

选择执行

II、选择执行

定义：有些代码可能执行，也可能不执行，有选择的执行某些代码

分类：①：if

1、if 最简单的用法

格式：

If (条件表达式)

语句

功能：

如果表达式为真，执行语句；

如果表达式为假，语句不执行。

if最简单的用法.cpp

```
1  #include<stdio.h>
2  int main(void)
3  {
4      if (3 > 2)
5          printf("AAAA\n");
6
7      return 0;
8  }
```

```
# include <stdio.h>

int main(void)
{
    if (3)
        printf("AAAA\n"); //会输出

    if (0)
        printf("BBBB\n"); //不会输出

    if (0 == 0)
        printf("CCCC\n"); //会输出

    return 0;
}
```

2、If 的范围问题【重点】

①：If (条件表达式)

语句 A;

语句 B;

解释：if 默认只能控制语句 A 的执行或不执行，if 无法控制语句 B 的执行或不执行（语句 B 一定会执行）。

```

1  #include <stdio.h>
2  int main(void)
3  {
4      if (1 > 2)
5          printf("AAAA\n");
6          printf("BBBB\n");//不属于if语句, if不成立, 依旧执行。|
7
8      return 0;
9  }
10 /*
11     在dev c++中的输出结果是:
12     -----
13     BBBB
14     -----
15     总结:
16     if默认的只能控制一个语句的执行或不执行
17 */

```

②: If (条件表达式)

```

{
    语句 A;
    语句 B;
}

```

此时 if 可以控制语句 A 和语句 B。

由此可见: if 默认只能控制一个语句的执行或不执行, 如果想控制多个语句的执行或不执行, 就必须把这些语句放在 {} 中。(把多个语句用 {} 括起来叫复合语句)

3、If...else... 的用法

if最简单的用法.cpp if的范围问题.cpp [*] if...else...的用法.cpp

```

1  #include <stdio.h>
2  int main(void)
3  {
4      int i, j;
5      scanf("%d %d", &i, &j);
6      if (i > j)
7          printf("i大于j\n");
8      else
9          printf("i小于等于j\n");
10
11     return 0;
12 }

```

4、If...else if...else...的用法

if...else if...else...的用法.cpp

```
1  #include <stdio.h>
2  int main(void)
3  {
4      double delta;
5      scanf("%d", &delta);
6      if (delta > 0)
7          printf("有两个解\n");
8      else if (delta == 0)
9          printf("有一个解\n");
10     else
11         printf("无解\n");
12
13     return 0;
14 }
15 /*
16 if(1)
17     A;
18 else if(2)
19     B;
20 else if(3)
21     C;
22 else(4)
23     D;
24
25 解释: 1成立, A执行;
26        如果1不成立, 2成立, 执行B;
27        如果1、2不成立, 3成立, 执行C; |
28        如果1、2、3不成立, 执行D。
29 */
```

```
/*
一定要明白该程序为何出错。
*/

#include <stdio.h>
int main(void)
{
    double delta;
    scanf("%d", &delta);

    if (delta > 0)
        printf("有两个解\n");
        printf("haha!\n");

    else if (delta == 0) //else前面没有对应的if。
        printf("有一个解\n"); |
    else
        printf("无解\n");

    return 0;
}
```

5、c 语言对真假的处理

非零是真；零就是假；

真用 1 表示；假用 0 表示。

6、If 举例--求分数的等级

```
1 #include <stdio.h>
2 int main(void)
3 {
4     float score;//score:分数;
5
6     printf("请输入你的考试成绩:");
7     scanf("%f", &score);
8
9     if (score > 100)
10         printf("做梦!");
11     else if (score >= 90 && score <= 100)//不能写成90<=score<=100, 测试详看<逻辑表达式最终值测试>。
12         printf("优秀!");
13     else if (score >= 80 && score < 90)
14         printf("良好!");
15     else if (score >= 60 && score < 80)
16         printf("及格!");
17     else if (score >= 0 && score < 60)
18         printf("不及格!继续努力");
19     else
20         printf("输入的分数过低, 不要如此自卑!");
21
22     return 0;
23 }
```

《逻辑表达式最终值的测试》

```
1 #include <stdio.h>
2 int main(void)
3 {
4     float score = -1000;
5
6     int i = (90 <= score <= 100);//逻辑运算符, 90 <= score 只有两个值 (0/1), 0/1都小于100, 所以这个式子最后的都为真, 值为1。
7     printf("%d\n", i);
8
9     return 0;
10 }
11 /*
12 在dev c++中的输出结果是:
13 -----
14 1
15 -----
16 */
```

《互换两个数字》

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int i = 3;
5     int j = 4;
6     int t;
7
8     /*
9     以下两行代码无法完成i和j的互换。
10    i = j; //将j的值赋给i, i=4, j=4;
11    j = i; //j=4;
12    printf("i = %d, j = %d", i, j);
13    */
14    /*正确写法, 一般临时变量使用t (增加一个容器)*/
15    t = i;
16    i = j;
17    j = t;
18    printf("i = %d, j = %d\n", i, j);
19
20    return 0;
21 }
```

《求三个数字的最大值》

