第5章 选择结构程序设计

本章重点：if 语句；if…else 语句；switch 语句。

本章难点：选择结构的嵌套。

选择结构程序设计是结构化程序设计的三种结构之一，在 C 语言中通过 if 语句和 switch 语句实现。选择结构中的程序代码根据条件的“真”和“假”决定要执行的语句，因此选择结构中的语句要么被执行了一次，要么不被执行。

5.1 为什么需要选择结构

在前面介绍的顺序结构中，程序中的每条语句是按照各个语句的先后顺序依次执行的，执行完一条语句后再无条件地执行下一条语句，这就是顺序结构。但在现实生活中，有许多问题用顺序结构是无法解决的，需要用到选择结构。也就是说执行完一条语句后不是无条件地执行下一条语句，而是需要进行选择，要选择就需要条件的判断。现实生活中很多问题都需要进行条件的判断，例如：

(1) 如果前面的交通信号灯是红色，那么我要等待，否则我可以通行。

(2) 如果一个整数能被 2 整除，那么我可以判断它是偶数，否则它是奇数。

(3) 如果 a 大于 b，那么输出 a，否则输出 b。

在第(1)个例子中，我是“等待”还是“通行”需要进行条件的判断，这个条件就是“交通信号灯是否是红色”；在第(2)个例子中，这个数是“偶数”还是“奇数”需要进行条件的判断，这个条件就是“能否被 2 整除”；在第(3)个例子中，到底是输出 a 还是 b 需要进行条件的判断，这个条件就是“a 是否大于 b”。

条件判断的结果是一个逻辑值，当条件成立时，表示为“真”，当条件不成立时，表示为“假”。条件“交通信号灯是否是红色”，条件判断的结果只有两个，要么是红灯，要么不是红灯；条件“能否被 2 整除”，条件判断的结果也只有两个，要么能，要么不能；条件“a 是否大于 b”，条件判断的结果同样也只有两个，要么是，要么否。当然，有些时候，我们也用“真”和“假”来表示条件判断的结果，当条件成立时，表示为“真”，条件不成立时，表示为“假”。

无论是在现实生活中，还是在科学计算、工程管理等领域中都经常要进行条件的判断。所以在程序中需要用选择结构来判断某个条件是否满足，然后再根据条件判断的结果来决定要进行的操作。

5.2 用 if 语句实现选择结构

5.2.1 单分支 if 语句

单分支 if 语句的形式为

```
if (表达式)
    语句
```

单分支 if 语句首先判断表达式的值,如果表达式的值为真,则执行后面的语句;如果表达式的值为假,则不执行后面的语句。这里的语句可以是一条简单的语句,也可以是一条复合语句。单分支 if 语句的控制流程图如图 5.1 所示。

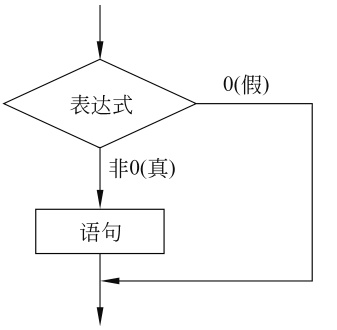

图 5.1 单分支 if 语句的控制流程图

【例 5.1】 编写程序,输入一个整数,判断这个数是否是偶数。

编程思路:

首先定义一个内存变量 x 存储整数,然后使用表达式(x%2==0)判断是否是偶数。

编写程序:

```
#include <stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    if(x%2==0)
        printf("%d是一个偶数。\n",x);
    return 0;
}
```

