一、顺序控制

1、简介

```
    程序从上到下逐行执行，中间没有任何判断和跳转
```

二、分支控制

1、简介

```
    让程序有选择地执行，分支控制有三种：单分支、双分支、多分支
```

2、单分支

```
# 简介
    当条件表达式为true时，就会执行{}的代码。如果为false，就不执行
```

```
# 语法
    if(条件表达式){
        执行代码块;
    }
```

3、双分支

```
# 简介
    当条件表达式成立，即执行代码块1；否则执行代码块2
```

```
# 语法
    if(条件表达式) {
        执行代码块1;
    } else {
        执行代码块2;
    }
```

4、多分支

```
# 简介
    1、多分支没有else，如果所有的条件表达式都不成立，则一个执行入口都没有
    2、如果有else，如果所有的条件表达式都不成立，则默认执行else代码块
```

```
# 语法
    if(条件表达式) {
        执行代码块1;
    } else if(条件表达式2) {
        执行代码块2;
    } else {
        执行代码块3;
    }
```

5、嵌套分支

```
# 简介
    在一个分支结构中又完整地嵌套了另一个完整地分支结构，里面地分支的结构成为内层分支，外面的分支结构称为外层分支
```

```
# 语法
    if() {
        if() {
            ...
        } else {
            ...
        }
    } else {
        ...
    }
```

6、switch分支

```
# 语法
    switch(表达式){
        case 常量1:
            语句块1;
            break;
        case 常量2:
            语句块2;
            break;
        default:
            语句块;
            break;
    }
```

```
# 细节
    1、表达式数据类型，应和case后的常量类型一致，或者是可以自动转成可以相互比较的类型，比如输入的是字符，而常量是int
    2、switch(表达式)中表达式的返回值必须是(byte、shot、int、char、enum、String)
    3、case子句中的值必须是常量，而不能是变量
    4、default子句是可选的，当没有匹配的case时，执行default
    5、break语句用来执行完一个case分支后使程序跳出switch语句块；如果没有写break，程序会顺序执行到switch结尾
```

7、switch和if比较

```
    1、如果判断的具体数值不多，而且符合byte、short、int、char、enum、String这6种类型，建议使用switch语句
    2、对区间判断、对结果为boolean类型判断，使用i
```

三、循环控制

1、for循环

```
# 语法
    for(循环变量初始化;循环条件;循环变量迭代){
        循环操作(语句);
    }
```

```
# 说明
    1、for关键字，表示循环控制
    2、for有四要素：(1)循环变量初始化 (2)循环条件 (3)循环操作 (4)循环变量迭代
    3、循环操作，这里可以有多条语句，也就是我们要循环执行的代码
```

```
# 细节
    1、循环条件是返回一个布尔值的表达式
    2、for(;循环判断条件;)中的初始化和变量迭代可以写道其他地方，但是两边的分号不能省略
    3、循环初始值可以有多条初始化语句，但要求类型一样，并且中间用逗号隔开，循环变量迭代也可以有多条变量迭代语句，中间用逗号隔开
```

2、while循环

```
# 语法
    while(循环条件){
        循环体(语句);
        循环变量迭代;
    }
```

```
# 说明
    1、while循环也有四要素
    2、只是四要素放的位置不一样
```

```
# 细节
    1、循环条件是返回一个布尔值的表达式
    2、while循环是先判断再执行语句
```

3、do...while循环

```
# 语法
    do{
        循环体(语句);
        循环变量迭代;
    }while(循环条件);
```

```
# 说明
    1、do while也有四要素，只是四要素的位置不一样
    2、先执行，再判断，也就是说，一定会执行一次
    3、最后有分号
```

```
# 细节
    1、循环条件是返回一个布尔值的表达式
    2、do...while循环是先执行，再判断，因此它至少执行一次
```

4、多重循环控制

```
# 语法
    for(...;...;...){
        for(...;...;...){
            ...;
        }
    }
```

```
# 说明
    1、将一个循环放在另一个循环体内，就形成了循环嵌套
    2、实质上，嵌套循环就是把内层循环当作外层循环的循环体
```

四、break

```
# 语法
    {
        ...;
        break;
    }
```

```
# 说明
    break语句用于终止某个语句块的执行，一般使用在switch或者循环[for,while,do-while]中
```

五、continue

```
# 语法
    {
        ...;
        break;
    }
```

```
# 说明
    1、continue语句用于结束本次循环，继续执行下一次循环
    2、continue语句出现在多层嵌套的循环语句体中是，可以通过标签指明要跳过的是哪一层循环
```

六、return

```
# 语法
    {
        ...;
        retuen;
    }
```

```
# 说明
    return使用在方法中，表示跳出所在的方法。如果return写在main方法，退出程序
```