**流程控制结构（顺序结构，选择结构，循环结构）**

**1.1顺序结构：**从上到下 从左到右 上和左不冲突

## 1.2选择结构（if和Switch）

## **if语句格式第一种（if）**

书写格式:

if(比较表达式) {

语句体;

}

执行流程:

先计算比较表达式的值，看其返回值是true还是false。

如果是true，就执行语句体；

如果是false，就不执行语句体；

**比较表达式只能是布尔类型的，运算结构有特定的语法规则，代码执行具体的逻辑运算运行判断逻辑运算的结果有两个（true，false），所以产生选择，按照不同选择执行不同代码**

**Ps：if（）小括号中的表达式只能是boolean类型**

**if语句格式第二种（if else ）**

书写格式

if(比较表达式) {

语句体1;

}else {

语句体2;

}

执行流程：

首先计算比较表达式的值，看其返回值是true还是false。

如果是true，就执行语句体1；

如果是false，就执行语句体2；

**if语句格式第三种（else if）**

书写格式

if(比较表达式1) {

语句体1;

}else if(比较表达式2) {

语句体2;

}else if(比较表达式3) {

语句体3;

}

...

else {

语句体n+1;

}

执行流程：

首先计算比较表达式1看其返回值是true还是false，

如果是true，就执行语句体1，if语句结束。

如果是false，接着计算比较表达式2看其返回值是true还是false，

如果是true，就执行语句体2，if语句结束。

如果是false，接着计算比较表达式3看其返回值是true还是false，

如果都是false，就执行语句体n+1。

**if语句格式第四种（if嵌套）**

书写格式

if(比较表达式1) {

if(比较表达式2) {

语句体1;

}else {

语句体2;

}

}else {

语句体3;

}

执行流程：

首先计算比较表达式1看其返回值是true还是false，

如果是true，就执行if的比较表达式2，看其返回值是true还是false。如果是true，就执行语句体1；如果是false，就执行语句体2.

如果是false，接着执行else后面的语句体3。

**Switch语句**

书写格式

Switch（表达式）{

Case 值1：

语句体1；

Break；

Case 值2：

语句体2；

Break；

Case 值3：

语句体3；

Break；

......

Default：

语句体n+1；

Break；

}

执行流程：

首先判断Switch表达式和那个case值相匹配

如果和case 值1相匹配，就执行语句体1，break语句结束跳出Switch语句。

如果和case 值2相匹配，就执行语句体2，break语句结束跳出Switch语句。

......

如果没有和case值没有相匹配的，则执行default里面的语句体n+1，break语句结束跳出Switch语句。

****switch case 执行时，一定会先进行匹配，匹配成功返回当前 case 的值，再根据是否有 break，判断是否继续输出，或是跳出判断。****

****Ps**** witch case 语句有如下规则****：****

switch 语句中的变量类型可以是： byte、short、int 或者 char。从 Java SE 7 开始，switch 支持字符串 String 类型了，同时 case 标签必须为字符串常量或字面量。

switch 语句可以拥有多个 case 语句。每个 case 后面跟一个要比较的值和冒号。

case 语句中的值的数据类型必须与变量的数据类型相同，而且只能是常量或者字面常量。

当变量的值与 case 语句的值相等时，那么 case 语句之后的语句开始执行，直到 break 语句出现才会跳出 switch 语句。

当遇到 break 语句时，switch 语句终止。程序跳转到 switch 语句后面的语句执行。case 语句不必须要包含 break 语句。如果没有 break 语句出现，程序会继续执行下一条 case 语句，直到出现 break 语句。

switch 语句可以包含一个 default 分支，该分支一般是 switch 语句的最后一个分支（可以在任何位置，但建议在最后一个）。default 在没有 case 语句的值和变量值相等的时候执行。default 分支不需要 break 语句。