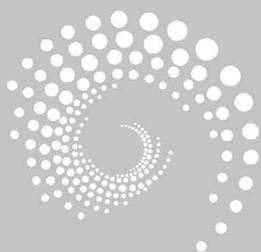


顺序结构

CHAPTER 3



顺序结构指的是程序按照代码的书写顺序,从上到下,依次执行语句的一种结构。顺序结构是程序中最基本的执行流程,没有使用任何控制流语句(如条件判断、循环等)来改变执行顺序。

3.1 程序语句

语句是构成程序的基本指令单元,它们定义了程序的行为和执行逻辑。程序就相当于由语句构成的一篇文章。C++中的语句包括如下类型。

(1) **声明语句**。C++中的声明语句用于指定变量等的唯一名字,一旦声明了一个名字,后面就可以使用了,不能未经声明就使用一个变量。一般来说,变量声明的位置应该尽可能靠近其使用的位置。比如之前声明整型变量*i,j,k*使用如下语句:

```
int i,j,k;
```

这类语句就是声明语句。

(2) **表达式语句**。表达式语句的语法形式如下。

[表达式];

表达式末尾加上分号就是表达式语句,最常用的表达式语句是赋值语句和函数调用语句。方括号表示可选项,也就是说如果没有表达式,只有一个分号,这也是合法的表达式语句,这种情况称之为空语句,相当于什么也没有做。空语句可以作为占位符,可以用在循环体中表示什么也不做的空语句。也可以用于辅助在复合语句或函数末尾放置标号。

(3) **控制语句**。控制语句可以改变程序的执行流程。具体来说,可以分为如下类型。

① 选择语句。

C++提供了选择执行代码的方法。*if*和*switch*关键字可以引导地选择执行的代码。

- *if*语句。按照条件真假选择执行路径。
- *switch*语句。按照条件是否满足多个分支中的一条选择执行路径。

② 循环语句。

循环语句可以将循环体中的语句或者复合语句根据某些循环终止条件执行0次或多次,当执行复合语句的时候,复合语句中的语句会按照语句本身的流程控制情况进行执行,如果不是选择语句、循环语句、跳转语句等几种转移流程的语句,那么复合语句中的语句将顺序执行。

- *while*语句。当某个条件成立时重复执行循环体。
- *do-while*语句。先执行循环体再判断是否继续循环。
- *for*语句。循环语句。
- 基于范围的*for*语句。逐个遍历某个范围(比如数组)的元素并执行循环体。

③ 跳转语句。

跳转语句包括:

- *break*语句。终止*switch*或终止循环语句。
- *continue*。用于在循环体内部继续下一次循环。
- *return*。从函数内部返回。
- *goto*。无条件跳转(不提倡使用)。

④ 标号语句。

标号语句主要有以下3种。

- 标识符语句。该类语句主要和 goto 语句搭配使用。
- case 语句。在 switch 语句中用于判断是否和某个条件相符。
- default 语句。在 switch 语句中用于不符合任何 case 分支条件时的默认分支。

(4) **复合语句**。复合语句就是用花括号{}将多条语句括起来的语句,语法形式如下:

```
{[语句列表]}
```

通常也将复合语句称为语句块,复合语句可以看成是一条实现了复合功能的单一语句。

3.2 三种执行流程

顺序结构、选择结构、循环结构是结构化程序设计的三种基本执行流程,通过这三种基本结构程序的有机构造,可以实现各类复杂的程序功能。

(1) **顺序结构**。程序中的语句按照它们在代码中的书写顺序从上到下,一条一条依次执行,不跳过,也不重复,如图 3.1 所示。

一般而言,只要语句不是控制语句,程序就会按照语句书写的先后顺序来执行。

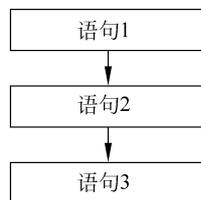


图 3.1 顺序结构流程

(2) **选择结构**。选择结构是程序中的语句根据某个条件来选择程序中的某些语句来执行。如图 3.2 所示。语句块为使用花括号括起来的一条或多条程序语句。选择结构包含单选(if)(见图 3.2(a)),二选一(if-else,三目运算符)(见图 3.2(b)),多选一(if-else 嵌套,switch)(见图 3.2(c))等。

