

第5讲 选择结构程序设计





目录

- 选择结构程序设计
- if语句
- 关系运算符和关系表达式
- 逻辑运算符和逻辑表达式
- 条件运算符和条件表达式
- switch语句

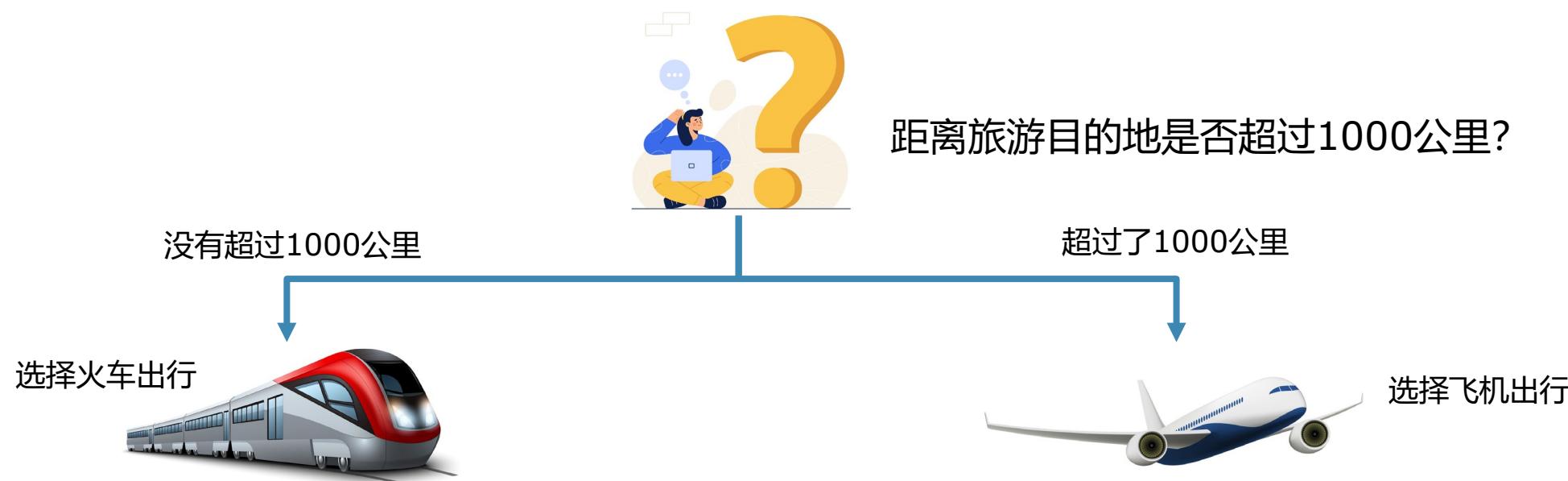
5.1 选择结构程序设计

选择结构程序设计

- 生活中处处存在着选择

- 根据旅游目的地的远近，决定搭乘火车还是飞机
- 根据超市购物金额多少，决定打折力度（例如购物超过300元打9折）
- 根据成绩是否达到60分，决定学生是否通过某一门课程
-

- 程序也需要根据条件，执行不同的程序分支



选择结构程序设计



```
#include <stdio.h>

int main(void)
{
    double distance;
    printf("请输入旅游目的地的距离 (单位: 公里) : ");
    scanf("%lf", &distance);
    if (distance <= 1000) { // if条件语句
        printf("选择火车出行! ");
    } else {
        printf("选择飞机出行! ");
    }
    return 0;
}
```

5.2 if语句

if语句的基本格式

if语句

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

表达式就是判断条件，一般是一个逻辑表达式，如果为真，则执行语句1，如果为假，则执行语句2。

else子句**不是必须的**，例如如果在不满足条件下不执行任何动作，就可以不写else子句。

只有一个语句时，可以不写花括号；如果超过一个语句，必须把这些语句用花括号括起来。

```
if (distance <= 1000)  
    printf("选择火车出行! ");  
else  
    printf("选择飞机出行! ");
```

```
if (distance <= 1000) {  
    printf("距离比较近! ");  
    printf("选择火车出行! ");  
} else {  
    printf("距离太远了! ");  
    printf("选择飞机出行! ");  
}
```

if语句的基本格式

例1：输入整数，求绝对值

```
#include <stdio.h>

int main()
{
    int x;
    scanf("%d", &x);
    if (x < 0)
        x = -x;
    printf("|x|=%d", x);
    return 0;
}
```

