1.逻辑的执行过程

1）顺序执行

2）条件执行

if(条件（boolean类型）){
}
else if(){
}
else(){
}

switch(数字/枚举/字符串){
case 分支1：
break;
case 分支2：
break;
default:
break;
}

3）循环执行

while(条件){
}

do{
语句;
}while(条件);

for(初始化语句;条件;变化语句){
}

死循环：while（true）

打印数组
for(int v:a){ //a为数组名
System.out.println(v);
}

控制语句：

break 1）switch中跳出某个分支
2）循环时，跳出最内层循环 break label;

continue:

4）子路径

**方法**

方法的定义

限定符（Modifiers修饰符） 返回值类型 方法名称 （形参列表） 异常类表

形参（paramaters） 实参（arguments）

方法的调用

方法名称（实参列表） max（6，1）；

方法调用的过程 （内存模型）

栈 先进后出，后进先出

调用过程是值拷贝的过程（形参的改变不会影响实参）

**重载(Overload)**

方法的签名： 方法名称+形参列表

println(int a，String s);
println(String s，int a);

返回类型不是签名的一部分。

**递归**

汉诺塔问题（Hannoi）

n代表盘子个数，source源柱子，dest目标柱子
if(n==1){
n盘子从source移到dest;
}
else{
move n-1 个盘子form source to other ;
move n-1个盘子form other to dest ;
}