

07 流程控制语句

- 顺序结构：按照代码先后顺序进行。
- 分支结构：

- if 语句

- 格式1

格式：

```
if (关系表达式) {  
    语句体;  
}
```



举例：

```
if (酒量 > 2) {  
    System.out.println("小伙子，不错!");  
}
```

- 注意点：

if 的注意事项：

1. 大括号的开头可以另起一行书写，但是建议写在第一行的末尾
2. 在语句体中，如果只有一句代码，大括号可以省略不写。个人建议，大括号还是不要省略
3. 如果对一个布尔类型的变量进行判断，不要用==号，直接把变量写在小括号即可

```
boolean flag = false;  
if (flag) {  
    System.out.println("flag的值为true");  
}
```

- 格式2

格式：

```
if (关系表达式) {  
    语句体1;  
} else {  
    语句体2;  
}
```

- 格式3

格式：

```
if (关系表达式1) {  
    语句体1;  
} else if (关系表达式2) {  
    语句体2;  
}  
...  
else {  
    语句体 n + 1;  
}
```

- switch 语句

```
switch(表达式) {  
    case 值1:  
        语句体1;  
        break;  
    case 值2:  
        语句体2;  
        break;  
    ...  
    default:  
        语句体n+1;  
        break;  
}
```

格式说明：

- I. 表达式：（**将要匹配的值**）取值为byte、short、int、char。
JDK5以后可以是枚举，JDK7以后可以是String。
- II. case：后面跟的是要和表达式进行比较的值（**被匹配的值**）。
- III. break：表示中断，结束的意思，用来结束switch语句。
- IV. default：表示所有情况都不匹配的时候，就执行该处的内容，和if语句的else相似。
- V. case后面的值只能是字面量，不能是变量
- VI. **case给出的值不允许重复**

- 注意点：

- default 可以省略，语法不会有问题，但是不建议省略。
- default 不一定可以写在最下面，可以写在任意位置，习惯写在最下面。
- **case 穿透**：先拿表达式的值和下面每一个case进行匹配。如果匹配上了，就会执行对应的语句体，如果发现了break，那么就结束整个switch语句。如果没有发现break，那么程序就会继续执行下一个case语句体，直至遇见break或右边的大括号。**若多个case的语句体重复了，那么我们考虑利用case穿透去简化代码。**
- **switch 简化**

```
int number = 1;
switch (number){
    case 1 -> System.out.println("一");
    case 2 -> System.out.println("二");
    case 3 -> System.out.println("三");
    default -> System.out.println("没有这种选项");
}
```

- switch 和 if 第三种格式各自的使用场景
 - if 的第三种格式：一般用于对范围的判断
 - switch：把有限个数据一一列举出来，让我们任选其一

- 循环结构：

- for循环

格式：

```
for (初始化语句 ; 条件判断语句 ; 条件控制语句) {
    循环体语句;
}
```

- while循环：先判断后执行

格式：

```
初始化语句 ;
while (条件判断语句){
    循环体语句;
    条件控制语句;
}
```

- for与while循环的对比

for 和 while 的区别：

- for循环中：知道循环次数或者循环的范围
- while循环：不知道循环的次数和范围，只知道循环的结束条件。

- do...while循环：先执行后判断

格式：

```
初始化语句;
do {
    循环体语句;
    条件控制语句;
} while(条件判断语句);
```

- 循环高级

- 无限循环：while格式更多一点

```
for (;;) {  
    System.out.println("学习");  
}
```

```
while(true) {  
    System.out.println("学习");  
}
```

```
do {  
    System.out.println("学习");  
} while(true);
```

- 注意事项：

- 无限循环的下面不能再写其他代码了。因为循环无法停止，下面的代码永远执行不了。

- 跳转控制语句

- **continue**：跳过本次循环，继续执行下次循环。
 - **break**：结束整个循环。