例2：求一元二次方程的实数根

```
#include <stdio.h>
#include <math.h>

int main(int argc, char const *argv[])
{
    double a, b, c;
    double x1, x2;
    double delta;
    printf("Please input a, b, c: ");
    scanf("%lf%lf%lf", &a, &b, &c);
    delta = b * b - 4 * a * c;
    if (delta >= 0) {
        x1 = (-b + sqrt(delta)) / (2 * a);
        x2 = (-b - sqrt(delta)) / (2 * a);
        printf("Root of the equation: x1=%.3lf, x2=%.3lf", x1, x2);
    } else {
        printf("The equation has no real roots!");
    }
    return 0;
}
```

if语句的基本格式

练习1

写一个程序，从键盘输入两个整数，输出两个数中较大的一个。

Sample Input:

```
10 20
```

Sample Output:

```
20
```

if语句的基本格式

处理多分支的if语句

```
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 (表达式N)  
    语句N;
```

有时，我们需要处理两种以上的情况，此时可以使用if-else if-else 的形式。

同样地，语句只有一句时，可以不写花括号；如果超过一句，必须把这些语句用一对**花括号**括起来（作为一个整体，称为复合语句）。

if语句的基本格式

练习2

键盘输入一个百分制的分数（分数是整数），输出该分数对应的等级（A、B、C、D或F）。

分数 \geq 90: A

80 \leq 分数 $<$ 90: B

70 \leq 分数 $<$ 80: C

60 \leq 分数 $<$ 70: D

分数 $<$ 60: F

Sample Input:

75

Sample Output:

C

提示：

要想表达 $80 \leq$ 分数 $<$ 90，可以这样写：`score >= 80 && score < 90`，`&&`称为逻辑与运算符。

if语句的嵌套

嵌套if语句

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

if语句可以嵌套使用，用于处理多级分支问题。

嵌套if语句涉及到if和else的配对问题。
C语言中，在嵌套的if语句里，else总是与它前面最近的、尚未配对的if配对（就近原则）。

if语句的基本格式

例3：某超市开展打折促销活动，活动规则是：

消费满300元，如果是超市会员，总消费金额打8.5折；如果不是超市会员，总消费金额打9折。
编写一个付款金额结算程序，输入消费金额，输出应付款金额。

```
#include <stdio.h>

int main()
{
    float amount, due;
    int vip = 0;
    printf("请输入是否是会员 (1表示会员, 0表示非会员) : ");
    scanf("%d", &vip);
    printf("请输入消费金额 (元) : ");
    scanf("%f", &amount);
    if (amount >= 300)
        if (vip == 1)
            due = amount * 0.85;
        else
            due = amount * 0.9;
    else
        due = amount;
    printf("应付款金额 (元) : %.2f", due);
    return 0;
}
```

if语句的基本格式

嵌套if语句都可以改写为非嵌套if语句，实践中具体采用哪种方式应具体问题具体分析，哪种逻辑更清晰就采用哪种写法。例如刚才的程序也可改写为非嵌套形式：

```
#include <stdio.h>

int main()
{
    float amount, due;
    int vip = 0;
    printf("请输入是否是会员: ");
    scanf("%d", &vip);
    printf("请输入消费金额(元) : ");
    scanf("%f", &amount);
    if (amount >= 300)
        if (vip == 1)
            due = amount * 0.85;
        else
            due = amount * 0.9;
    else
        due = amount;
    printf("应付款金额(元) : %.2f", due);
    return 0;
}
```



```
#include <stdio.h>

int main(int argc, char const *argv[])
{
    float amount, due;
    int vip = 0;
    printf("请输入是否是会员: ");
    scanf("%d", &vip);
    printf("请输入消费金额(元) : ");
    scanf("%f", &amount);
    if (amount >= 300 && vip == 1)
        due = amount * 0.85;
    else if (amount >= 300 && vip == 0)
        due = amount * 0.9;
    else
        due = amount;
    printf("应付款金额(元) : %.2f", due);
    return 0;
}
```

if语句的基本格式

练习3

键盘输入三个数，找出其中的最大值。

Sample Input:

```
10 20 15
```

Sample Output:

```
20
```

if语句的基本格式

练习4

输入三角形三边的边长，求三角形的面积。

提示：海伦-秦九韶公式： $S = \sqrt{p(p - a)(p - b)(p - c)}$ 其中 $p = \frac{a + b + c}{2}$

Sample Input:

