

第 3 章 C++ 控制语句

程序是由一行一行的语句组成的。任何一个结构化程序都是由三种基本结构组成,即顺序结构、分支结构和循环结构。

本章介绍 C++ 中实现程序控制的各种语句,主要包括分支语句和循环语句。

3.1 C++ 语句概述

C++ 的语句可以分为控制语句、函数调用语句、表达式语句、空语句等 4 类。

1. 控制语句

控制语句主要完成分支结构程序和循环结构程序的控制,C++ 有以下 9 个控制语句:

- if-else 语句 分支语句
- switch 语句 多分支语句
- for 语句 循环语句
- while 语句 循环语句
- do-while 语句 循环语句
- continue 语句 结束本次循环语句
- break 语句 结束循环或结束 switch 语句
- goto 语句 转向语句
- return 语句 从函数返回语句

本章将介绍前 8 个语句,最后一个 return 语句将在第 4 章介绍。

2. 函数调用语句

C++ 有大量的系统函数,程序员也可以自己定义函数,对这些函数的调用可以作为一条语句。

3. 表达式语句

由一个表达式构成一个语句,即在表达式后添加一个分号,如赋值表达式语句等。

例如:

```
a=23;  
b=a+20 * 4;
```

自增自减运算加上分号也形成表达式语句,如:

```
i++;  
j--;
```

4. 空语句

空语句只有一个分号。即:

```
;
```

因此空语句什么也不做。

除了上面的简单语句外,可以将一组简单语句用大括号{}括起来,称之为复合语句。

3.2 if 语句实现选择结构

选择程序结构也称为分支结构,就是在程序运行过程中,根据具体条件执行不同的程序段。在 C++ 中,实现选择结构的语句有 if 语句和 switch 语句。if 语句比较适合分支较少的程序,switch 语句适合分支较多的程序。

3.2.1 引例

例 3.1 输出两个变量 a 和 b 较大的值。根据两个变量的大小执行不同的程序段,如果 a 大于 b,则输出 a 的值,否则输出 b 的值。

```
#include<iostream.h>
void main()
{
    int a,b;
    a=10;
    b=20;
    if(a>b)
    {
        cout<<"max="<<a<<endl;
    }
    else
    {
        cout<<"max="<<b<<endl;
    }
}
```

程序运行结果:

```
max=20
```

分析: 程序首先定义两个整型变量并分别赋值,如果表达式(a>b)为真,就执行 if 下面的程序段,输出 a 的值,否则执行 else 下面的程序段,输出 b 的值。

3.2.2 if 语句的基本结构

if 语句用来判定给定条件是否成立,根据判定结果决定执行两个分支中的哪一个。if 语句的一般格式如下:

```
if(表达式)
{
    语句组 1;
```

```

}
else
{
    语句组 2;
}

```

if 语句的执行过程：如果表达式的值为真，则执行语句组 1，否则执行语句组 2。可以用图 3.1 所示的流程图直观地表示执行过程。

注意：if 语句中的“表达式”必须用圆括号括起来。如果语句组只有一个简单的语句，则大括号可以省略。

有时并不需要两个分支，例如判断某课程的成绩是否需要补考，如果需要补考就输出补考信息，否则不需要进行处理，这时可以省略 else 部分。

例 3.2 输入一个成绩，判断是否需要补考，如果需要打印出来。

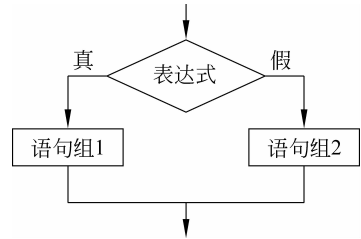


图 3.1 if-else 语句流程图

```

#include<iostream.h>
void main()
{
    int score;
    cout<<"请输入成绩：";
    cin>>score;
    if(score<60)
        cout<<"不及格,需要补考!"<<endl;
}

```

程序运行时，如果输入的成绩小于 60，就会输出“不及格，需要补考！”，如果输入的成绩大于或等于 60，则没有任何输出。

3.2.3 if 语句的嵌套

所谓 if 语句的嵌套是指在 if 语句的“语句组 1”或“语句组 2”中，又包含有 if 语句的情况。

if 语句嵌套时，else 子句总是与在它前面、距它最近、且尚未匹配的 if 配对。为明确匹配关系，避免匹配错误，建议将内嵌的 if 语句一律用花括号括起来。

例 3.3 输入任意三个数 a、b 和 c，输出其中最大的数。

```

#include<iostream.h>
void main()
{
    int a, b, c, max;
    cout<<"请输入 3 个整数,以空格分隔：";
    cin>>a>>b>>c;
    if (a>b)
    {

```

```

        if(a>c)
            max=a;
        else
            max=c;
    }
else
{
    if(b>c)
        max=b;
    else
        max=c;
}
cout<<"max="<<max<<endl;
}

```

程序运行结果：

请输入 3 个整数,以空格分隔: 45 64 32

max=64

3.2.4 用 if 语句实现多分支

if 语句不仅可以实现两个分支的程序结构,也可以实现多分支程序结构,参见下例。

例 3.4 从键盘上输入一个成绩;如果大于等于 90,输出“优”;如果大于等于 80,小于 90,输出“良”;如果大于等于 70,小于 80,输出“中”;如果大于等于 60,小于 70,输出“及格”;如果小于 60,输出“不及格”。