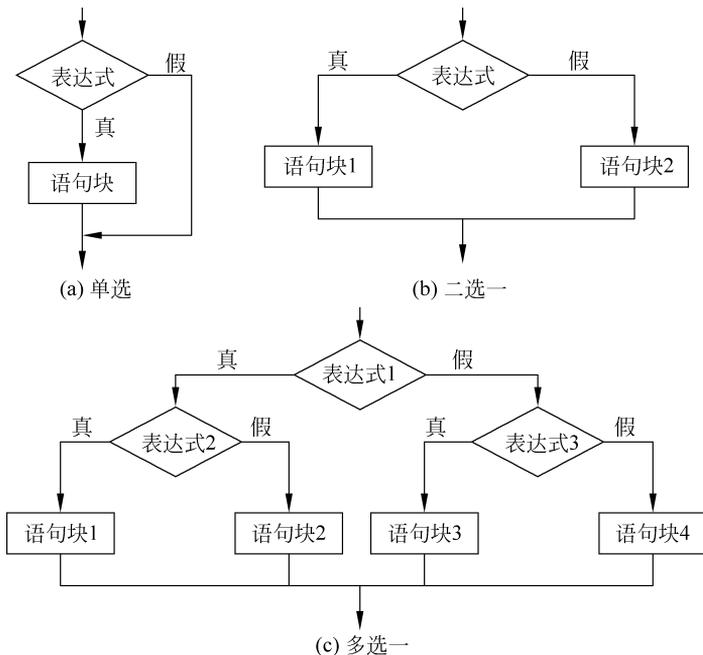


图 3.2 选择结构

在程序从上而下排列的书写布局中,看起来就是某些语句会被跳过而不执行。

(3) 循环结构。

循环结构是程序会根据条件是否满足来重复执行某些语句。循环结构可以分成当型(while, for)循环(见图 3.3(a))和直到型(do-while)循环(见图 3.3(b))结构两种。

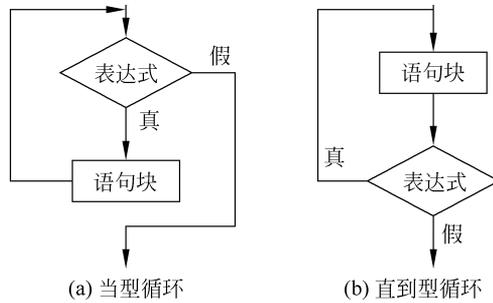


图 3.3 循环结构

3.3 顺序结构

程序代码在计算机中的执行遵循一定的流程,之前看到的所有程序均是顺序执行,也就是其中的程序语句是一条接着一一条执行的,中间不会跳过其他语句,也不会执行先前已经执行过的语句。例 3.1 计算 BMI 指数也是一个顺序结构的程序。

【例 3.1】 计算 BMI 指数。

思路：

体质指数 BMI=体重(千克)/身高(米)的平方,因此将计算过程分成 3 阶段:第 1 阶段准备好体重和身高数据;第 2 阶段按照公式计算 BMI;第 3 阶段输出 BMI。

程序代码：

```

01 #include <iostream>
02 #include <iomanip>
03 using namespace std;
04 int main()
05 {
06     //步骤 1:输入身高和体重
07     double weight,height;
08     cout<<"Please input the weight(kg):";
09     cin>>weight;
10     cout<<"Please input the height(m):";
11     cin>>height;
12     //步骤 2:计算 BMI
13     double bmi=weight/(height*height);
14     //步骤 3:输出 BMI
15     cout<<"The BMI:";
16     cout<< setiosflags(ios::fixed)<< setprecision(2)<< bmi<< endl;
17     return 0;
18 }
  
