



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

编程实现选择分支（2）

- ◆ 1、程序顺序结构
- ◆ 2、程序选择结构

3.4.1 if语句



【例3.3】

利用下面的Heron公式计算三角形面积。

设三角形的三边长 a 、 b 、 c ，构成三角形的条件是： $a + b > c$ ， $b + c > a$ ，

$c + a > b$ 同时成立。则面积： $s = \sqrt{t(t - a)(t - b)(t - c)}$

其中 $t = \frac{a + b + c}{2}$ 。

3.4.1 if语句

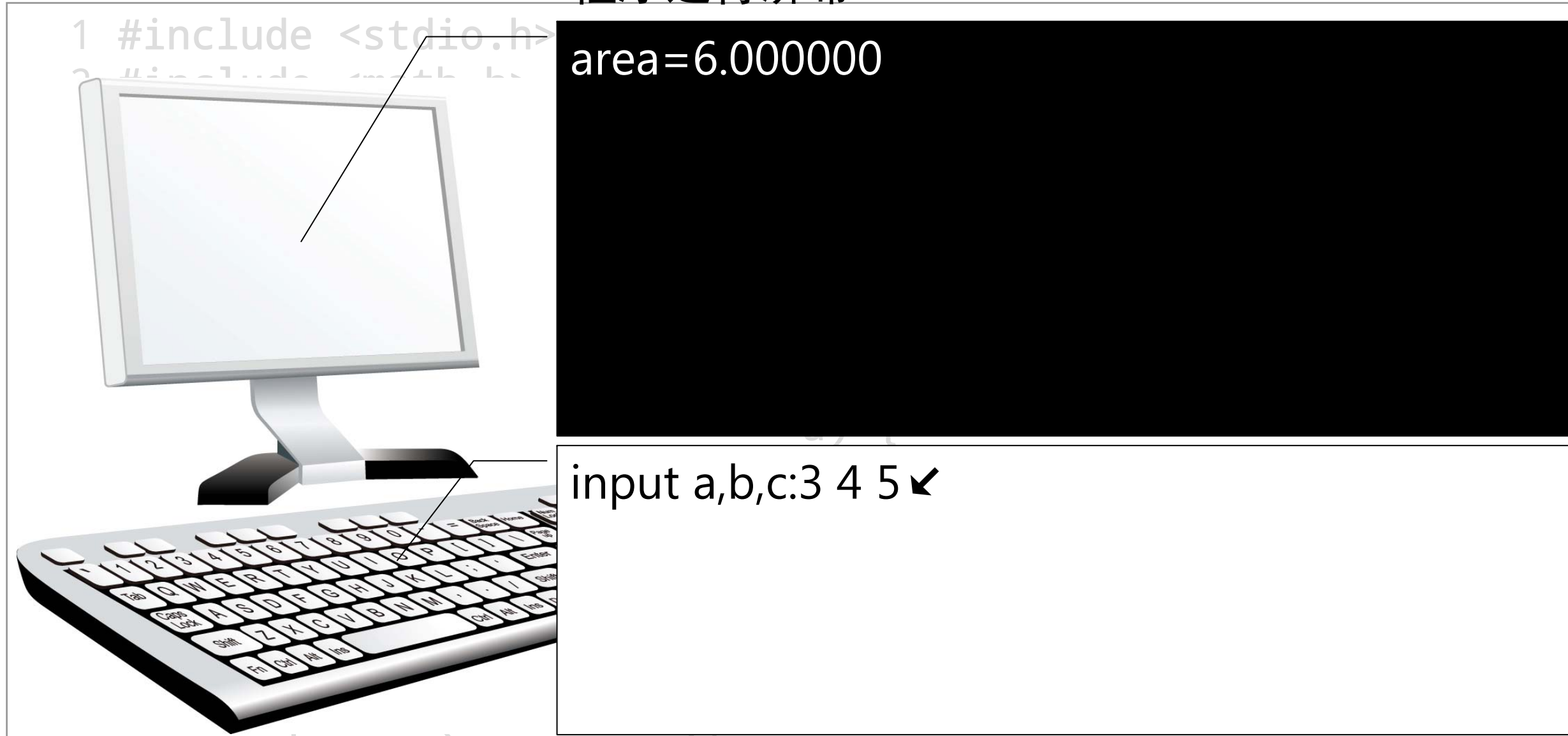
例3.3

```
1 #include <stdio.h>
2 #include <math.h>
3 int main()
4 {
5     double a,b,c;
6     printf("input a,b,c:");
7     scanf("%lf%lf%lf",&a,&b,&c); //输入三角形三边长
8     //判断三边长是否构成三角形
9     if (a+b>c && a+c>b && b+c>a) {
10         double s,t;
11         t=(a+b+c)/2.0;
12         s=sqrt(t*(t-a)*(t-b)*(t-c)); //Heron公式计算三角形面积
13         printf("area=%lf\n",s);
14     }
15     else printf("error\n");
```