3 4 5

Sample Output:

6.0

5.3 关系运算符和关系表达式

关系运算符

从刚才的案例中容易发现，要让if语句能够完成预期的功能，关键在于写对判断条件，而写对判断条件，除了正确分析问题之外，还必须正确使用**关系运算符**和**逻辑运算符**。



C语言支持的关系运算符包括：

优先级	关系运算符	说明
高	>	大于
	>=	大于等于
	<	小于
	<=	小于等于
低	==	等于
	!=	不等于

相同优先级的关系运算符，优先级不分先后，使用时按“**从左到右**”的结合性确定运算的顺序。

关系运算符

练习

试分析以下表达式中各个部分间的运算优先级：

c > a + b

a > b != c

a == b < c

a = b > c

算术运算符

<、<=、>、>=

==、!=

赋值运算符=

逗号运算符

优先级从高到低



注意：

相等关系运算符 == 不要与赋值运算符 = 混淆！！！

关系表达式

- 关系表达式就是使用了关系运算符、进行关系运算的表达式。

例如：

a > b a - b <= c - d 'a' > 'b' a > b == b < c

- 关系表达式的值是一个逻辑值：真 (true) 或假 (false) 。
- 在C语言中，一般用1 (或其他非零值) 表示真，用0表示假。

```
#include <stdio.h>

int main()
{
    printf("%d,%d\n", 3>2, 5==6);
    return 0;
}
```

输出为1,0

关系表达式

- 在C99之后的标准中规定，当引入<stdbool.h>头文件后，可以使用true和false分别代表真和假。
- 这里的true和false是通过简单的宏定义实现的，true实际上就是1，false实际上就是0。

```
#include <stdio.h>
#include <stdbool.h>

int main()
{
    printf("true: %d\n", true);
    printf("false: %d\n", false);
    return 0;
}
```

输出

```
true: 1
false: 0
```

5.4 逻辑运算符和逻辑表达式

逻辑运算符

逻辑运算符包括：

- 逻辑与 (AND) : A AND B, A、B都为真，表达式为真
- 逻辑或 (OR) : A OR B, A、B只要有一个为真，则为真
- 逻辑非 (NOT) : A为真，则NOT A为假



C语言中的逻辑运算符：

优先级	关系运算符	说明	举例
高	!	逻辑非 (NOT)	<code>! (a == b)</code>
中	<code>&&</code>	逻辑与 (AND)	<code>a > b && a > c</code>
低	<code> </code>	逻辑或 (OR)	<code>a > c b > c</code>

逻辑表达式

- 逻辑表达式就是使用了逻辑运算符、进行逻辑运算的表达式。
- 逻辑表达式的值是一个逻辑值：真 (true) 或假 (false)。
- 在C语言中，一般用1 (或其他非零值) 表示真，用0表示假。
- 字符以其ASCII码值参与运算。

练习

假设 $a=5$, $b=-2$, $c='A'$, 试分析以下逻辑表达式值为真 (1) 还是假 (0) :

a

3 && a

a - 2 && !(b + 3)

c > 2 || 6 <= 3

逻辑非 !

算术运算符

关系运算符

逻辑与 &&

逻辑或 ||

赋值运算符 =

逗号运算符

优先级从高到低

逻辑表达式的短路效应

短路效应1

对于：表达式1 `&&` 表达式2 `&&` 表达式3

只有当表达式1为真（非0）时才需要判断表达式2的值；

只有表达式1和2都为真（非零）时，才需要判断表达式3的值；

当表达式1为假（0），则发生短路，不执行表达式2和3，整个表达式值为0。

练习：

请给出printf输出的a、b、c、d的值。

```
#include <stdio.h>

int main()
{
    int a = 0, b = 1, c = 2, d;
    d = (a++) && (b++) && (++c);
    printf("a=%d, b=%d, c=%d, d=%d\n", a, b, c, d);
    return 0;
}
```

逻辑表达式的短路效应

短路效应2

对于：表达式1 || 表达式2 || 表达式3

只有表达式1为假（0）时才需要判断表达式2的值；

只有表达式1和2都为假（0）时，才需要判断表达式3的值；

当表达式1为真（非0），则发生短路，不执行表达式2和3，整个表达式值为1。

练习：

请给出printf输出的a、b、c、d的值。

```
#include <stdio.h>

int main()
{
    int a = 0, b = 1, c = 2, d;
    d = (++a) || (b++) || (--c);
    printf("a=%d, b=%d, c=%d, d=%d\n", a, b, c, d);
    return 0;
}
```

