

C++ 循环语句

一、循环的作用

在程序中重复执行特定代码块，直到满足终止条件

二、for循环

1. 语法结构

```
1  for (初始化表达式; 条件表达式; 更新表达式) {  
2      // 循环体  
3  }
```

2. 执行流程

1. 执行初始化表达式（仅一次）
2. 检查条件表达式
 - 为真：执行循环体 → 执行更新表达式 → 回到步骤2
 - 为假：退出循环

3. 适用场景

已知循环次数的情况

4. 代码示例

```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main() {  
5      // 打印1-10的平方  
6      for (int i = 1; i <= 10; i++) {  
7          cout << i << "^2 = " << i*i << endl;  
8      }  
9  
10     // 嵌套循环示例（九九乘法表）  
11     for (int i = 1; i <= 9; i++) {  
12         for (int j = 1; j <= i; j++) {  
13             cout << j << "x" << i << "=" << i*j << "\t";  
14         }  
15         cout << endl;  
16     }  
17     return 0;  
18 }
```

三、while循环

1. 语法结构

```
1 while (条件表达式) {  
2     // 循环体  
3     // 需要包含改变条件的语句  
4 }
```

2. 执行流程

1. 检查条件表达式

- 为真：执行循环体 → 回到步骤1
- 为假：退出循环

3. 适用场景

循环次数不确定，需要根据条件判断的情况

4. 代码示例

```
1  #include <iostream>  
2  using namespace std;  
3  
4  int main() {  
5      // 猜数字游戏  
6      int target = 42;  
7      int guess;  
8  
9      cout << "请输入猜测的数字 (1-100) : ";  
10     cin >> guess;  
11  
12     while (guess != target) {  
13         if (guess > target) {  
14             cout << "太大了! ";  
15         } else {  
16             cout << "太小了! ";  
17         }  
18         cout << "请重新输入: ";  
19         cin >> guess;  
20     }  
21  
22     cout << "恭喜你猜对了! " << endl;  
23     return 0;  
24 }
```

四、do-while循环

1. 语法结构

```
1 do {  
2     // 循环体  
3 } while (条件表达式);
```

2. 执行流程

1. 先执行循环体
2. 检查条件表达式
 - 为真：回到步骤1
 - 为假：退出循环

3. 适用场景

需要至少执行一次循环体的情况

4. 代码示例

```
1 #include <iostream>  
2 using namespace std;  
3  
4 int main() {  
5     // 菜单系统  
6     int choice;  
7  
8     do {  
9         cout << "\n==== 菜单 =====" << endl;  
10        cout << "1. 开始游戏" << endl;  
11        cout << "2. 加载存档" << endl;  
12        cout << "3. 设置" << endl;  
13        cout << "0. 退出" << endl;  
14        cout << "请输入选项: ";  
15        cin >> choice;  
16  
17        // 处理用户选择  
18        switch(choice) {  
19            case 1: /* 游戏逻辑 */ break;  
20            case 2: /* 加载逻辑 */ break;  
21            case 3: /* 设置逻辑 */ break;  
22            case 0: cout << "再见! " << endl; break;  
23            default: cout << "无效输入! " << endl;  
24        }  
25    } while (choice != 0);  
26  
27    return 0;  
28 }
```

五、循环控制语句

- `break`：立即终止当前循环
- `continue`：跳过当前迭代，进入下一次循环
- 注意事项：
 - 谨慎使用可能造成无限循环
 - 确保循环条件最终会变为假

六、循环选择指南

| 循环类型 | 适用场景 | 特点 |
|----------|----------|----------|
| for | 已知循环次数 | 结构紧凑 |
| while | 条件控制循环 | 可能一次都不执行 |
| do-while | 必须至少执行一次 | 先执行后判断 |

七、常见问题

- 无限循环的两种形式：

```
1  for (;;) { /* ... */ } // 合法但危险
2  while (true) { /* ... */ }
```

- 循环变量作用域：
 - for循环的初始化变量只在循环内有效（C++11及以后）
 - while循环需要在外部声明变量
- 效率考虑：
 - 避免在循环条件中进行复杂计算
 - 提前计算好循环不变量

这份讲稿包含了理论说明、代码示例和实用建议，可根据听众水平调整讲解深度。建议配合实际编程演示和练习题目进行教学。