第 3 章 选择结构

C 语言程序设计是面向过程的结构化程序设计，其中可以执行的流程分为 3 种：顺序结构、选择结构、循环结构。顺序结构是程序流程控制的基本结构，为了实现各种功能的算法，除了按照正常的流程顺序执行外，还需要改变顺序控制流程以实现某种功能，这时需要用到选择结构、循环结构。

(1) 顺序结构：自上而下顺序执行每一条语句。

(2) 选择结构：根据某种条件是否成立，选择执行不同的语句。

(3) 循环结构：根据条件(循环条件)决定是否重复执行一些语句(循环体)。

本章主要介绍选择结构，包括 if 语句和 switch 语句及它们的应用。学习时要关注以下问题：

(1) 两种主要选择结构的语法和用法，包括条件表达式的编写、执行路径的选择等；

(2) 如何利用选择结构解决实际问题。

3.1 语句

一个程序由若干语句构成，语句的作用是向计算机系统发出操作指令，要求执行相应的操作。C 语言中的语句主要有控制语句、表达式语句、空语句、复合语句、函数调用语句等。

1. 控制语句

完成一定的控制功能，C 语言中有以下 9 种控制语句。

if…else(条件语句)、for(循环语句)、while(循环语句)、do…while(循环语句)、continue(结束当前循环语句)、break(终止执行 switch 或循环语句)、switch(多分支选择结构)、return(从函数返回语句)、goto(转向语句)。

2. 表达式语句

表达式加分号就构成了表达式语句。

最典型的是由赋值表达式加分号构成一个赋值语句。例如，i=i+1 或 i++ 是表达式，而“i=i+1;”和“i++ ;”是表达式语句。赋值表达式可以包含在其他表达式中，但赋值语句不能包含在其他语句和表达式中。例如，while((ch=getchar())!=='\n');语句中的条件表达式包含赋值表达式 ch=getchar()，但不能包含赋值语句 ch=getchar();。该语句的执行是首先进行赋值运算 ch=getchar()，然后判断变量 ch 是否为换行符'\n'，如果不是，则继续执行循环体内的代码，直到遇到换行符为止。

任何表达式都可以加上分号而成为语句，包括算术运算表达式、关系运算表达式、逻辑运算表达式等。如 3+5*4;是一条合法语句，只是没有将运算结果赋值给任何一个变量，所以单独执行没有意义。

3. 空语句

只有分号的语句就是空语句。它什么都不做。空语句有时用作转向点，或用作循环语句中的循环体，表示循环体什么也不做。例如，while((ch=getchar())==''); 该语句表示将跳过输

入字符串中的空白字符，直到遇到非空白字符为止。执行过程中，首先会执行 ch＝getchar()，将输入的字符赋值给变量 ch，然后检查这个字符是否为空白字符，如果是，则继续执行循环体内的代码，即继续读取下一个字符，直到遇到非空白字符为止。

4. 复合语句

用一对大括号“{”和“}”括起来的语句就是复合语句，多用于选择结构、循环结构中。在程序结构上，复合语句看作一个整体，但是内部可能完成了一系列工作。例如：

```
{   x = 3;
    printf("x = %d",x);
}
```

5. 函数调用语句

由函数调用加一个分号构成一个语句，例如：

```
printf("Hello world!");          //调用库函数 printf()
```

Tips

① C 语言中每一条语句在最后必须出现分号，分号是语句中不可缺少的组成部分，而不是两个语句间的分隔符号。

② 赋值表达式可以包含在其他表达式中，但赋值语句不能包含在其他语句和表达式中。

③ 空语句可用作循环语句中的循环体，表示循环体什么也不做。

④ 复合语句是用一对大括号“{”和“}”将语句括起来完成了一系列工作。

视频讲解

3.2 if 语句

C 语言中的 if 语句用来判断给定条件是否满足，根据判断结果决定执行相应的操作，C 语言提供了 4 种形式的 if 语句，分别是单分支的 if 语句、双分支的 if…else 语句、多分支的 else…if 语句及嵌套的 if 语句。

▶ 3.2.1 单分支 if 语句

单分支 if 语句格式如下，其中表达式一般是关系表达式或逻辑表达式。

```
if(表达式)
    语句段
```

执行时先计算表达式的值，如果表达式的值为真，执行后面的语句段；否则不执行，即跳过该语句段，继续执行后面的语句。其流程图如图 3.1 所示。

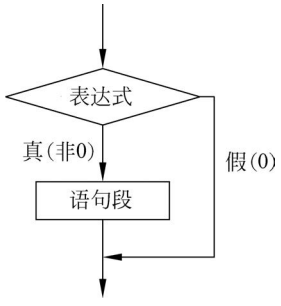

图 3.1 单分支 if 语句的流程图

【例 3.1】 3 个数字排序。

【问题描述】 输入 3 个整数 a,b,c,要求将它们从大到小排序后输出。

【问题分析】 对 3 个数进行排序,可以将最大的放在 a 中,最小的放在 c 中,按照 a,b,c 的顺序输出即可。将最大的放在 a 中,需要将 a 分别与 b、c 进行比较,如果 a 小进行交换即可;将最大的数换在 a 中后,再比较一下 b 和 c,将较小的放在 c 中即可。

