

1. 程序运行流程

计算机程序代码按照一定的次序运行的过程称为程序流程。

流程分类：顺序结构、分支结构、循环结构

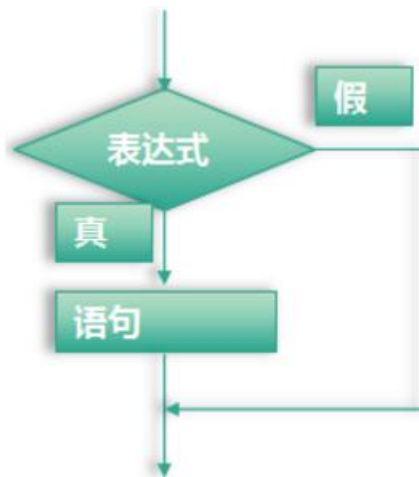
2. 顺序结构

按照语句出现的次序依次执行的程序结构，称为顺序结构

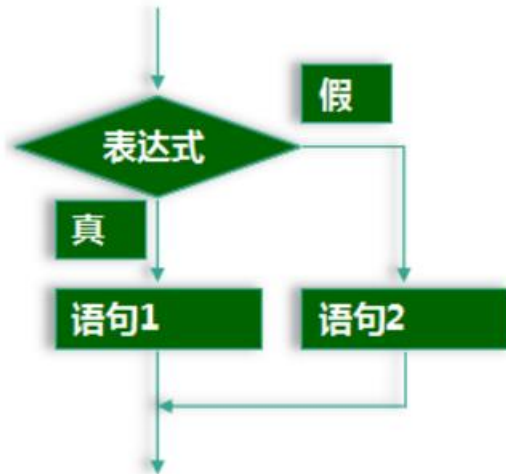


3. 分支结构

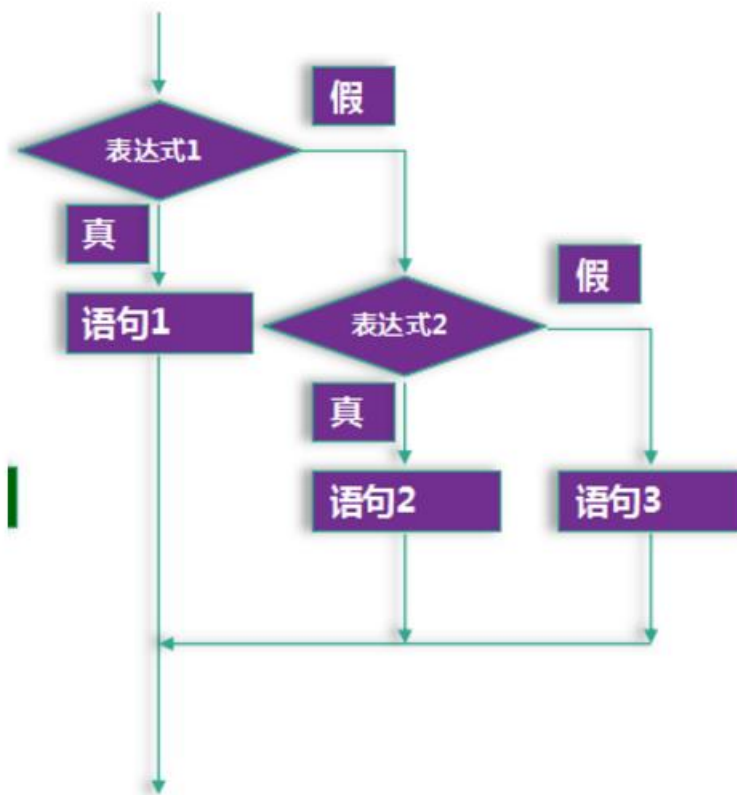
又称为选择结构，有条件的执行或跳过特定的语句或语句块，实现有选择的执行的流程控制



单路分支（if 语句）



双路分支 (if-else)



4. if 语句

if 语句用于实现分支结构，else 不是必需的

if 语句的三种形式

使用 if 语句实现单路分支结构

```

if (<条件表达式>) {
    [<语句块>]
}
  
```

使用 if-else 语句实现双路分支结构

```

if (<条件表达式>) {
    [<语句块>]
} else {
  
```

```

[<语句块>]
    }
    使用 if-else 语句实现多路分支结构
    if ( <条件表达式> ) {
[<语句块>]
    }
    .....
    else if ( 条件 ) {
[<语句块>]
    }
    [else {
[<语句块>]
    }] //中括号部分表示可选，可以省略的部分

```

5. switch 语句

用于实现多路分支结构

```

    switch(<表达式>){
case <常量 1>:
[<语句块 1>];
break;
case <常量 2>:
[<语句块 2>];
break;
.....
case <常量 n-1>:
[<语句块 n-1>];
break;
[default: <语句块 n>]
    }