3.4.1 if语句

例3.3

程序运行屏幕



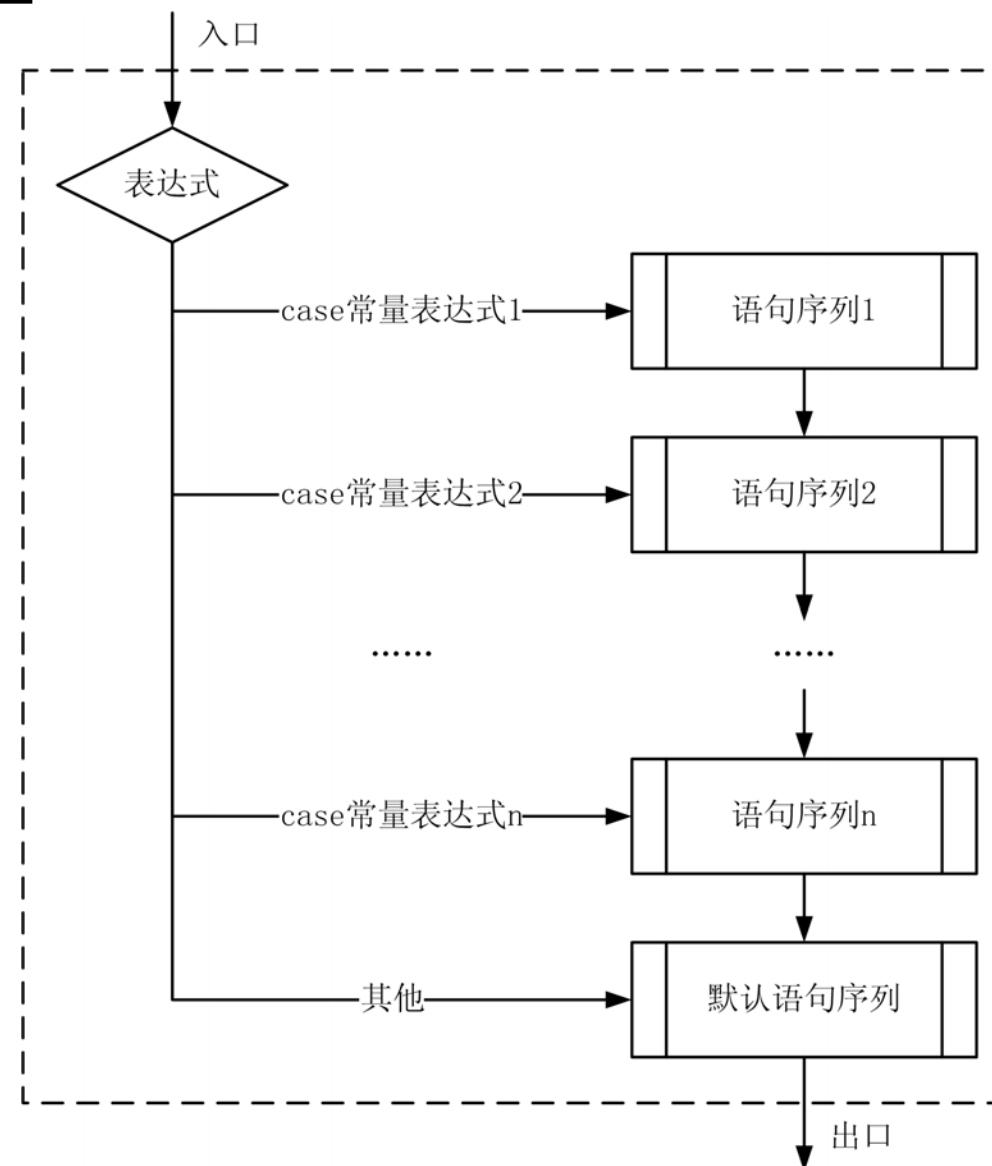
3.4.2 switch语句

- ▶ switch语句的作用是计算给定的表达式，根据结果选择从多个分支入口执行，语句形式为：

```
switch (表达式) {  
    case 常量表达式1 : ..... 语句序列1  
    case 常量表达式2 : ..... 语句序列2  
        ⋮  
    case 常量表达式n : ..... 语句序列n  
    default           : ..... 默认语句序列  
}
```

3.4.2 switch语句

图3.3 switch语句执行流程



3.4.2 switch语句

- ▶ (1) switch语句中case分支的语句序列可以是一个语句，也可以是任意多的语句序列，也可以没有语句；既可以是简单语句，又可以是复合语句和控制语句。
- ▶ 如果没有语句，则一旦执行到这个case分支，什么也不做，继续往下执行。

3.4.2 switch语句

举例

```
switch (n) {  
    case 7 : printf("step5\n");  
    case 6 :  
    case 5 : printf("step4\n");  
    case 4 :  
        {  
            printf("step3\n");  
            printf("step2\n");  
        }  
    case 2 : printf("step1\n");  
    default: printf("step0\n");  
}
```

3.4.2 switch语句