运行结果:

```
18
18是一个偶数。
```

5.2.2 双分支 if 语句

双分支 if 语句的形式为

```
if (表达式)
    语句1
else
    语句2
```

双分支 if 语句首先判断表达式的值,如果表达式的值为真,则执行语句 1;如果表达式

的值为假，则执行语句 2。双分支 if 语句的控制流程图如图 5.2 所示。

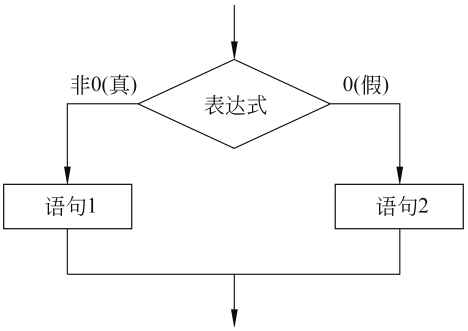

图 5.2　双分支 if 语句的控制流程图

【例 5.2】 编写程序，输入一个整数，判断这个数是奇数还是偶数。

编程思路：

首先定义一个内存变量 x 存储整数，然后使用表达式(x%2==0)判断是奇数还是偶数。如果表达式的值为 1，则这个数是偶数，否则这个数就是奇数。

编写程序：

```
#include <stdio.h>
int main()
{
    int x;
    scanf("%d",&x);
    if(x%2==0)
        printf("%d是一个偶数。\n",x);
    else
        printf("%d是一个奇数。\n",x);
    return 0;
}
```

运行结果：

```
5
5是一个奇数。
```

5.2.3　多分支 if 语句

多分支 if 语句的形式为

```
if (表达式 1)
    语句 1
else if (表达式 2)
        语句 2
    else if (表达式 3)
            语句 3
```

```
...
else if (表达式 n)
    语句 n
```


多分支 if 语句首先判断表达式 1 的值,如果表达式的值为真,则执行语句 1,后面的语句不再执行;如果表达式的值为假,则再判断表达式 2 的值。如果表达式 2 的值为真,则执行语句 2,后面的语句不再执行;如果表达式 2 的值为假,则继续判断表达式 3 的值。以此类推找到成立的条件然后执行对应的语句。多分支 if 语句的控制流程图如图 5.3 所示。

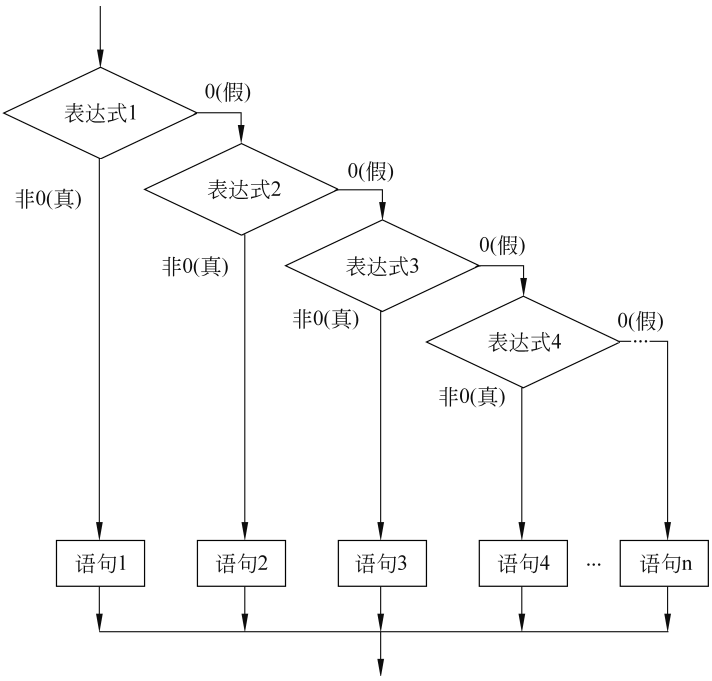

图 5.3　多分支 if 语句的控制流程图

【例 5.3】 编写程序,计算分段函数的值。

$$y=\begin{cases}2x & x\leqslant -10\\ 2/x & -10<x<0\\ 2+x & 0\leqslant x\leqslant 10\\ 2-x & x>10\end{cases}$$

编程思路:

首先定义一个内存变量 x,输入 x 的值后判断 x 的范围,然后根据分段函数计算函数值。

编写程序:

```
#include <stdio.h>
int main()
{
    float x,y;
    printf("请输入 x 的值:");
```

```
    scanf("%f",&x);
    if(x<=-10)
        y=2*x;
    else
        if(x>-10&&x<0)
            y=2/x;
        else
            if(x>=0&&x<=10)
                y=2+x;
            else
                if(x>10)
                    y=2-x;
    printf("y=%f\n",y);
    return 0;
}
```

运行结果：

```
请输入 x 的值:8
y=10.000000
```

5.3 用 switch 语句实现选择结构

switch 语句的一般形式为

```
switch(表达式)
{
    case 常量表达式 1:语句 1
    case 常量表达式 2:语句 2
    case 常量表达式 3:语句 3
    …
    case 常量表达式 n:语句 n
    default:语句 n+1
}
```

