

# 语句

2018年11月3日 11:25

## 1. 语句分类

### a. //方法调用语句: println()方法, out对象调用

```
System.out.println("HelloWord");
```

### b. 赋值语句

```
x = 90;
```

### c. import和package语句

#### i. 常用的系统类库

java.lang包: lang表示核心的类(System,String)

java.lang包: 自动导入。不需要手动导入(import)

java.util包: 工具类 List Set Map

java.io包: 输入输出相关的类

java.sql包: 和数据库相关的类

#### ii. package语句必须作为类 文件的第一条有效语句

package语句不能有多条

#### iii. import位置: 放到package下边, class上边

import语句可以有多条

### d. 流程控制语句: 分支语句、循环控制语句、跳转语句

## 2. 分支语句

### a. if语句

#### i. if (表达式) {语句体}

#### ii. if (表达式){

语句体

}else{//else不能独立使用, 它是if的字句

语句体

}

.....

#### iii. if(表达式){

语句体

}else if (表达式) {

语句体

}

.....

else{

语句体

}

★iv. 注意: 表达式必须为boolean类型

else后面没有小括号

{}如果不加, 那么只识别一条语句

#### v. 案例

```
/**
 * 接收用户输入的成绩score
 * score >= 85 优秀
 * score >= 70 中等
 * score >= 60 及格
 * 否则 不及格
 */
void test1() {
    Scanner sc = new Scanner(System.in);
    System.out.print("请输入成绩: ");
    int score = sc.nextInt();
    if(score >= 85) {
        System.out.println("优秀");
    } else if(score >= 70) {
        System.out.println("中等");
    } else if(score >= 60) {
        System.out.println("及格");
    } else {
        System.out.println("不及格");
    }
}
```

#### b. switch语句

- i. switch后面的括号里的数据类型:  
byte、short、int、char、string、enum (枚举类型)
- ii. case 值不可以重复
- iii. break表示跳出switch语句体, 如果没有break, 继续项下执行
- iv. default: 可有可无, 不满足case条件的时候, 执行default

```
void test2() {
    Scanner sc = new Scanner(System.in);
    System.out.println("请输入数字: ");
    int number = sc.nextInt();

    switch(number) {
        case 1:
            System.out.println("妈妈的电话号码");
            break;

        case 2:
            System.out.println("爸爸的电话号码");
            break;
        case 3:
        default:
            System.out.println("没有这样的电话号码");
    }
}
```

#### c. 循环语句 (有规律)

种类: for、while、do...while

i. for循环 (day03\_03.Demo1)

1) 格式: for (表达式1; 表达式2; 表达式3) {循环体}

for (int i=1; int<=5; i++) {循环体; }

2) 原理: 从表达式1进入for循环, 判断表达式2返回结果是否为true, 如果为true, 执行循环体, 最后执行表达式3; 同样判断表达式2返回结果是否为true, 如果为true, 执行循环体; 最后执行表达式3....., 当表达式2返回结果为false时, 跳出循环体。

ii. while循环

1) 格式:

```
void test1() {  
    int i = 1;  
    while(i <= 5) {  
        System.out.print(i);  
        i++;  
    }  
}
```

2) 案例:

```
/**  
 * 系统随机产生1-100之间的数, 用户猜数  
 */  
  
void test2() {  
    int random = new Random().nextInt(100)+1;  
    Scanner sc = new Scanner(System.in);  
    System.out.print("请输入1-100之间的一个数: ");  
    int i = sc.nextInt();  
  
    while(i != random) {  
        if(i > random) {  
            System.out.print("猜大了, 重新输入: ");  
        } else {  
            System.out.print("猜小了, 重新输入: ");  
        }  
        i = sc.nextInt();  
    }  
    System.out.println("猜对了! ");  
}
```

iii. do...while循环 (至少执行一次)

1) 格式:

```
void test1() {  
    int i = 1;  
    do {  
        System.out.println(i);  
        i++;  
    } while(i <= 5);  
}
```

2) 案例:

```
/**  
 * 使用do...while
```

```

* 需求: ATM输入取款密码, 如果输入错误, 重新输入, 直到输入正确
*/
void test3() {
    Scanner sc = new Scanner(System.in);
    int password = 123456789;
    int pass;
    int i=1;
    do {
        System.out.print("请输入密码: ");
        pass=sc.nextInt();
        i++;
        if(i>3) {
            System.out.println("密码错误超过3次, 无法登录! ");
            System.exit(0);//退出系统
        }

    }while(password!=pass);
    System.out.println("登录成功! ");
}

```

## ★ System.exit(0);//退出系统

### iv. 三种循环的特点

- 1) 具体的循环次数: for
- 2) 当.....时候.....: while ( ) { }
- 3) 直到.....: do.....whil ( )

### d. 跳转语句

#### i. 种类: break、continue

#### ii. break

- 1) 使用环境: switch、循环
- 2) 表示跳出语句体、循环体 (向下跳)

#### iii. continue

- 1) 使用环境: 循环体
- 2) 表示结束本次循环, 开始下次循环 (向上跳)

### iv. 注意: return语句、跳转语句的后边不可以再有其他的语句 (后边的语句永远无法执行)

## 3. 嵌套循环

### a. 格式:

```

for ( ) { //外层循环行
    for(){ //内层循环列

    }
}

```

### b. 循环原理: 外层循环执行一次, 内层循环完毕

### c. 二维表格: 有规律

```

/**
 * 九九乘法表
 */
void test6() {

```

```
for(int i = 1;i<10;i++) {  
    for(int j = 1;j<=i;j++) {  
        System.out.print(j+"*"+i+"="+i*(j)+"\t");  
    }  
    System.out.println();  
}  
}
```