【参考代码】

```
#include <stdio.h>
int main()
{
  int a,b,c,t;
  scanf("%d%d%d",&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("%d %d %d\n",a,b,c);
  return 0;
}
```

【代码分析】 使用 if 语句时,如果表达式的值为非 0,则其后只能执行一条语句时,可采取逗号表达式:

```
if(a<b)
  t=a,a=b,b=t;
```

【拓展思考】 如何对 4 个数进行排序?

> Tips
>
> ① if 语句根据表达式的真假来决定是否执行语句段。
>
> ② if 语句后面的表达式可以是逻辑表达式或关系表达式,也可以是其他表达式,甚至是一个变量,表达式非 0 为逻辑真,0 为逻辑假。
>
> ③ 这里所说的语句段可以是一条语句,也可以是多条语句,甚至是一个控制结构(如果是个控制结构,就组成了嵌套的控制结构);如果是一条语句,不要漏掉语句后的分号;如果是多条语句,需要用“{”和“}”括起来,形成复合语句。

▶ 3.2.2 双分支 if…else 语句

if…else 双分支语句的一般形式如下,可以根据表达式结果选取两路指令序列中的一路进行操作。

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

执行时先计算表达式的值，如果表达式的值为真（非 0），执行语句段 1；否则执行语句段 2。其流程图如图 3.2 所示。

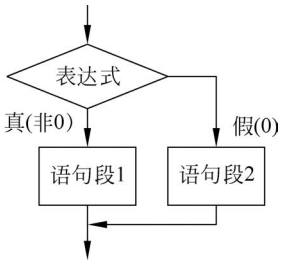

图 3.2　双分支 if…else 语句的流程图

【素质拓展】　鱼与熊掌不可兼得

根据条件真假来选择是执行语句段 1 还是语句段 2，如同“鱼与熊掌不可兼得”，探讨生活中遇到的两难选择时要如何取舍，人生路上有很多需要做选择的情况，要树立正确的人生观和价值观，特别当面临个人利益与社会利益乃至国家利益冲突时，要以大局为重，以社会利益、国家利益为重。

【例 3.2】　求两个数中的较大者。

【问题描述】　输入两个整数 a，b，求其中的较大者。

【问题分析】　本题旨在考查 if…else 语句，对给定的两个数进行判断，输出较大的数。

【参考代码】

```
#include <stdio.h>
int main()
{
  int a,b,c;
  scanf("%d%d",&a,&b);
  if(a<b)
     c=b;
  else
     c=a;
  printf("%d\n",c);
  return 0;
}
```

【拓展思考】　输入 3 个数，输出其中的最小数，要求采用 if…else 双分支语句来实现。

Tips

① if…else 语句是根据表达式的真假从语句段 1 和语句段 2 中选择一路来执行。

② 在双分支结构中，else 必须与 if 配对使用，构成 if…else 语句，实现双分支选择。

③ 语句段 1 和语句段 2 可以是一条语句，也可以是多条语句，甚至是一个控制结构；如果是多条语句，需要用“{”和“}”括起来，形成复合语句。

▶ 3.2.3　多分支 else…if 语句

根据一个条件的真假，可以分成两支解决问题的路线。若一个问题有 n 种情况，就需要 n－1 个条件，依次判断给定的 n－1 个条件，以确定从 n 组操作中选择某一组的结构称为多分支结构，其格式如下。

```
if(表达式 1)
    语句段 1
else if(表达式 2)
    语句段 2
else if(表达式 3)
    语句段 3
…
else if(表达式 n－1)
    语句段 n－1
else
    语句段 n
```

在多分支 else…if 语句中依次计算表达式 i(i=1,2,…,n－1)的值,当表达式 i 为真时,执行与之相关的语句段 i,并以此结束整个多分支结构的执行。其流程图如图 3.3 所示。