- ▶ (2) switch语法中各个case分支和default分支的出现次序在语法上没有规定，但次序的不同安排会影响执行结果。

3.4.2 switch语句

举例

```
1 //①程序A
2 switch (n) {
3     case 1 : printf("1");
4     case 2 : printf("2");
5     default: printf("0");
6 }
```

```
//②程序B
switch (n) {
    default: printf("0");
    case 1 : printf("1");
    case 2 : printf("2");
}
```

n为3时，程序A输出0，程序B输出012，因为程序A只执行第5行，而程序B依次执行第3、4、5行。

3.4.2 switch语句

- ▶ (3) switch语法中default分支是可选的，若没有default分支且没有任何case标号的值相等时，switch语句将什么也不做，直接执行后续语句。

3.4.2 switch语句

- ▶ (4) switch语句后面的圆括号是语法规定必须有的。分支选择可以是C语言的任意表达式，但其值必须是整数（含字符类型）、枚举类型，或者包含能转换成这两种类型的类型。如果有其他类型的值，例如浮点数或逻辑型结果，则会产生隐式类型转换，如果不能隐式类型转换，则出现语法错误。通常情况下，分支选择是整型的算术运算表达式，应该谨慎出现别的表达式。

```
switch ( 整数/字符类型/枚举类型 ) {  
    :  
}
```

3.4.2 switch语句

举例

```
switch (k<=12 || k>=65)
{
    case 0: printf("false\n");
    case 1: printf("true\n");
}
```

