分支选择结构(if, else, switch, continue, break)

1. 为什么要使用选择结构

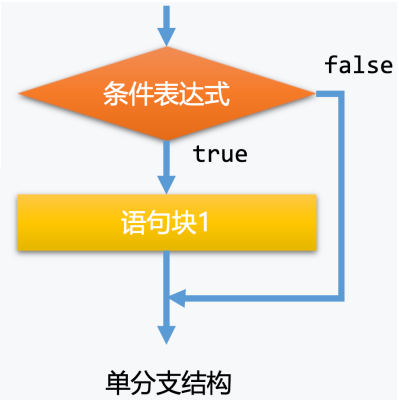
有的功能只有顺序结构的话,程序执行不了

2. 关于选择结构中,保留的关键字有哪些?关键字只有小写

‘if, else , switch, case, break, default

3. 单分支语句

```
1. if (条件表达式)
2. {
3.     ……
4. }
```

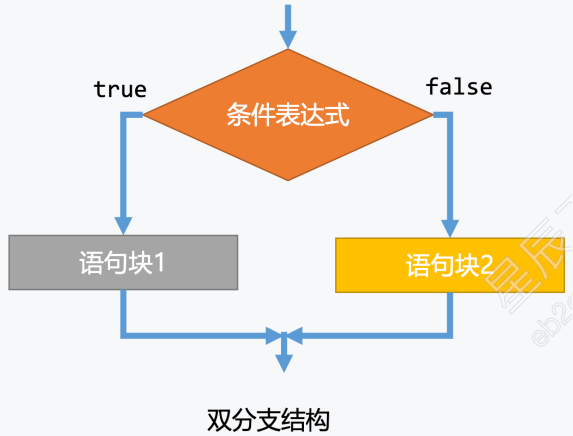


条件表达式用括号包起来
大括号应该对齐,语句快中的语句应该缩进对齐
如果语句块中只有一个语句,那么可以不用写大括号

```
1. if (x > y)
2.     cout << x;
```

- 初学者容易犯的错误!!!
  注意 == 和 = 的区别, ==是判断是否相等 =是赋值运算符
- 判断偶数奇数
  ‘a % 2 ==0 / a % 2 ==1

4. 双分支语句



双分支结构

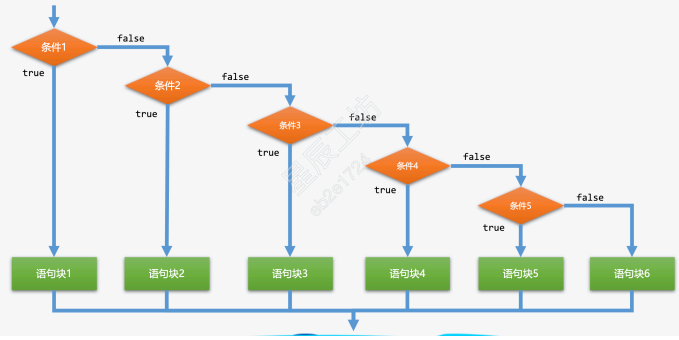
```
1.  if (条件表达式)
2.  {
3.      //条件表达式成立时被执行
4.  }
5.  else
6.  {
7.      //条件表达式不成立时被执行
8.  }
```

If-else 语句, if else 要对齐

同样的语句块中只有一个语句的时候

```
1.  if (x > y)
2.      cout << x;
3.  else
4.      cout << y;
```

5. 多分支语句




```
1.  if (条件表达式1)
2.  {
3.      ……
4.  }
5.  else if (条件表达式2)
6.  {
7.      ……
8.  }
9.  else if (条件表达式3)
10. {
11.     ……
12. }
13. ……
14. else
15. {
16.     ……
17. }
```

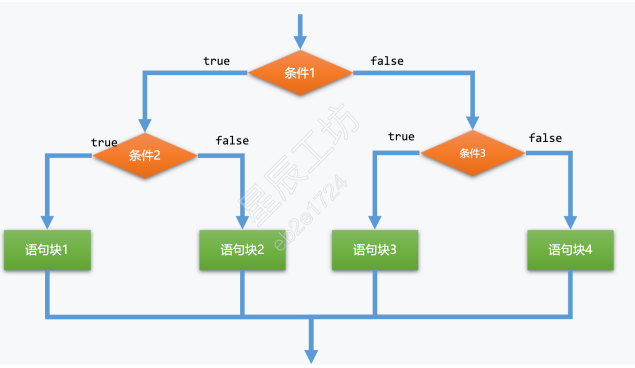
6. If 语句的特点

- if、else if和else后面都没有分号。
- if、else if后面有表达式（一般为逻辑表达式或关系表达式）。
- else后面没有表达式。
- 一个if语句中有且只有1个if，必须在else if和else之前。
- 一个if语句后可跟一个可选的else if...else语句，这可用于测试多种条件。
- 一个if语句中，有0个或者多个else if，必须在if之后，else之前。
- 一个if语句中，有0个或者1个else，必须在if和所有的else if之后。
- 一旦某个else if匹配成功，其他的else if或else将不会被测试。