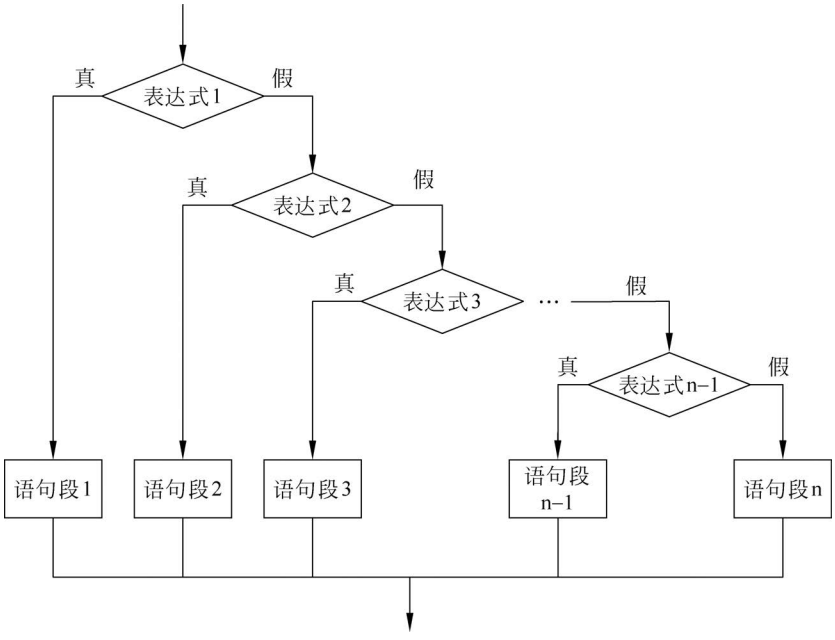

图 3.3 多分支 else…if 语句的流程图

【例 3.3】 判断字符。

【问题描述】 从键盘输入一个字符,根据字符的 ASCII 值,判断它是数字、大写字母、小写字母,还是其他字符,并输出说明结果。

【问题分析】 根据输入的字符,逐步判断字符所在的范围,N-S 图如图 3.4 所示。

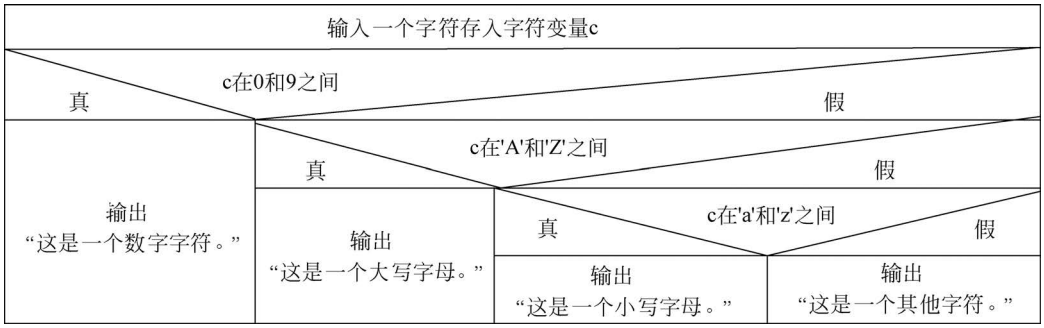

图 3.4 例 3.3 的 N-S 图

【参考代码】

```
#include<stdio.h>
int main()
{
    char c;
    printf("请输入一个字符:");
    c=getchar();
    if(c>='0'&&c<='9')
        printf("这是一个数字字符。\n");
    else if(c>='A'&&c<='Z')
        printf("这是一个大写字母。\n");
    else if(c>='a'&&c<='z')
        printf("这是一个小写字母。\n");
    else
        printf("这是一个其他字符。\n");
    return 0;
}
```

【拓展思考】 输入一个字母(大写或小写),输出其对应的另一字母(小写或大写)。

▶ 3.2.4 if 嵌套

在 if…else 语句中语句段 1 和语句段 2 既可以是一个简单的语句,也可以是另一个 if…else 语句,从而形成 if…else 语句的嵌套。一般结构形式如下:

```
if(表达式 1)
    if(表达式 2)
        语句段 1                    //表达式 1 为真,且表达式 2 也为真时执行
    else
        语句段 2                    //表达式 1 为真,且表达式 2 为假时执行
else
        语句段 3                    //表达式 1 为假时执行
```

也可以有如下结构:

```
if(表达式 1)
    语句段 1                        //表达式 1 为真时执行
else
    if(表达式 2)
        语句段 2                    //表达式 1 为假,且表达式 2 为真时执行
    else
        语句段 3                    //表达式 1 为假,且表达式 2 为假时执行
```

或如下结构:

```
if(表达式 1)
    if(表达式 2)
        语句段 1                    //表达式 1 为真,且表达式 2 也为真时执行
    else
        语句段 2                    //表达式 1 为真,且表达式 2 为假时执行
else
```

```
    if(表达式 3)
        语句段 3                          //表达式 1 为假,且表达式 3 为真时执行
    else
        语句段 4                          //表达式 1 为假,且表达式 3 也为假时执行
```