程序如下：

```

#include<iostream.h>
void main()
{
    int s;
    cout<<"请输入成绩";
    cin>>s;
    if (s>=90)
        cout<<"优 "<<endl;
    else if (s>=80)
        cout<<"良 "<<endl;
    else if (s>=70)
        cout<<"中 "<<endl;
    else if (s>=60)
        cout<<"及格 "<<endl;
    else
        cout<<"不及格 "<<endl;
}

```

多分支 if 语句的语法如下：

```

if(表达式 1)
{
    语句组 1;
}
else if(表达式 2)
{
    语句组 2;
}
:
else if(表达式 n)
{
    语句组 n;
}
else
{
    语句组 n+1;
}

```

上面 if 语句的执行过程如图 3.2 所示(以 3 个表达式,即 $n=3$,有 4 个分支的情况为例)。

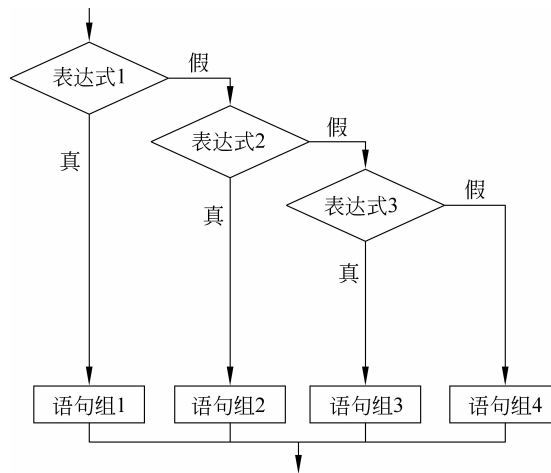


图 3.2 if-else 实现多分支流程图

如果表达式 1 成立,执行语句组 1,否则再判断表达式 2 是否成立,如果成立,执行语句组 2,否则再继续判断表达式 3……如果一直到表达式 n 都不成立,则执行语句组 $n+1$ 。

3.2.5 程序实例

例 3.5 根据以下函数编写程序。

$$y = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}$$

要求当输入一个 x 的值时,求出 y 的值并输出。

```
#include<iostream.h>
void main()
{
    int x,y;
    cout<<"请输入 x 的值: ";
    cin>>x;
    if(x<0)
        y=-1;
    else
    {
        if(x==0)
            y=0;
        else
            y=1;
    }
    cout<<"y="<<y<<endl;
}
```

如果输入的 x 值小于 0,将 -1 赋给 y ,否则执行 else 分支,再判断 x 是否等于 0,如果等于 0,则将 0 赋给 y ,否则 x 的值只能大于 0,将 1 赋给 y ,最后输出 y 的值。

例 3.6 求一元二次方程 $ax^2+bx+c=0$ 的解。

根据一元二次方程 3 个系数的关系,有以下几种情况:

- (1) $a=0$,不是二次方程。
- (2) $b^2-4ac=0$,有两个相等的实根。
- (3) $b^2-4ac>0$,有两个不相等的实根。
- (4) $b^2-4ac<0$,有两个共轭复根。

```
#include "iostream.h"
#include "math.h"
void main()
{
    double a,b,c,disc,x1,x2,p,q;
    cin>>a>>b>>c;
    if(fabs(a)<1e-6)
        cout<<"不是一元二次方程"<<endl;
    else
    {
        disc=b*b-4*a*c;
        if (fabs(disc)<=1e-6) //有两个相等的实根
            cout<<"x1=x2="<<-b/(2*a)<<endl;
        else
        {
            if (disc>1e-6) //有两个不相等的实根
```

```

    {
        x1=(-b+sqrt(disc))/(2*a);
        x2=(-b-sqrt(disc))/(2*a);
        cout<<"x1="<<x1<<endl;
        cout<<"x2="<<x2<<endl;
    }
else //有两个共轭复根
{
    p=-b/(2*a);
    q=sqrt(fabs(disc))/(2*a);
    cout<<"x1="<<p<<"+"<<q<<"i"<<endl;
    cout<<"x2="<<p<<"-"<<q<<"i"<<endl;
}
}
}
}

```

fabs()和sqrt()是C++预先定义的函数,分别用来求绝对值和平方根,我们称之为库函数。对于C++的库函数,我们可以直接使用,由于这两个函数是在头文件math.h中声明的,因此只要在使用之前添加下面的头文件包含即可。

```
#include "math.h"
```

由于disc(即 b^2-4ac)是一个实数,而实数在计算机中存储时,经常会有一些微小误差,所以不能直接判断disc是否等于0,而是判断disc的绝对值是否小于一个很小的数(例如 10^{-6})。如果小于此数,就认为disc等于0。