```

运行结果：

```
Please input the weight(kg):73 ✓
Please input the height(m):1.75 ✓
The BMI:23.84
```

代码分析:

代码第 6 行、第 12 行,第 14 行为单行注释,可以在程序设计时提前将注释写好,用这种方法将思路体现到代码中,方便自己编写代码。

代码第 7 行定义了 2 个双精度浮点数。

代码第 8 行输出提示信息“Please input the weight(kg):”,提示用户输入体重,输出见运行结果第 1 行前部。

代码第 9 行要求用户输入具体体重数据。

代码第 10 行输出提示信息“Please input the height(m):”,提示用户输入身高,输出见运行结果第 2 行前部。

代码第 11 行要求用户输入具体身高数据。

代码第 13 行计算 BMI 指数。

代码第 15 行输出提示信息“The BMI:”

代码第 16 行设置双精度浮点数输出为固定 2 位小数输出,输出见运行结果第 3 行。输出控制需要在程序头部包含<iomanip>头文件。

可以看出,程序代码是从第一行代码一句一句依次向下执行的,直到程序结束,这即是典型的顺序结构的程序。



视频讲解

🔑 3.4 应用

【例 3.2】 求圆面积。

思路:

圆的面积公式 $s = \pi r^2$, r 是圆的半径, π 是一个无理数常数,面积需要根据用户输入的半径来计算,因此程序总体上分为 3 个步骤:

步骤 1: 输入半径。

步骤 2: 计算面积。

步骤 3: 输出面积。

程序代码:

```
01 #include <iostream>
02 #include <iomanip>
03 using namespace std;
04 #define PI 3.14
05 int main()
06 {
07     double r;
08     cout << "Please input the radius of circle:";
09     cin >> r;
```

```

10     double area=PI * r * r;
11     cout <<"The area of the circle:";
12     cout << setiosflags(ios::fixed)<< setprecision(2)<< area << endl;
13     return 0;
14 }

```

运行结果：

```

Please input the radius of circle:3.5 ✓
The area of the circle:38.47

```

代码分析：

代码第 7~9 行是思路的步骤 1,实现输入半径的目的。见运行结果第 1 行。

代码第 10 行是思路的步骤 2,实现计算面积的目的。

代码第 11 行、第 12 行是算法的步骤 3,实现输出面积的目的。输出见运行结果第 2 行。

可以看出程序对应思路,并且整个执行流程是顺序结构。

【例 3.3】 小写字符转大写。**思路：**

系统输入一个小写字符后,直接根据其 ASCII 码对应转换到大写的 ASCII 码,然后输出大写的 ASCII 码字符。总体上也是 3 个步骤：

步骤 1: 输入小写字符。

步骤 2: 转换字符到大写字符。

步骤 3: 输出大写字符。

程序代码：

```

01 #include < cstdio >
02 using namespace std;
03 int main()
04 {
05     char c=getchar();
06     char dif='A'-'a';
07     c=c+dif;
08     putchar(c);
09     putchar('\n');
10     return 0;
11 }

```

运行结果：

```

y ✓
Y

```

代码分析：

代码第 5 行输入字符,这里并没有对用户输入的字符进行检测,而是假设用户输入正确的字符。输入字符的方法是调用 `getchar()` 函数(函数是预先定义好的功能模块,在使用的时候只要按照一定的规范调用就可以了),这是一个在 `cstdio` 头文件中声明的函数,所以第 1 行包含了该头文件。`getchar()` 函数可以输入键盘上输入的任何字符,甚至包括空格,而若

这样使用:

```
char c;
cin >> c;
```

则 cin 会自动跳过其中的空格,所以在某些特定的情况(非本问题)下需要读入全部输入字符时,不能使用 cin 来读取。这一步的执行见运行结果第 1 行。

代码第 6 行通过‘A’-‘a’计算大写字符和小写字符的差,虽然这个差是固定的,包括各个字符的 ASCII 码都是固定的,不过也没有必要记忆,只要知道原理,知道大写字符连续编码,小写字符连续编码,大写和小写字符之间间隔固定,然后在程序中可以用这个方法获得固定的间隔。