【例 3.4】　求一元二次方程的根。

【问题描述】　利用公式 $x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$ 求一元二次方程 $ax^2+bx+c=0$ 的根,其中 a 不等于 0。要求输出一行,表示方程的解。

若 $b^2=4ac$,则两个实根相等,输出为 $x_1=x_2=\frac{-b}{2a}$。

若 $b^2>4ac$,则两个实根不等,输出为 $x_1=\frac{-b+\sqrt{b^2-4ac}}{2a}$,$x_2=\frac{-b-\sqrt{b^2-4ac}}{2a}$,其中 $x_1>x_2$。

若 $b^2<4ac$,则有两个虚根,输出为"x_1=实部+虚部 i; x_2=实部-虚部 i",实部为 0 时不可省略。实部=-b/(2a),虚部=sqrt(4ac-b^2)/(2a)。

所有实部要求精确到小数点后 5 位,数字、符号之间没有空格。

【问题分析】　本题旨在考查 if…else 嵌套结构和输出的格式控制,需要根据不同的情况进行处理。

【参考代码】

```
#include <stdio.h>
#include"math.h"
int main()
{
  double a,b,c,delta,x1,x2,y1,y2,t;
  scanf("%lf%lf%lf",&a,&b,&c);
  if(fabs(a)<1e-6)                              //判断 a 是否相当小
    printf("输入的系数不能构成二次方程!");       //a=0 时不是二次方程
  else
  {
    delta=b*b-4*a*c;
    if(fabs(delta)<1e-6)                        //Δ=0 时有两个相等的根
      printf("x1=x2=%.5lf\n",-b/(2*a));
    else
      if(delta>0)                               //Δ>0 时有两个实数根
      {
        x1=(-b+sqrt(delta))/(2*a);              //sqrt 函数用来求出参数的平方根
        x2=(-b-sqrt(delta))/(2*a);
        if(x1<x2)                               //交换两个实根
          t=x1,x1=x2,x2=t;
        printf("x1=%.5lf;x2=%.5lf",x1,x2);
```

```
        }
        else                                    //Δ<0 时有两个复数根
        {
          x1 = -b/(2*a);
          y1 = sqrt(-delta)/(2*a);
          y2 = -sqrt(-delta)/(2*a);
          if(y1<y2)                             //交换两个虚根
           t = y1,y1 = y2,y2 = t;
          printf("x1 =%.5lf",x1);
          if(y1>0)
            printf("+%.5lfi;x2 =%.5lf",y1,x1);
          else
            printf("%.5lfi;x2 =%.5lf",y1,x1);
          if(y2>0)
            printf("+%.5lfi",y2);
          else
            printf("%.5lfi",y2);
        }
    }
    return 0;
}
```

【代码分析】 程序中用变量 delta 表示公式 b^2-4ac，先计算 delta 的值，以减少以后进行判断和计算时的重复计算。当判断变量 a 和 delta（即 b^2-4ac）是否等于 0 时，要注意一个问题，由于 a 和 delta 是实数，而实数在计算和存储时会有一些微小的误差，因此不能直接用“a==0”和“delta==0”进行判断，因为这样可能会对本来是零的量，由于出现误差而被判断为不等于 0，从而导致结果错误。本例中采取的办法是使用 fabs()函数判断它们的绝对值是否小于或等于一个很小的数（如 10^{-6}），如果小于或等于此数，就认为它们等于 0。

【素质拓展】 工匠精神

该问题采用了选择结构嵌套来解决，是由简单到复杂的过程，只有独立思考，才能学到知识、掌握技能。面对复杂问题，也只有思考方能明白其中奥妙，懂得正确选择。程序编写过程中，有不完善之处，如对于是否等于 0 的判断，要写成 fabs(delta)<1e-6，这要求学生要养成逻辑严密、精益求精的工匠精神，尽量把事情做到极致。

【拓展思考】 下述分支结构中，语句 2 的执行条件是什么？

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

【提示】 else 与表达式 2 对应的 if 匹配，所以语句 2 执行的条件是表达式 1 为真，表达式 2 为假。在编写程序时要正确地安排语句的缩进格式，本练习中的缩进格式就不正确，容易引起误解。

视频讲解

3.3 switch 语句

对于多分支结构，使用 else…if 语句形式编程，得到的程序往往变得冗长，降低可读性，这时可以采用 C 语言提供的 switch 语句（又称开关语句）来实现多分支结构。其一般格式如下：

```
switch(表达式)
{
  case 常量表达式 1:语句段 1;[break;]
  case 常量表达式 2:语句段 2;[break;]
  …
  case 常量表达式 n:语句段 n;[break;]
  [default:语句段 n+1;]
}
```