说明：

(1) switch 语句是多分支选择语句，其特点是可以根据一个表达式的多种值来选择多个分支。虽然也可以用多分支的 if 语句或嵌套的 if 语句实现，但如果分支较多，则会导致多分支 if 语句和嵌套的 if 语句层次多，程序冗长且可读性降低。

(2) switch 后面的表达式的值可以是整型、字符型、枚举型等。常量表达式的值必须互不相同，当表达式的值与某一个 case 后面的常量表达式的值相同时，就执行该 case 后面的语句。如果没有任何一个 case 后面的常量表达式的值与其相匹配时，则执行 default 后面的语句。

(3) 在执行 switch 语句时，根据表达式的值找到入口，也即对应的 case，执行完成对应

的语句后,程序继续执行下一个case,而不再继续判断。

(4) 执行完一个case以后,如果要终止switch语句的执行,可以用break语句来实现。break语句的一般形式:


```
break;
```


【例5.4】 编写程序,输入考试成绩(百分制),打印出其对应的等级。

- 小于60:不及格。
- 大于或等于60,且小于70:及格。
- 大于或等于70,且小于80:中等。
- 大于或等于80,且小于90:良好。
- 大于或等于90,且小于等于100:优秀。

编程思路:

首先要设计好switch后面的表达式,因为考试成绩为0～100的实数,如果把每个值都用一个case列出来,那是不可能实现的。所以可以把0～100的实数划分成0～60、60～70、70～80、80～90、90～100、100进行处理。

编写程序:

```
#include <stdio.h>
int main()
{
    float score;
    printf("请输入学生成绩:");
    scanf("%f",&score);
    switch((int)(score/10))
    {
        case 10:
        case 9:printf("优秀\n");break;
        case 8:printf("良好\n");break;
        case 7:printf("中等\n");break;
        case 6:printf("及格\n");break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:printf("不及格\n");break;
        default:printf("输入有误!\n");
    }
    return 0;
}
```

运行结果:

```
请输入学生成绩:78
中等
```

5.4 选择结构的嵌套

选择结构中又包含一个或多个选择结构，称为选择结构的嵌套。一般形式为

```
if(表达式 1)
    if(表达式 2)  语句 1
    else  语句 2
else
    if(表达式 3)  语句 3
    else  语句 4
```

有些时候可能不是 if(表达式 1)和 else 后同时又有 if 语句结构，可能只在 if(表达式 1)或 else 后才有 if 语句结构，这种情况仍然是选择结构的嵌套。甚至有些时候 if(表达式 1)后面的 if(表达式 2)中又还有选择结构。

【例 5.5】 编写程序，输入 3 个整数 a、b、c，输出 3 个整数中最大的数。

编程思路：

要找出 3 个数中最大的数，首先要找出 a 和 b 两个数中较大的数，然后再把前面两个数中较大的数与第 3 个数比较，从而找出最大的数。

编写程序：

```
#include <stdio.h>
int main()
{
    int a,b,c;
    printf("请输入 3 个整数:");
    scanf("%d,%d,%d",&a,&b,&c);
    if(a<b)
        if(b<c)
            printf("3 个整数中的最大数是:%d\n",c);
        else
            printf("3 个整数中的最大数是:%d\n",b);
    else
        if(a<c)
            printf("3 个整数中的最大数是:%d\n",c);
        else
            printf("3 个整数中的最大数是:%d\n",a);
    return 0;
}
```

运行结果：

```
请输入 3 个整数:89,45,120
3 个整数中的最大数是:120
```

5.5 选择结构程序设计综合举例

【例5.6】 编写程序,输入一个年份,判断这个年份是否是闰年。

编程思路:

定义一个内存变量year存储年份,根据前面介绍的判断闰年的条件:(year%4==0&&year%100!=0)||(year%400==0),然后使用if语句实现。

编写程序:

```
#include <stdio.h>
int main()
{
    int year;
    printf("请输入一个年份:");
    scanf("%d",&year);
    if((year%4==0&&year%100!=0)||( year%400==0))
        printf("%d是闰年。\n",year);
    else
        printf("%d不是闰年。\n",year);
    return 0;
}
```

运行结果:

```
请输入一个年份:1992
1992是闰年。
```

【例5.7】 编写程序,求一元二次方程 $ax^2+bx+c=0$ 的根。

编程思路:

求一元二次方程的根,首先要保证二次项系数a不等于0,然后再判断 $b^2-4*a*c$ 的值,如果 $b^2-4*a*c$ 的值大于或等于0,则一元二次方程有实根;如果 $b^2-4*a*c$ 的值小于0,则一元二次方程有两个共轭复数根。

编写程序:

```
#include <stdio.h>
#include <math.h>
int main()
{
    float a,b,c,x1,x2,rp,ip;
    printf("请输入一元二次方程的二次项系数、一次项系数和常数项:");
    scanf("%f,%f,%f",&a,&b,&c);
    if(a==0)
        printf("不是一元二次方程!\n");
    else
        if (b*b-4*a*c>0)
```

```
        {
            x1=(-b+sqrt(b*b-4*a*c))/(2*a);
            x2=(-b-sqrt(b*b-4*a*c))/(2*a);
            printf("方程有两个实根:%8.4f,%8.4f\n",x1,x2);
        }
        else
            if (b*b-4*a*c==0)
                printf("方程有两个相等的实根:%8.4f\n",-b/(2*a));
            else
            {
                    rp=-b/(2*a);
                    ip=sqrt(-(b*b-4*a*c))/(2*a);
                    printf("方程有两个复数根:\n");
                    printf("%8.4f+%8.4fi\n",rp,ip);
                    printf("%8.4f-%8.4fi\n",rp,ip);
            }
    return 0;
}
```

运行结果：

```
请输入一元二次方程的二次项系数、一次项系数和常数项:2,8,1
方程有两个实根:- 0.1292,- 3.8708
```

【例 5.8】 编写程序，输入 3 个数，然后按照由大到小的顺序输出这 3 个数。

编程思路：

这是一种最简单的排序问题。定义 3 个内存变量 a、b、c，要进行排序，首先将 a 和 b 比较，如果 a＜b，则内存变量 a 和 b 的值要进行交换，然后再将 a 和 c 比较，如果 a＜c，则内存变量 a 和 c 的值要进行交换，最后再将 b 和 c 比较，如果 b＜c，则内存变量 b 和 c 的值要进行交换。通过这样的比较后，内存变量 a 存储的是最大值，b 存储的是较大值，c 存储的就是最小值。依次输出就是按由大到小的顺序输出。

编写程序：

```
#include <stdio.h>
int main()
{
    float a,b,c,t;
    printf("请输入 3 个数:");
    scanf("%f,%f,%f",&a,&b,&c);
    if(a<b)
      {t=a;a=b;b=t;}
    if(a<c)
      {t=a;a=c;c=t;}
    if(b<c)
      {t=b;b=c;c=t;}
```

```
    printf("%6.2f,%6.2f,%6.2f\n",a,b,c);
    return 0;
}
```


运行结果：

```
请输入3个数:8,2,9
    9.00,    8.00,    2.00
```

【例5.9】 编写程序，制作一个学生成绩管理菜单，要求按下数字0～6中的某一个数字时执行如下相应操作：

- 0——退出，即退出菜单选择。
- 1——输出“显示”。
- 2——输出“添加”。
- 3——输出“排序”。
- 4——输出“查找”。
- 5——输出“计算”。
- 6——输出“保存”。

编程思路：

这是一种简单的菜单制作，采用switch语句实现选择。其中退出功能需要使用exit(0)函数，exit()通常是在子程序中用来终结程序的，使用后程序自动结束，exit(0)表示正常退出，所在函数库为stdlib.h，也就是在程序中要包含头文件“stdlib.h”。

编写程序：

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  int code;
  printf("\n\t       学生成绩管理\n");
  printf("\t┌──────────────────┐\n");
  printf("\t|          0. 退出          |\n");
  printf("\t|          1. 显示          |\n");
  printf("\t|          2. 添加          |\n");
  printf("\t|          3. 排序          |\n");
  printf("\t|          4. 查找          |\n");
  printf("\t|          5. 计算          |\n");
  printf("\t|          6. 保存          |\n");
  printf("\t└──────────────────┘\n");
  printf("          请选择代码(0,1,2,3,4,5,6):\n");
  { scanf("%d",&code);          //输入数字
    switch(code)
    {
      case 0:exit(0);           //退出
```