要重复执行的若干语句;  
}
```

- exp1 只被执行一次
- exp2 决定是否进行循环
- exp3 在循环体内的语句执行完毕之后再执行（随后进入新一轮的迭代）

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

for 语句

```
int main(){  
    double total = 0.0, a = 0.0;  
    for(int i = 0 ; i < 3 ; ++i){  
        cin >> a;  
        total += a;  
    }  
    cout << "total = " << total << endl;  
    return 0;  
}
```

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

for 语句

```
int main(){  
    double total = 0.0, a = 0.0;  
    for(int i = 0 ; i < 3 ; ++i){  
        cin >> a;  
        total += a;  
    }  
    cout << "total = " << total << endl;  
    return 0;  
}
```

for 语句

```
int main(){  
    double total = 0.0, a = 0.0;  
    for(int i = 0 ; i < 3 ; ++i){  
        cin >> a;  
        total += a;  
    }  
    cout << "total = " << total << endl;  
    return 0;  
}
```

- 变量 `i` 只在 `for` 循环体内有效

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

for 语句

```
int main(){  
    double total = 0.0, a = 0.0;  
    for(int i = 0 ; i < 3 ; ++i){  
        cin >> a;  
        total += a;  
    }  
    cout << "total = " << total << endl;  
    return 0;  
}
```

- 变量 i 只在 for 循环体内有效 (局部变量)
- 变量的作用域 (变量的有效范围)

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

问题

- 输出“九九乘法表”

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

问题

- 输出“九九乘法表”
- 提示

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

问题

- 输出“九九乘法表”
- 提示
 - 嵌套的循环结构

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

问题

- 输出“九九乘法表”
- 提示
 - 嵌套的循环结构
 - for 循环 循环次数已知

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

问题

- 输出“九九乘法表”
- 提示
 - 嵌套的循环结构
 - for 循环 循环次数已知
 - while 循环 循环次数未知

问题

- 输出“九九乘法表”
- 提示
 - 嵌套的循环结构
 - for 循环 循环次数已知
 - while 循环 循环次数未知
 - 格式化输出（题外话）

do - while 循环

- do - while 语句

do{

 需要重复执行的若干语句;

}while(循环条件);

do - while 循环

- do - while 语句

do{

需要重复执行的若干语句;

}while(循环条件);

- 注意

- 结尾处的分号 (;)
- 关键字 do 和 while 之间的大括号 ({})

循环语句

- 前测循环（先判断后执行）
 - *while* 循环
 - *for* 循环
 - 循环最少迭代次数 0
- 后测循环（先执行后判断）
 - *do - while* 循环
 - 循环最少迭代次数 1
- 三种循环语句相互等价

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

调试的方法

- 添加 `cout` 语句
- debug

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

调试的方法

- 添加 `cout` 语句
- `debug`(只针对 CodeBlocks 而言)

循环语句
while 语句
改变循环流程的语句
for 语句
do - while 语句
调试

调试的方法

- 添加 `cout` 语句
- `debug`(只针对 CodeBlocks 而言)
 - 项目状态
 - 设置断点
 - `debug` 状态
 - 逐步调试
 - `watch` 窗口

Q & A