执行时先计算 switch 后面表达式的值，然后逐个与 case 中的常量表达式的值比较。如果两者相等，执行相应 case 后面的语句段，如果 case 语句段后有 break 语句，执行完语句段后则控制跳出 switch 语句，执行 switch 之后的语句；如果 case 语句段后没有 break 语句，执行完语句段后将继续往下执行后面所有 case 后的语句，直到有 break 或者遇到 switch 语句的“}”中止该结构。如果两者不相等，继续执行下一个 case 结构进行判断。若表达式的值与所有 case 中的表达式的值均不相等时，执行 default 后面的语句。switch 语句的流程图如图 3.5 所示。

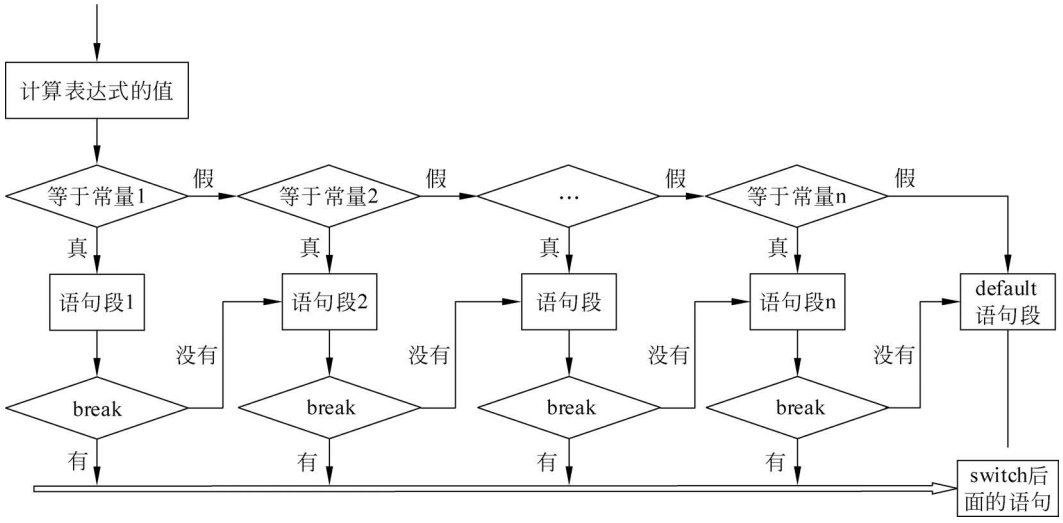

图 3.5 switch 语句的流程图

【例 3.5】 等级成绩。

【问题描述】 学院的考试采用等级制，即将百分制转换为 A、B、C、D、E 这 5 个等级，设成绩为 X，则 90≤X≤100 为 A，80≤X<90 为 B，70≤X<80 为 C，60≤X<70 为 D，X<60 为 E。编写一个程序，将输入的分数转换成 A、B、C、D、E 这 5 个等级。

【问题分析】 本题旨在考查 switch 语句的应用。对问题进行分析，将成绩分为 5 个区间，考虑到 switch 后面的表达式只能是整型，因此需要将输入的浮点数转换成整型，因此，可以有效利用整数的整除，将输入的成绩转换为 0～10 的整数。

【参考代码】

```
#include <stdio.h>
int main()
{
  float score;
  scanf("%d",&score);
  switch((int)score/10)
  {
    case 10:
    case 9:printf("A");break;
    case 8:printf("B");break;
    case 7:printf("C");break;
    case 6:printf("D");break;
    default:printf("E");
  }
  return 0;
}
```

【代码分析】 (int)score 是将输入的浮点数转换为整数，整除 10 后会产生 0～10 的相关整数。考虑到 0～5 这 6 种情况对应等级 E，因此将其放入 default 中。当结果是 10 和 9 时，均为等级 A，因此，case 10 后面不增加任何代码，只是作为一个标号存在。输出相应的等级后，增加 break 语句，结束相应的判断。

【拓展思考】 用 switch 语句编制程序，输入 10 个学生的 C 语言成绩（百分制整型数），统计各等级的人数。其中 A 级：90～100，B 级：80～89，C 级：70～79，D 级：60～69，E 级：＜60。

Tips

① switch 后面小括号中表达式的类型必须是整型或字符型。

② case 后面的“常量表达式”的值必须是整型或字符型，case 后面的常量表达式的值互不相同，否则会出现矛盾的现象，用中括号括起来的 default 子句可以省略。