逻辑表达式

练习5

键盘输入一个年份，输出该年份是否是闰年。

闰年的判断方法是：

- 当某一年份不能被100整除时，如果该年份能被4整除，则该年份是闰年
- 当某一年份能被100整除时，如果该年份能被400整除，则该年份是闰年

Sample Input:

1900

Sample Output:

NO

5.5 条件运算符和条件表达式

条件运算符和条件表达式

条件表达式的形式是：

表达式1 ? 表达式2 : 表达式3

条件表达式可以视为if语句的一种简化写法。

说明：

- 条件表达式根据表达式1的值确定整个表达式的值：
 - 当表达式1为真时，整个表达式的值为表达式2的值；
 - 当表达式1为假时，整个表达式的值为表达式3的值；
- 条件运算符的优先级高于赋值运算符，低于逻辑运算符；
- 条件运算符的结合性是从右到左。

条件运算符和条件表达式

条件表达式的形式是：

表达式1 ? 表达式2 : 表达式3

条件表达式可以视为if语句的一种简化写法。

```
#include <stdio.h>

int main()
{
    int a, b, max;
    scanf("%d%d", &a, &b);
    if (a > b)
        max = a;
    else
        max = b;
    printf("Max: %d", max);
    return 0;
}
```



```
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int a, b, max;
    scanf("%d%d", &a, &b);
    max = a > b ? a : b;
    printf("Max: %d", max);
    return 0;
}
```

5.6 switch语句

switch语句的基本格式

switch语句

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

switch语句根据表达式的选择某一个case后面的所有语句组执行（直至switch末尾或者遇到了break），如果所有case都不符合，则执行default后面的语句组。

default类似于if语句中的else，如果不需
要可以不写default这个分支。

常量表达式，可以是除浮点类型之外的任意
类型（之所以不能是浮点类型，是因为浮
点类型不能进行精确比较）。常用的类型是整
型和字符型。常量表达式中不能有变量，即
使变量已经赋值。

switch语句的基本格式

例：用switch语句写一个程序，输入某天是星期几，输出当天是工作日还是休息日。

```
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int day = 0;
    printf("请输入今天是星期几 (1~7) : ");
    scanf("%d", &day);
    switch (day) {
        case 1:
        case 2:
        case 3:
        case 4:
        case 5: printf("今天是工作日! \n");
        case 6:
        case 7: printf("今天是休息日! \n");
    }
    return 0;
}
```

请输入今天是星期几 (1~7) : 2
今天是工作日!
今天是休息日!



switch语句的基本格式

例：用switch语句写一个程序，输入某天是星期几，输出当天是工作日还是休息日。

```
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int day = 0;
    printf("请输入今天是星期几 (1~7) : ");
    scanf("%d", &day);
    switch (day) {
        case 1:
        case 2:
        case 3:
        case 4:
        case 5: printf("今天是工作日! \n"); break;
        case 6:
        case 7: printf("今天是休息日! \n");
    }
    return 0;
}
```

切记，switch语句具有“穿透执行”的特性，符合条件的case是执行的起点，而不是说仅执行这个case的语句。所以如果不想继续执行下去，一定要加上break！

switch语句的应用

练习6

键盘输入年、月、日，输出这个日期是一年中的第几天。

Sample Input:

```
2024 3 1
```

Sample Output:

```
61
```

switch语句的应用

练习6参考答案（充分利用switch的穿透执行特性）

```
#include <stdio.h>

int main()
{
    int year, month, day;
    int s = 0;
    scanf("%d%d%d", &year, &month, &day);
    switch (month) {
        case 12: s+=30;
        case 11: s+=31;
        case 10: s+=30;
        case 9: s+=31;
        case 8: s+=31;
        case 7: s+=30;
        case 6: s+=31;
        case 5: s+=30;
        case 4: s+=31;
        case 3: s+=28;
        case 2: s+=31;
        case 1: s+=day;
    }
    if ((year % 4 == 0 && year % 100 != 0 || year % 400 == 0) && month >= 3)
        s++;
    printf("%d\n", s);
    return 0;
}
```