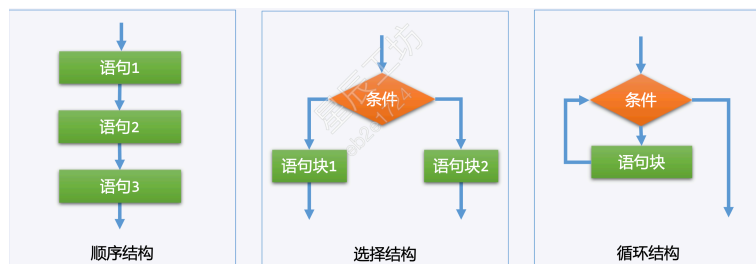


循环结构

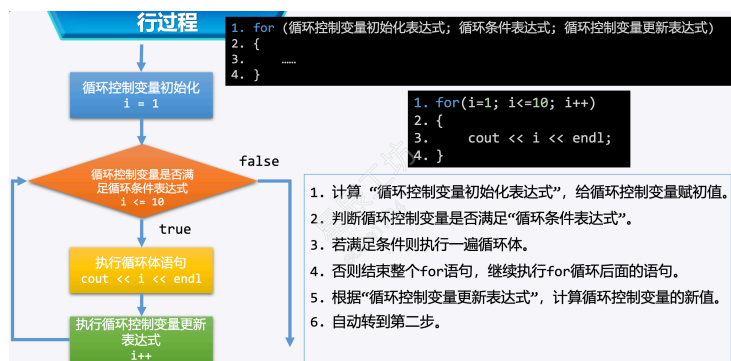
- 自减自加运算符 ++,--,x++,++x 的区别是前者先返回 x 再对 x 进行加一的操作,后者是先对 x 进行加一的操作然后返回
- 什么是循环呢?生活中经常遇到规律性并且需要重复执行的事情,我们把重复处理的步骤叫做循环
- 三大结构:



- 循环结构可以提高我们执行程序的效率,代码量更少

1. -for 循环结构

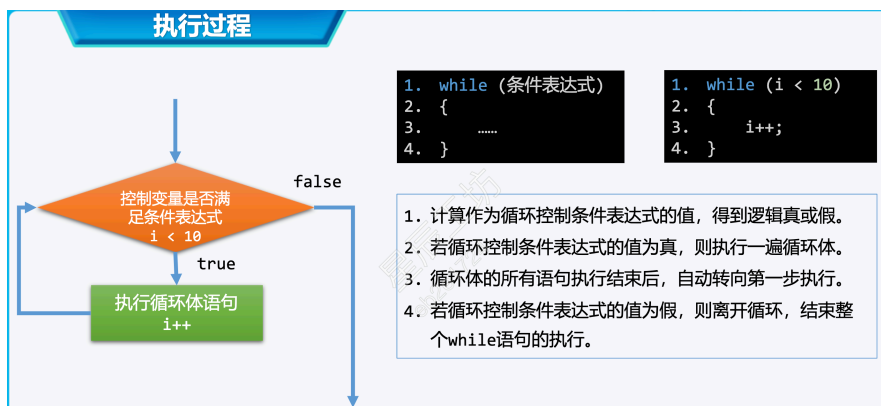
- 循环结构的三要素:循环变量,循环体,循环终止条件
- -for 循环的执行顺序:



- 同样的,for 循环中只有一个执行语句的时候,可以省略花括号
- -for 循环的初始表达式可以写多个初始化语句
- -for 循环中声明的变量的作用范围是 for 循环内,出了 for 循环就不管用了

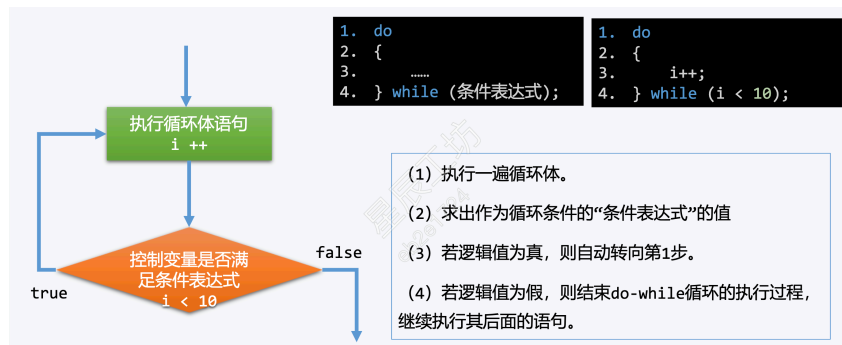
2. -while 循环

1. -while 循环的执行顺序



3. -do-while 语句

2. -do-while 语句的执行顺序



3. -do while 语句至少执行一次, while 语句可能一次都不执行

4. 最大公约数和最小公倍数

- 约数和倍数是相互依存的不能单独存在
- 公约数:几个整数中共有的约数,叫做这个数的公约数
- 公倍数:几个自然数共有的倍数
- 最大公约数:也叫最大公因数指的是,两个或者多个整数中最大的一个
- 最小公倍数,公倍数中除 0 以外最小的自然数
- 枚举法求最大公约数,从 $\min(m,n)$ 开始,若 $\gcd > 1$ 并且没有同时整除 m,n ,那么 $\gcd-1$,重复判断是否整除
- 公式法求最小公倍数 : $\text{最小公倍数}(m,n) = m \times y / \text{最大公约数}(m,n)$
-