③ case 后面的“常量表达式”起一个程序入口的标号作用。系统一旦找到入口标号，从此标号处开始执行，不再与其后 case 语句的常量表达式进行比较判断，所以为了终止一个分支的执行，需要在相应的分支末尾加一个 break 语句，跳出 switch 语句，使得程序转向 switch 后面的语句。

④ 多个 case 语句可以共同使用一组执行语句，某个 case 后面如果有多条执行语句，不必用“{”和“}”括起来。

⑤ switch 语句的执行都用“{”和“}”括起来，不能省略。

3.4 本章小结

本章所涉及的知识思维导图如图 3.6 所示。首先介绍了 C 语句的分类和按语句顺序执行的顺序结构及应用，然后重点讲解了选择结构的两种基本语句：if 语句（if 单分支、if…else 双分支、else…if 多分支、if 嵌套）和 switch 语句（多分支的操作，注意合理使用 break）。结合具体案例能够合理使用选择结构，这是学习本章的目的。

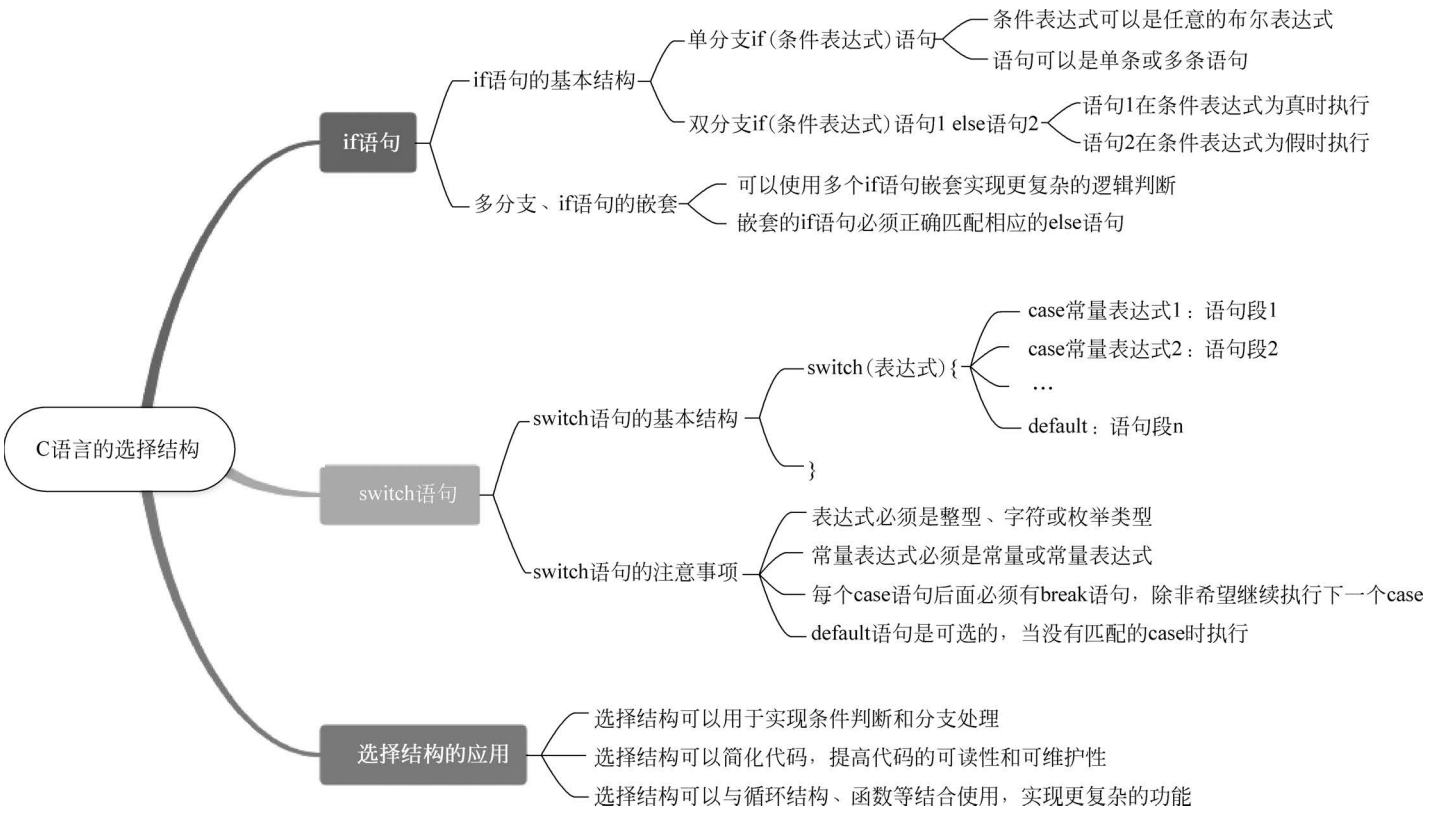
C语言的选择结构
if语句
if语句的基本结构
单分支if（条件表达式）语句
条件表达式可以是任意的布尔表达式
语句可以是单条或多条语句
双分支if（条件表达式）语句1 else语句2
语句1在条件表达式为真时执行
语句2在条件表达式为假时执行
多分支、if语句的嵌套
可以使用多个if语句嵌套实现更复杂的逻辑判断
嵌套的if语句必须正确匹配相应的else语句
switch语句
switch语句的基本结构
switch（表达式）{
case常量表达式1：语句段1
case常量表达式2：语句段2
…
default：语句段n
}
switch语句的注意事项
表达式必须是整型、字符或枚举类型
常量表达式必须是常量或常量表达式
每个case语句后面必须有break语句，除非希望继续执行下一个case
default语句是可选的，当没有匹配的case时执行
选择结构的应用
选择结构可以用于实现条件判断和分支处理
选择结构可以简化代码，提高代码的可读性和可维护性
选择结构可以与循环结构、函数等结合使用，实现更复杂的功能