```

- 表达式的返回值必须是以下几种类型之一：

byte short int char 枚举类型 封装类类型 String 类型

- case 后面必须跟着不同的常量
- break 语句用来跳出 switch 语句，break 不能丢！
- default 语句是可选语句，位置在上面也是没有影响的！

6. 三目运算的分支结构：

```

int result = 4 > 5 ? (1+2) : (2-1);
int country = 1;
int prvince = 2;

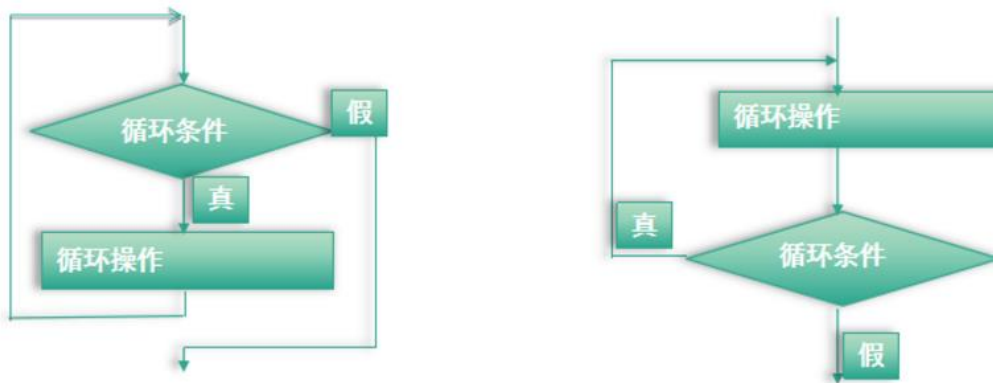
```

```
int result = 5>3 ? country++ : prvince++;
```

没有选中的分支并不会被执行！

7. 循环结构

在一定的条件下会重复执行特定代码的一种程序结构



8. for 循环

```
for(<初始化表达式>;<循环条件表达式>;<迭代表达式>){
    [<循环语句或者语句块>]
}
```

a. 执行流程

b. 执行初始化表达式

c. 计算循环条件表达式的值，如果为 true，则继续执行第 3 步，否则结束循环

d. 执行循环体语句

e. 执行迭代表达式，然后返回第 2 步继续执行

多个初始化参数的 for 循环

```
public static void forMutiV() {
    PrintStream out = System.out;
    out.println("For muti-v demo: ");
    //for (int i = 1, j=9; (i <= 9)&&(j<=9); i++, j--) {
    for (int i = 1, j=20; i <= 9; i++, j-=2) { //j-=2, 使用自定义的增减步长
        out.print(j+ " * "+ i + " = " + i * j + "\t");
    }
    out.println();
    System.out.println();
}
```

9. while、do-while 语句

```
while(<条件表达式>){
    [<循环语句或者语句块>]
}
```

```
do{
[<循环语句或者语句块>]
}while(<条件表达式>);
```

注意事项

- A. do-while 语句至少执行循环体一次，while 有可能一次也不执行循环体
- B. 循环外定义循环变量；循环体内要有循环变量的迭代表达式，否则会形成死循环

环

10. for-each 循环 主要用于迭代数值或集合中的元素，从 JDK5.0 开始引入该语句，又称为增强 for 循环

```
for(<迭代变量声明> : <数值或集合>){
[<循环语句或者语句块>]
}

String[] strs = { "AA", "BB", "CC" }; //声明 String 类型的数组 strs 并赋值
for (String string : strs) {
System.out.println("数组的元素为: " + string);
}
```

11. continue、break、return 语句

continue 语句 结束所在的循环语句的本次运行，即跳过其后的循环体语句，并开始下一次循环

break 语句 终止所在的 switch 语句或者循环语句的运行

return 直接终止了方法的执行！！——但是它逃脱不了异常处理的 finally 语句块；

12. 所有条件判断()后面紧跟的{}语句块，只有在括号内的最终值为 boolean 型的 true 才会执行！！

13. 打破循环有哪些方式：

- A. 使用表达式的循环条件打破：——打破循环条件所在的循环 eg: i <=5
- B. 使用 break 语句打破：——打破循环条件所在的循环, 多和 if 搭配
- C. 使用 return 语句打破，不仅打破循环，而且退出方法了 ——打破所有的循环
- D. 使用 flag 标志位，循环条件括号里面用此 boolean 型的标志位变量，
改变标志位的值，从而打破循环。——比较友好温柔的方式

其他知识：

A. Eclipse 中：ctrl+alt+向下箭头，复制选中的行；

B. Eclipse 的自动提示配置：Java、JavaScript、html，XML

修改编辑器的提示触发值

为 .abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ@({[<=&=:

手动触发的快捷键是 alt+/

C. Eclipse 中：在代码区，输入 syso 回车，自动完成：System.out.println();