```
1  #include <stdio.h> //参考课本p64
2  int main(void)
3  {
4      int a, b, c; //等价于int a; int b; int c;
5
6      printf("请输入三个整数, (中间以空格分隔): ");
7      scanf("%d %d %d", &a, &b, &c);
8
9      //编写代码 a是最大值, b是中间值, c是最小值。
10     //程序 = 算法+语言
11
12     /*
13     如果a>b, 将a与c比较, 如果a>c, 最大值是a, 如果a!>c, 最大值是c。
14     如果a!>b, 将b与c比较, 如果b>c, 最大值是b, 如果b!>c, 最大值是c。
15     */
16     if (a > b)
17     {
18         if (a > c)
19             printf("最大值是a=%d", a);
20         else
21             printf("最大值是c=%d", c);
22     }
23     else
24     {
25         if (b > c)
26             printf("最大值是b=%d", b);
27         else
28             printf("最大值是c=%d", c);
29     }
30
31     return 0;
32 }
```

《三个数字的排序》

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int a, b, c;
5      int t;
6
7      printf("请输入三个数字(中间用空格隔开):");
8      scanf("%d %d %d", &a, &b, &c);
9
10     if (a < b)
11     {
12         t = a;
13         a = b;
14         b = t;
15     }
16
17     if (a < c)
18     {
19         t = a;
20         a = c;
21         c = t;
22     }
23
24     if (b < c)
25     {
26         t = b;
27         b = c;
28         c = t;
29     }
30     printf("%d %d %d\n", a, b, c);
31     return 0;
32 }
```


7、If 的常见问题解析

①：空语句的问题：

If (3 > 2);

等价于

If (3 > 2)

; //这是一个空语句。

```
1  /*
2      空语句
3  */
4  #include <stdio.h>
5  int main(void)
6  {
7      if (1 > 2); //; -- 空语句
8      printf("AAAA\n");
9      printf("BBBB\n");
10
11     return 0;
12 }
```

②：加分号的问题：

If (3 > 2); //else 处会报错，因为：是个空语句，else 无匹配的 if

Printf("haha!\n");

Else

Printf("heihei!\n");

```
1  #include <stdio.h>
2  int main(void)
3  {
4      if (3 > 2) //如果这里加分号，则会导致6行报错：else无匹配的if
5          printf("哈哈! \n");
6      else
7          printf("嘿嘿! \n");
8
9      return 0;
10 }
```

③: if (表达式 1)

A;

Else if (表达式 2)

B;

else

C;

//即使表达式 1, 表达式 2 都成立, 也只执行 A 语句

```
1 #include <stdio.h>
2 int main(void)
3 {
4     if (3 > 1)
5         printf("A\n");
6     else if (3 > 2)
7         printf("B\n");
8     else
9         printf("C\n");
10
11     return 0;
12 }
13
14 /*
15 在dev c++中的输出结果是:
16 -----
17 A
18 -----
19
20 总结
21 if (表达式1)
22     A;
23 else if (表达式2)
24     B;
25 else
26     C;
27
28 即使表达式1, 表达式2都成立;
29 也只会执行A语句。
30 */
```

④: if (表达式 1)

A;

Else if (表达式 2)

B;

Else If (表达式 3)

C;

//这样语法上不会出错, 但逻辑上有漏洞。

⑤: if (表达式 1)

A;

Else if (表达式 2)

B;

Else If (表达式 3)

C;

Else (表达式 4) //7 行

D;

//这样写是不对的, 正确的写法是: 要么去掉 7 行的 (表达式 4); 要么在 7 行的 else 后面加上 if。

⑥:

```
if (表达式1)
    A;
else if (表达式2)
    B;
else if (表达式3)
    C;
else (表达式4);
    D;
```

//这样写在语法上不会出错，但逻辑上错误的。

```
Else (表达式);
    D;
等价于:
Else
    (表达式4);
    D;
```

switch

格式:

```
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 n: 语句 n;
    default: 语句 n+1;
}
```

第一：所有 **case** 后面的常量表达式为便于描述，我们姑且称之为标签，这些标签只能是：①：枚举常量（有些书称之为枚举元素）、②：数值常量、③：字符常量、④：常变量、⑤：宏名中的一种，注意普通变量，枚举变量是不能作为标签使用的。

第二：switch 后面括号里的“表达式”允许是任何类型的。

第三：执行完一个 **case** 语句后，流程控制就转移到下一个 **case** 语句继续执行。“**case**”常量表达式只是起语句标号的作用，并不是在该处进行条件判断。再执行 **switch** 语句时，根据 **switch()** 中表达式的值找到与之匹配的 **case** 子句，从此 **case** 子句开始执行下去，不再进行判断。

第四：switch 是选择不是循环，如果在 switch 中出现了 **break** 语句。该语句的功能知识退出 switch 语句转去执行它下面的语句。在 switch 中出现 **continue** 是错误的，除非 switch 本身就属于 **for** 或 **while** 循环的一部分。

能看懂电梯程序就可以。