

C++程序设计基础 第三章

前言

第3章 语句控制结构

3.1 语句

3.1.1 空语句

3.1.2 复合语句

3.2 分支结构

3.2.1 if语句

1. 嵌套的if语句

1. 避免悬垂else

3.2.2 switch语句

3.3 循环结构

3.3.1 while语句

3.3.2 do while 语句

3.3.3 for语句

3.4 跳转语句

3.4.1 break语句

3.4.2 continue语句

前言

本文档由 @ItsJiale 创作，作者博客：<https://jjiale.domcer.com/>，作者依据数学与大数据学院2024级大数据教学班的授课重点倾向编写而成。所有内容均为手动逐字录入，其中加上了不少自己的理解与思考，耗费近一周时间精心完成。

此文档旨在助力复习 C++ 程序设计基础，为后续学习数据结构筑牢根基。信计专业的同学，也可参考本文档规划复习内容。需注意，若个人学习过程中存在不同侧重点或对重难点的理解有差异，应以教材内容为准。倘若文档内容存在任何不妥之处，恳请各位读者批评指正。

By: ItsJiale

2025.4.8

第3章 语句控制结构

由于C++和Java在基础方面比较相近，因此在C++基础方面简略介绍，而在不同处重点介绍

3.1 语句

语句是程序执行的基本单元，C++的语句大多数以分号结束。一个表达式后面加上一个分号就变成了一个表达式语句。

3.1.1 空语句

只有一个分号构成的语句称为空语句，例如：

```
| ;
```

空语句不会执行任何操作，一般情况下不会影响程序的执行

3.1.2 复合语句

用花括号括起来的语句序列称为复合语句，复合语句构成一个语句块。它们的作用域从定义的地方开始到语句块结束处结束，例如：

```
{  
    int sum = 0;           //语句块开始  
    .....                //语句块结束  
}
```

3.2 分支结构

3.2.1 if语句

if语句的语法格式如下：

```
if (expr) {                //条件表达式  
    statement;            //语句
```

```
}
```

if语句也可以有else分支，格式如下：

```
if (expr) {  
    statement1;    //分支语句1  
}else {  
    statement2;    //分支语句2  
}
```

1.嵌套的if语句

如果有一个分支，则选用if语句；如果有两个分支，则选用if和else语句结构；如果有两个以上的分支，则选用嵌套的if语句结构。内嵌的if语句可以嵌套在if语句中，也可以嵌套在else语句中。

1. 避免悬垂else

C++规定else和离它最近的尚未匹配的if匹配

注：else语句数目不能多于if语句数目，这是因为else语句是可以省略的，而if是不能省略的。

3.2.2 switch语句

switch后跟整型表达式，case后跟整型常量表达式 **(都必须是整型)**

```
switch( 整型表达式 ){  
    case 整型常量表达式:  
        .....  
    break;  
}
```

如果遇到break语句，则会退出整个switch结构

- 若 `条件表达式` 的结果为 `false`，则整个三目运算符表达式的值是 `表达式2` 的值。

3.3.2 do while 语句

do while语句的语法格式为：

```
do {  
    statement;           //循环体语句  
}while( expr);         //条件表达式，注意以分号结束
```

do while 语句和while 语句非常相似，唯一的区别是：while语句先计算条件表达式，然后根据结果判断是否执行循环体；而do while 先执行循环体后计算条件表达式，**即先执行一次**

3.3.3 for语句

for语句的语法格式为：

```
for (expr1; expr2; expr3) {  
    statement;  
}
```

3.4 跳转语句

3.4.1 break语句

break语句只能用于switch语句或循环语句中，用来跳出离它最近的switch语句或终止循环的执行

3.4.2 continue语句

continue语句只在循环结构中起作用，用来终止当前操作，进入下一次循环，下一次循环是否被执行取决于循环条件是否成立