7. 运算符的优先级

运算符	符号
逻辑非运算符	!
算数运算符	+、-、*、/、%
关系运算符	>、>=、<、<=、==、!=
逻辑与运算符	&&
逻辑或运算符	||
赋值运算符	=、+=、-=、*=、/=、%=

8. 选择结构的嵌套

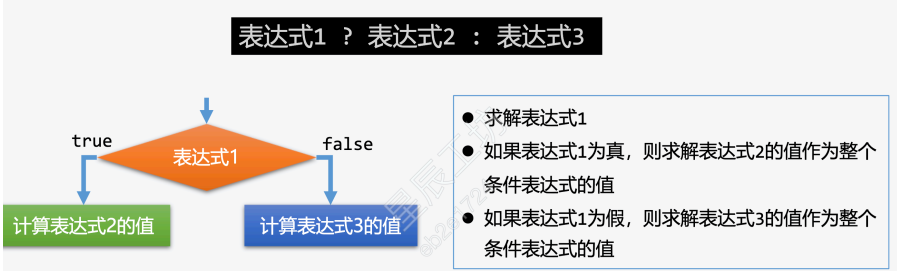



```
if(条件表达式1)
{
    ……
    if(条件表达式2)
    {
        ……
    }
    else
    {
        ……
    }
}
else
{
    ……
}
```

每一级中的 if else if else 都要各自对齐

每一级中各分支语句都要缩进

9. 条件运算符

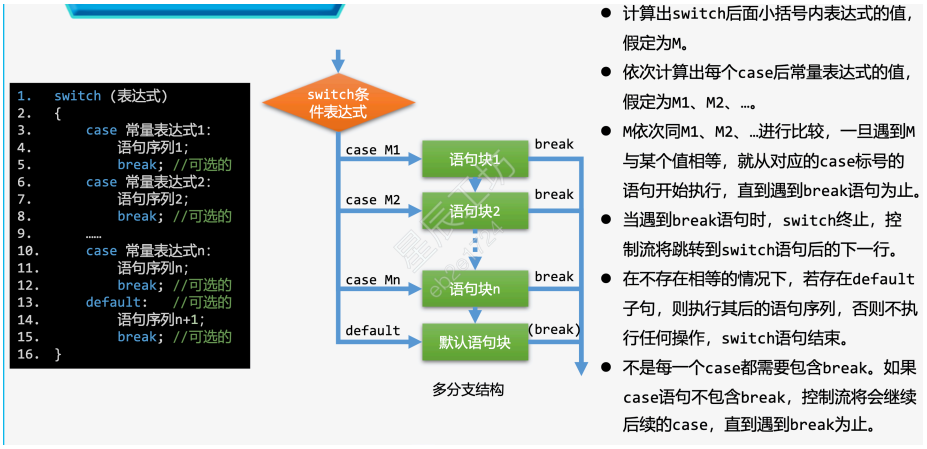


10. Switch 语句

【程序设计风格提示】

- switch(表达式)单独一行。
- case后面只能是常量（整数常量或字符常量）。
- 常量表达式后面是冒号，不是分号。
- 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。
- 在switch语句中，break和default是可以根据实际要求来选择的。

```
switch (表达式)
{
    case 常量表达式1:
        语句序列1;
        break; //可选的
    case 常量表达式2:
        语句序列2;
        break; //可选的
    ……
    case 常量表达式n:
        语句序列n;
        break; //可选的
    default:   //可选的
        语句序列n+1;
        break; //可选的
}
```



- 计算出switch后面小括号内表达式的值，假定为M。
- 依次计算出每个case后常量表达式的值，假定为M1、M2、…。
- M依次同M1、M2、…进行比较，一旦遇到M与某个值相等，就从对应的case标号的语句开始执行，直到遇到break语句为止。
- 当遇到break语句时，switch终止，控制流将跳转到switch语句后的下一行。
- 在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，switch语句结束。
- 不是每一个case都需要包含break。如果case语句不包含break，控制流将会继续后续的case，直到遇到break为止。