



switch 选择结构

应用条件语句可以很方便地使程序实现分支。分支比较多时，虽然可以用嵌套的 if 语句来解决，但是程序结构会显得复杂凌乱。为方便实现多分支选择，C++ 提供了 switch 开关语句。



第一个 switch 结构程序

下面就是一个典型的 switch 结构程序。同学们可以看看是否能够读懂。

```
1. #include <iostream>
2. using namespace std;
3. int main ()
4. {
5.     char grade;
6.     cin >> grade;
7.     switch(grade)
8.     {
9.         case 'A' :
10.             cout << "优秀" << endl;
11.             break;
12.         case 'B' :
13.             cout << "良好" << endl;
14.             break;
15.         case 'C' :
16.             cout << "通过" << endl;
17.             break;
18.         case 'D' :
19.             cout << "不及格" << endl;
20.             break;
21.         default :
22.             cout << "缺考" << endl;
23.             break;
24.     }
25.     return 0;
26. }
```



switch 语句的语法格式及执行过程

switch 语句的语法格式如下所示：

```
1. switch(表达式)
2. {
3.     case 常量表达式 1:
4.         语句序列 1;
5.         break; //可选的
6.     case 常量表达式 2:
7.         语句序列 2;
8.         break; //可选的
9.     .....
10.    case 常量表达式 n:
11.        语句序列 n;
12.        break; //可选的
13.    default: //可选的
14.        语句序列 n+1;
15.        break; //可选的
16. }
```

switch 语句执行过程如下：

- ① 计算出 switch 后面圆括号内表达式的值，假定为 M。
- ② 依次计算出每个 case 后常量表达式的值，假定为 M1、M2、…。
- ③ M 依次同 M1、M2、…进行比较，一旦遇到 M 与某个值相等，则就从对应标号的语句开始执行，直到遇到 break 语句为止。
- ④ 不存在相等的情况下，若存在 default 子句，则执行其后的语句序列，否则不执行任何操作。

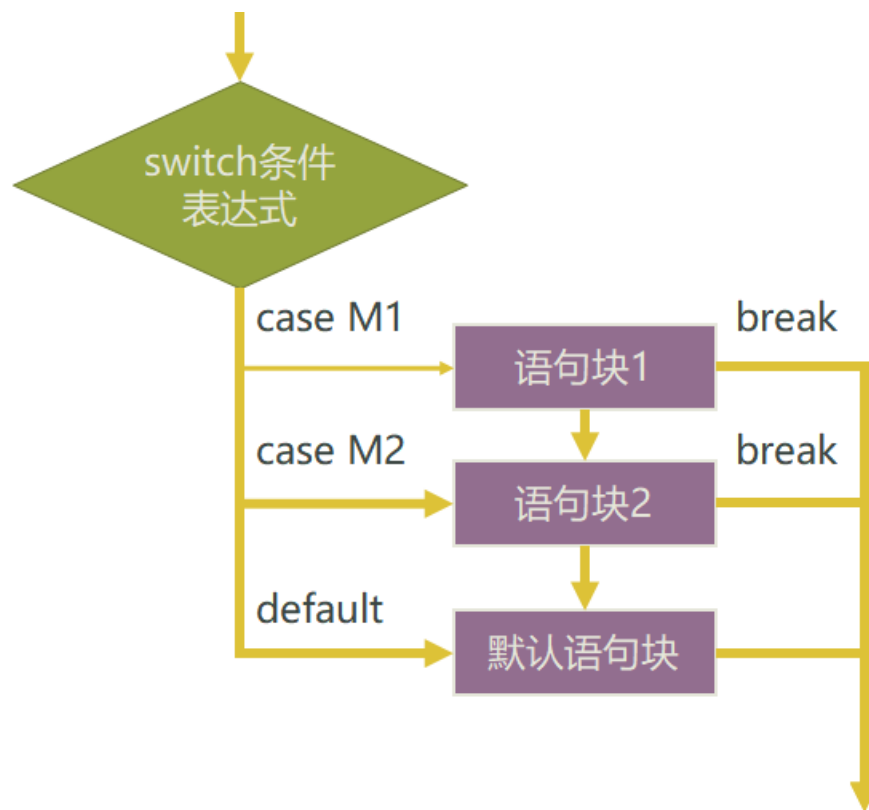


图1. switch 多分支结构流程图



switch 语句的注意事项

在整个 switch 语句中，可以使用一次或多次 case 标号，只能使用零次或一次 default 标号。

下面这种写法是对的。

```
1. switch(a)
2. {
3.     case 1:
4.         ...
5.         break;
6.     case 2:
7.         ...
8.         break;
9. }
```