分支选择为逻辑运算表达式，按整数处理，真为1，假为0，除此之外的其他case分支是不可能转向去的。

3.4.2 switch语句

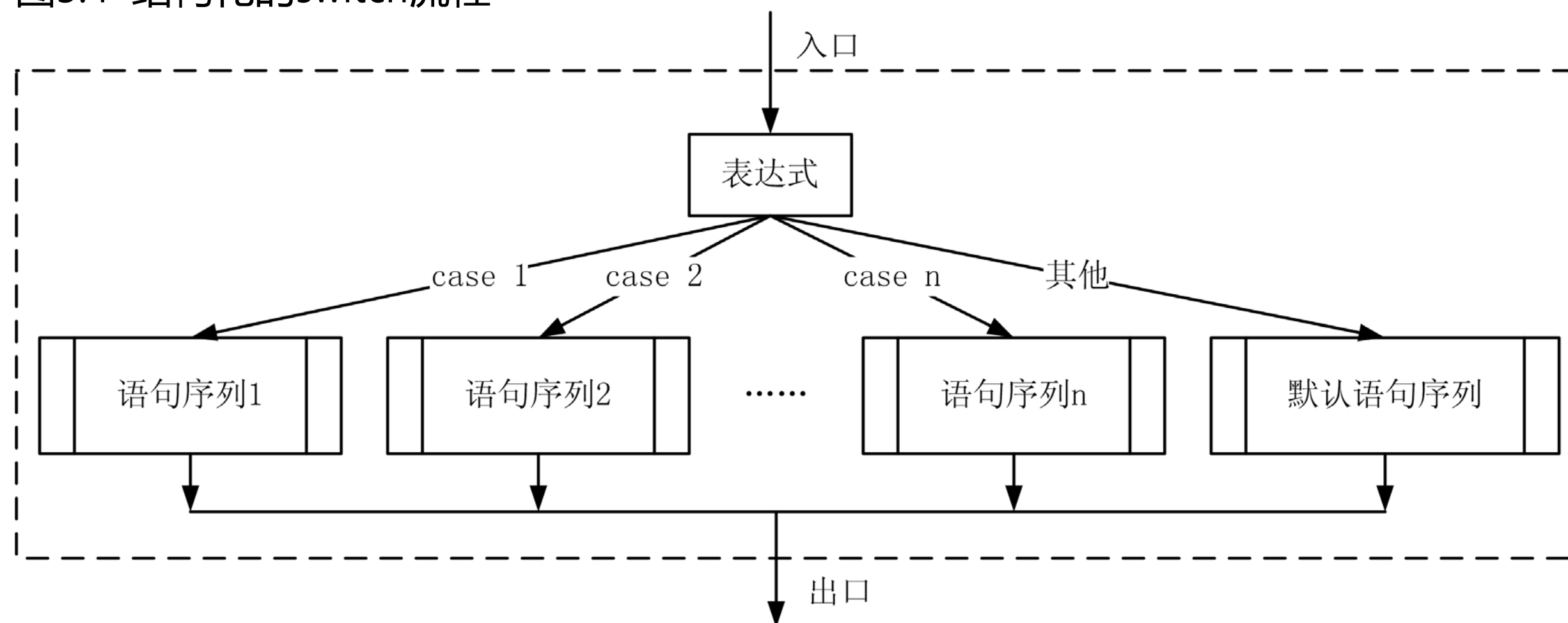
- ▶ (5) switch语法中的case分支必须是常量表达式且互不相同，即为整型、字符型、枚举类型的常量值，但不能包含变量。例如若c是变量，则“case c>='a' && c<='z':”的写法是错的。case分支后面的冒号是必须的，即使没有后面的语句序列。

3.4.2 switch语句

- ▶ 从switch语句的语法上来看，一旦开始执行某个分支，就会一直执行到没有语句为止；换言之，从某个分支开始执行，也会将另一个分支的语句序列执行到。
- ▶ 实际问题求解中，多分支选择却不是这样的，它往往要求某个分支的语句执行后，switch就结束，所要求的执行流程如图所示。

3.4.2 switch语句

图3.4 结构化的switch流程



3.4.2 switch语句

- ▶ 为了实现这样的控制流程，可以使用break语句，语句形式为：

```
break;
```

- ▶ 在switch语句中任意位置上，只要执行到break语句，就结束switch语句的执行，转到后续语句。

3.4.2 switch语句

- ▶ 所以，更常见的switch结构应该如下，它提供了程序多分支选择执行流程。

```
switch (表达式) {  
    case 常量表达式1 : .....语句序列1; break;  
    case 常量表达式2 : .....语句序列2; break;  
    :  
    case 常量表达式n : .....语句序列n; break;  
    default           : .....默认语句序列  
}
```

- ▶ 最后分支是可以不用break语句的。

CP 程序设计