5. 循环嵌套

- 什么叫做嵌套循环?就是在循环体里面完整的包含了一个循环结构(思考嵌套选择结构)

在一个循环结构的循环体里面，又包含了一个完整的循环结构，称为嵌套循环。

例题的代码是一个典型的2层嵌套循环结构；嵌套还可以是多层嵌套结构。

根据嵌套结构的性质，可以分为外层循环和内层循环。

外层循环每次都会等内层循环循环完毕，才算执行一次。

如果外层循环语句重复 n 次，内层循环语句重复 m 次，则内层循环的循环体总共会重复执行 $n \times m$ 次。

嵌套循环可以是任意两种循环语句进行相互嵌套。

- 语法格式:内层循环语句要相对外层循环语句进行统一缩进

```

for (外层循环变量初始化表达式; 外层循环条件表达式; 外层循环控制变量更新表达式)
{
    for (内层循环变量初始化表达式; 内层循环条件表达式; 内层循环控制变量更新表达式)
    {
        ...
    }
    ...
}

while (外层循环条件表达式)
{
    for (内层循环变量初始化表达式; 内层循环条件表达式; 内层循环控制变量更新表达式)
    {
        ...
    }
    ...
}

```

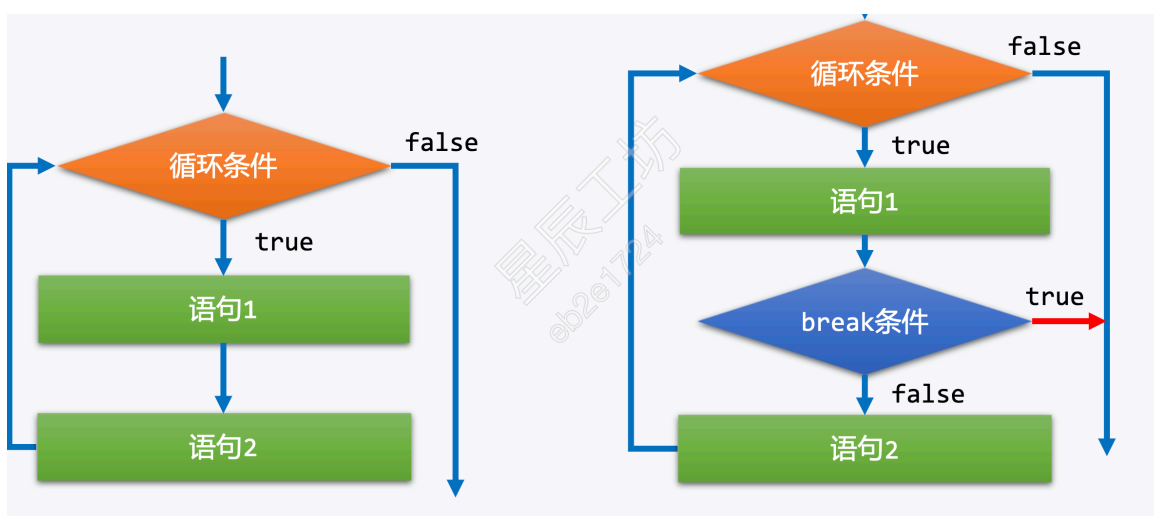
6. 趣味数学(思考如何减少循环次数,简化程序)

- 百钱买百鸡 ()
- 水仙花数 (三重循环变成单重循环)

7. 循环中的 continue 和 break

- **break** 有时为了满足条件,有时候需要提前退出循环体,此时 c++ 提供了 break 语句, break 是 c++ 的关键字,常与 if 语句连用

当 break 语句出现在一个循环内时, 循环会**立即终止**, 程序将继续执行紧接着**循环的下一条语句**。
如果是嵌套循环, break 语句会停止执行**break 所在层的循环**, 然后开始执行该块之后的下一行代码。



- **continue** 循环体中,有时为了满足条件,有时候需要跳过当前循环中的代码,强迫开始进行下一次的循环,c++ 提供了 continue 语句

continue语句有点像break语句。但它不是强迫终止整个循环。对于for循环，continue语句会执行条件测试和循环增量部分。对于while和do-while循环，continue语句会回到条件测试上。

8. 无限循环

- 无限循环之 for 循环

```
1. for (控制变量初始化表达式; 条件表达式; 增量表达式)
2. {
3.     .....
4. }
```

```
1. for ( ; ; )
2. {
3.     .....
4. }
```

- 事实上，3个部分都是可以省略的。
- for(;;)就是一个无限循环，如果不采取措施（break），它就永远不会结束。

- 无限循环之 while 循环

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

- while循环表示无限循环，一般括号中用true或一个常量1来表示。

```
1. while (true)
2. {
3.     .....
4. }
```

```
1. while (1)
2. {
3.     .....
4. }
```

9. 分解质因数

- 把一个合数用质因数相乘的形式表示出来,叫做分解质因数