图 3.6　第 3 章思维导图

在线测试

3.5 拓展习题

1. 基础部分

（1）如果整型变量 a 原来的值为 4，则 if(a==3)和 if(a=3)的判断结果有什么区别？

（2）编制程序，输入一个整数，用单分支结构求其绝对值，并输出。

（3）输入一个 0～100 的考试得分数，如果该得分数小于 60，输出“不及格”；如果该得分数大于或等于 60，输出“及格”。

（4）输入一个整数，判断并输出它是偶数还是奇数。

（5）超市买白菜问题：已知某超市内大白菜的单价是根据单次购买的重量来决定的，单次购买 5 kg 以下 1.8 元/kg，5 kg 以上（包括 5 kg，下同）1.6 元/kg；10 kg 以上 1.4 元/kg；20 kg 以上 1.0 元/kg。编程输入购买大白菜的千克数，输出应付的钱数。

（6）某公司销售人员的工资算法如表 3.1 所示。销售的产品单价为 1000 元/件，若每月销售 5 件以下（不含 5），没有提成；5 件以上 10 件以下（含 5，不含 10），该部分每件提成 4%；若每月销售 10 件以上 15 件以下（含 10，不含 15），该部分每件提成 6%；每月销售 15 件以上（含 15），该部分每件提成 10%。编制程序完成以下要求：输入销售件数，输出销售人员的工资。

表 3.1 销售提成计算表

销售件数 x	底薪工资/元	提成比例
[1,5)	2000	0
[5,10)	2000	4%
[10,15)	2200	6%
≥15	2500	10%

（7）画出 if 嵌套中 3 种分支结构的流程图。

（8）执行下面的程序后，变量 y 的值是（　　）。

A. 1000　　B. 1200　　C. 1400　　D. 3400

```
#include<stdio.h>
int main()
{
    int x,y;
    x=7;
    if(x<6)
        y=1000;
    else
      if(x<=18)
            y=(x-6)*200+1000;
      else
            y=(x-18)*300+3400;
    printf("y=%d.\n",y);
    return 0;
}
```

（9）某市出租车的收费情况只有两种。一种是起步价为 7 元，车程 3 km；3 km 以上 1.2 元/km；6 km 以上要加收 50%的回空费，即 1.8 元/km。另一种是排气量在 1.8 L 以上的豪华型轿车，起步价 10 元，车程也是 3 km；3 km 以上 1.8 元/km；6 km 以上加收 50%的回空费，即 2.7 元/km。低速行驶及等待，每 5 min 按照 1 km 计费。编写程序完成以下要求，输入

乘坐的车型、行驶里程数和低速行驶时长，计算并输出应付的出租车费（四舍五入到元）。

（10）输入年、月数，求该年该月的天数。

（11）简单计算器模拟，输入两个整数和一个运算符，输出运算结果。

（12）定义如下的分段函数，输入 x 值编程实现 y 值的求解。

$$y=\begin{cases}-x^2 & x<-1\\ \sqrt{x}+2x-1 & x>5\\ x^2+3x+1 & \text{其他}\end{cases}$$

2. 提高部分

（1）输入一个正整数，不超过 1000，判断它是否是两位数（即大于或等于 10 且小于或等于 99）。

（2）键盘上按照“年月日”的格式输入年份、月和日期，运行程序后，判断这一天是这一年的第几天。

（3）编制程序判断输入的数值 n 能否被 3 整除。

（4）水仙花数是一个三位正整数，它的每个位的 3 次方之和恰好等于它本身，如 $153=1^3+5^3+3^3$。输入一个正整数 n（$100\leqslant n\leqslant 999$），编程判断该数是否为水仙花数，如果是输出 Yes，否则输出 No。