代码第 7 行计算获得大写字符。

代码第 8 行输出大写字符,第 9 行输出回车符,输出字符时使用了 putchar()函数,该函数也是在 cstdio 头文件中声明的,该函数的参数可以为字符也可以为整数,如果是整数,putchar()函数会将该整数当作 ASCII 码解释,最终输出对应的字符。这两步的输出见运行结果第 2 行。

【例 3.4】 三位数各位数字求和。

思路:

该问题的核心是如何获取三位数的各位数字。总体上也是 3 个步骤:

步骤 1: 输入三位数。

步骤 2: 获得各位数,并求和。

步骤 3: 输出和。

程序代码:

```
01 #include <iostream>
02 using namespace std;
03 int main()
04 {
05     int n,s(0);
06     cin >> n;
07     s+=n%10;
08     n/=10;
09     s+=n%10;
10     n/=10;
11     s+=n;
12     cout << s << endl;
13     return 0;
14 }
```

运行结果:

```
789 ↵
24
```

代码分析:

代码第 5 行定义了一个整型变量 n,准备用来接收用户的输入;另外定义了一个整型变量 s 并初始化为 0,目的是存储最终各位数字的和。

代码第6行用户输入数据 n 。见运行结果第1行。

代码第7行 $s += n \% 10$ 功能有两方面：首先对于 $n \% 10$ 可以获得该数的个位数，然后 $+=$ 的操作可以将该个位数加到 s 中。比如 n 为 789，那么这一步就获得了个位数 9，并将该个位数加入到 s 中。

代码第8行 $n /= 10$ 的目的是用 n 去除个位数以外的数更新 n ，比如 n 为 789，那么这一步 n 就被更新成为 78。仔细体会第7行和第8行的功能，相信下面依次的几行代码就不难理解了。

代码第9行和第7行代码相同，这一步将继续获得更新后的 n 的个位数，比如上一步得到的 n 为 78，那这一步获得个位数为 8，并将该数加入到 s 中。

代码第10行和第8行相同，获得除个位外的数，比如这一步开始执行时候的 n 为 78，那么执行完这一句代码后， n 将变为 7。对于最初的三位数，这一步执行完后已经得到了最高位了。

代码第11行直接将最后的一位数加到 s 中。

代码第12行输出 s ，输出见运行结果第2行。

3.5 本章小结

本章介绍了 C++ 中的程序语句、执行流程和顺序结构。程序语句是构成程序的基本单元，它们定义了程序的行为和执行逻辑。执行流程描述程序语句的执行顺序，合计有顺序结构、选择结构、循环结构三种流程，其中顺序结构是最简单的执行流程。顺序结构的逻辑简单直观，易于理解和编写。程序按照代码的物理顺序执行，不需要额外的控制结构。

在 C++ 中，大多数程序都以顺序结构作为基础，然后根据需要添加控制流语句。在学习本章的过程中，需要综合先前章节所学习的基础知识，通过将常量定义、变量定义及各类表达式以程序语句的形式表达出来，并将这些程序语句按先后关系有序地组织。编写顺序结构程序要求用户提前清晰、有条理地描述解决任务的具体过程，这就是所谓“谋定而后动”。

习题 3

1. 简述 C++ 中语句的概念。
2. 简述 C++ 中的执行流程。
3. 简述 C++ 中的顺序结构。
4. 写程序求一元二次方程 $ax^2 + bx + c = 0$ 的解，其中浮点数系数 a, b, c 通过键盘输入。
5. 写程序将摄氏温度 C 转换为华氏温度 F ，转换公式为 $F = \left(\frac{9}{5}\right)C + 32$ ，其中 C 通过键盘输入。
6. 写程序计算圆柱体的体积，体积计算公式为 $V = \pi r^2 h$ ，其中底面半径 r 和高度 h 通过键盘输入。
7. 写程序计算 1 万元人民币可以兑换多少外币，键盘输入人民币兑换外币的汇率 a 。