下面这种写法是错的。

```
1. switch(a)
2. {
3.     case 1:
4.         ...
5.         break;
6.     case 2:
7.         ...
8.         break;
9.     default:
10.        ...
11.    default:
12.        ...
13. }
```

在同一个语句序列的前面可以有多个 case 标号。

下面这种写法是对的。

```
1. switch(a)
2. {
3.     case 1:
4.         ...
5.         break;
6.     case 2:
```



```
7. case 3:
8.     ...
9.     break;
10. }
```

语句标号后面需要是结果可以用整数表示的常量或常量表达式。

下面这种写法是对的。

```
1. switch(a)
2. {
3.     case 1:
4.         ...
5.         break;
6.     case 2+3:
7.         ...
8.         break;
9.     case 'a':
10.        ...
11.        break;
12. }
```

下面这种写法是错的。

```
1. int x, y;
2. switch(a)
3. {
4.     case x+y:
5.         ...
6.         break;
7.     case 3.5:
8.         ...
9.         break;
10.    case "a":
11.        ...
12.        break;
13. }
```

使用 switch 语句时，需要注意以下几点：

- case 语句后的各常量表达式的值不能相同，否则会出现错误。
- 每个 case 或 default 后，可以包含多条语句，不需要使用“{”和“}”括起来。



- 在包含 break 的情况下，各 case 和 default 子句的先后顺序可以变动，不会影响程序执行结果。
- default 子句可以省略，default 子句放在最后时可以省略 break。



switch 语句的程序设计风格

switch（表达式）单独一行。

各 case 分支和 default 分支要缩进并对齐。

分支处理语句要相对再缩进，以体现不同层次的结构。

```
1. switch(a)
2. {
3.     case 1:
4.         x++;
5.         break;
6.     case 2:
7.         y++;
8.         break;
9.     case 3:
10.        z++;
11.        break;
12.    default:
13.        cout<<"error";
14. }
```

每个 case 后面的语句可以写在冒号后的同一行或换到下行写，因此下面的程序是和上面的程序等价的。

```
1. switch(a)
2. {
3.     case 1:x++;break;
4.     case 2:y++;break;
5.     case 3:z++;break;
6.     default :cout<<"error";
7. }
```




学生成绩等级判断

【问题描述】

已知学生成绩的等级划分，通过程序实现输入学生等级显示出分数范围：

- 如果是“A”级，显示“85~100”；
- 如果是“B”级，显示“70~84”；
- 如果是“C”级，显示“60~69”；
- 如果是“D”级，显示“0~60”；
- 否则，显示“error”。

【输入样例】

A

【输出样例】

85~100

【参考程序】

```
#include<iostream>
using namespace std;
int main()
{
    char grade;
    cin >> grade;
    switch (grade)
    {
        case 'A':
            cout << "85~100";
            break;
        case 'B':
            cout << "70~84";
            break;
        case 'C':
            cout << "60~69";
            break;
```



```
        case 'D':  
            cout << "0~60";  
            break;  
        default:  
            cout << "error";  
    }  
    return 0;  
}
```



星期几的英文名称

【问题描述】

输入表示星期几的数字，输出对应的英文名称。

【输入样例】

1

【输出样例】

Monday

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int weekday;
6.     cin >> weekday;
7.     switch(weekday)
8.     {
9.         case 1:
10.            cout << "Monday" << endl;
11.            break;
12.        case 2:
13.            cout << "Tuesday" << endl;
14.            break;
15.        case 3:
16.            cout << "Wednesday" << endl;
17.            break;
18.        case 4:
19.            cout << "Thursday" << endl;
20.            break;
21.        case 5:
22.            cout << "Friday" << endl;
23.            break;
24.        case 6:
25.            cout << "Saturday" << endl;
26.            break;
27.        case 7:
```



```
28.         cout << "Sunday" << endl;
29.         break;
30.     default:
31.         cout << "input error!";
32.     }
33.     return 0;
34. }
```