（5）小林同学在操场跑步，他总共跑了 x 秒，请编程将其拆解为小时、分钟、秒输出。

（6）某商场客户分为白金卡、金卡、银卡和普通用户。为了回馈广大顾客，现推出以下优惠活动：

① 白金卡会员，享受 7 折优惠；

② 金卡会员，享受 8 折优惠；

③ 银卡会员，享受 9 折优惠；

④ 普通用户，享受 9.5 折优惠。

根据用户的身份和购买商品的价值，编程给出应付金额。其中会员身份可用字母表示，例如，P 表示白金卡会员，G 表示金卡会员，S 表示银卡会员，其他字母表示普通用户。

（7）输入两个时间点（24 小时制），输出两个时间点之间的时间间隔，时间间隔用“时：分：秒”表示。如 3 时 5 分 25 秒应表示为 03:05:25。假设两个时间在同一天内，时间先后顺序与输入无关。

3.6 拓展阅读

计算机程序之母——艾达·洛芙莱斯（Ada Lovelace）

艾达·洛芙莱斯（Ada Lovelace，1815 年 12 月 10 日—1852 年 11 月 27 日，见图 3.7）是 19 世纪英国数学家查尔斯·巴贝奇（Charles Babbage）的合作伙伴，被认为是世界上第一个真正意义上的计算机程序员。她为了给程序设计“算法”，制作了第一份程序设计流程图，作为计算机程序的创始人，建立了循环和子程序等现代编程领域极为重要的概念。如果不是她，这世界上就没有程序员这个职业了！

洛芙莱斯是著名英国诗人拜伦之女，颜值很高，擅于思考，从小就在数学和科学方面展示出了非凡的天赋。她接受了一流的教育，包括数学、科学和音乐方面的培训。这些教育背景为她后来的成就打下了坚实的基础。

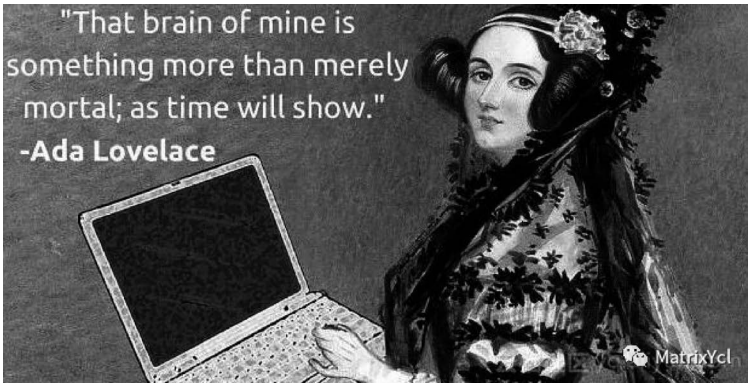

图 3.7　艾达·洛芙莱斯

天资聪颖的艾达之所以成为一名程序员还得从巴贝奇和他的分析机说起。出生于伦敦的艾达，17 岁时在剑桥大学第一次遇见了著名的数学家、发明家兼机械工程师查尔斯·巴贝奇，而这次相遇成了艾达人生的转折点。巴贝奇当时正在设计一种称为分析机的计算设备，而艾达则致力于为这台分析机编写算法。

艾达不仅理解了分析机的计算能力，还设想出一种方法来编写指令，以使机器能够进行一系列复杂的数学计算。她的想法包括使用循环、条件和递归等程序设计概念，这些概念在当时是前所未有的。

艾达在她的笔记中写下了一些关于分析机的见解，这些见解被认为是计算机科学史上的里程碑。她提出了一种概念，即分析机不仅可以进行数学计算，还可以应用于各种领域，甚至可以创造出音乐和艺术作品。这些预言性的见解揭示了计算机在未来可能发展的潜力。

尽管由于当时技术的局限性导致分析机最终无法实现，但是艾达在这个过程中提出的各种编程概念以及她对于计算的理解，对日后编程界产生了巨大的影响。从这一点上看，艾达当之无愧成为世界公认的第一位程序员。艾达的故事激励着许多人，特别是女性，鼓励她们在科学和技术领域追求自己的梦想。艾达是一个先驱者，她的贡献促使人们意识到计算机不仅是数学工具，还有更广泛的应用。她的故事证明了无论性别，只要有才华、热情和坚定的信念，每个人都能在科学和技术领域取得突破性的成就。

后来为了纪念艾达的杰出贡献，在 1980 年 12 月 10 日，美国国防部将一种新型的高级计算机编程语言命名为 Ada，以纪念艾达·洛夫莱斯。