第一次运行输入 1 2 3,结果如下:

```
x1=-1+1.41421i
x2=-1-1.41421i
```

第二次运行输入 2 4 1,结果如下:

```
x1=-0.292893
x2=-1.70711
```

3.3 switch 语句实现多分支结构

用if语句虽然可以实现多分支结构,但比较烦琐。switch语句是专门用于实现多分支结构程序的。

3.3.1 引例

例 3.7 用switch语句实现例3.4的功能。

```
#include<iostream.h>
void main()
```

```

{
    int s,i;
    cout<<"请输入成绩";
    cin>>s;
    i=s/10;
    switch(i)
    {
    case 9:
    case 10: cout<<"优 "<<endl;
            break;
    case 8:  cout<<"良 "<<endl;
            break;
    case 7:  cout<<"中 "<<endl;
            break;
    case 6:  cout<<"及格 "<<endl;
            break;
    default: cout<<"不及格 "<<endl;
            break;
    }
}

```

程序开始输入一个成绩,然后将成绩除以 10 的结果赋给变量 i,因为两个整数相除的结果仍然是整数,如果 s 的值在 90~100 之间,i 的值是 10;如果 s 的值在 80~89 之间,i 的值是 8;如果 s 的值在 70~79 之间,i 的值是 7;如果 s 的值在 60~69 之间,i 的值是 6。switch 语句根据 i 的值执行不同的语句,如果 i 的值是 9,就从 case 9 后面开始执行,如果 i 的值是 10,就从 case 10 后面的语句开始执行,遇到 break 后,跳出 switch 语句段;如果 i 的值是 8,就执行 case 8 后面的语句,遇到 break 后,跳出 switch 语句段;如果 i 的值是 7,就执行 case 7 后面的语句,遇到 break 后,跳出 switch 语句段;如果 i 的值是 6,就执行 case 6 后面的语句,遇到 break 后,跳出 switch 语句段;如果 i 的值不在 6,7,⋯,10 这几个值之中,就执行 default 后面的语句,遇到 break 后,跳出 switch 语句段。

3.3.2 switch 语句的一般结构

switch 语句的一般格式如下:

```

switch(表达式)
{
    case 常量 1: 语句组 1;break;
    case 常量 2: 语句组 2;break;
    :
    case 常量 n: 语句组 n;break;
    default: 语句组 n+1;break;
}

```

switch 语句的执行过程如图 3.3 所示。

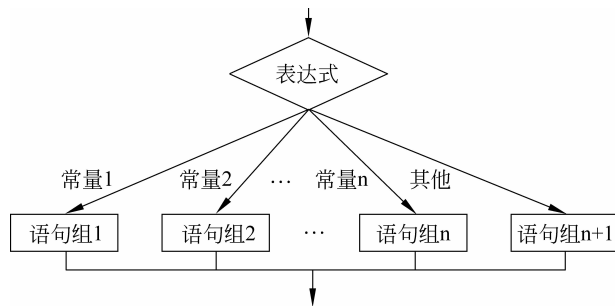


图 3.3 switch 语句流程图

当 switch 后面“表达式”的值，与某个 case 后面的“常量”值相同时，就执行该 case 后面的语句组；当执行到 break 语句时，跳出 switch 语句，执行 switch 语句的下一条语句。如果没有任何一个 case 后面的“常量”与“表达式”的值相同，则执行 default 后面的语句组；当执行到 break 语句时，跳出 switch 语句，执行 switch 语句的下一条语句。

注意：switch 后面的“表达式”，只能是整型、字符型或枚举型。每个 case 后面的“常量”值必须各不相同。case 后面的常量表达式仅起语句标号作用，并不进行条件判断，当程序执行某一个 case 后面的语句时，如果没有遇到 break 语句（break 语句是可以不出现的），就会一直执行下去。

3.3.3 程序实例

例 3.8 编一个程序计算指定年月有几天，年月由键盘输入。

分析：每年的 1、3、5、7、8、10、12 月份有 31 天，4、6、9、11 月份有 30 天，如果不是闰年，2 月份有 28 天，闰年的 2 月有 29 天。判断闰年的条件是：能被 4 整除但不能被 100 整除，或者能被 400 整除。程序如下：

```

#include<iostream.h>
void main()
{
    int year,month,days;
    cout<<"请输入年和月,用空格分隔" ;
    cin>>year>>month;
    switch(month)
    {
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        days=31;
        break;

```

```

case 4:
case 6:
case 9:
case 11:
    days=30;
    break;
case 2:
    if ((year%4==0) && (year%100 !=0) || (year%400==0) )
        days=29;
    else
        days=28;
    break;
default:
    days=0;
    break;
}
if(days==0)
    cout<<"月份输入有误!"<<endl;
else
    cout<<"该月份有 "<<days<<"天."<<endl;
}

```

3.4 循环结构

在许多问题中都需要用到循环结构,循环结构就是同一段程序要重复执行多次。C++语言中用于实现循环的语句有: for 语句、while 语句、do-while 语句。

3.4.1 引例

例 3.9 求 1~100 的累加和。

```

#include<iostream.h>
void main()
{
    int i;
    int sum=0;
    for(i=1; i<=100; i++)
    {
        sum+=i;
    }
    cout<<"sum="<<sum<<endl;
}

```

程序运行情况如下:

```
sum=5050
```