

# Java基础

ZCX

## 一、程序流程控制

### 1. 程序的三种执行顺序

- 顺序结构
- 分支结构 if switch
- 循环结构 for while do-while

### 2. 分支结构

#### i. if-else结构

- 根据条件的真或假，来决定执行哪个分支代码
- 分支结构共有三种写法：
  - a. if ——单分支结构
  - b. if else ——非此即彼
  - c. if else if else ——多分支结构

```
package com.cxz.branch;
import java.util.Scanner;
public class if_demo {
    public static void main(String[] args) {
        if_demo();
    }
    public static void if_demo(){
        Scanner sc = new Scanner(System.in);
        System.out.println("请输入该员工的分数：");
        int grade = sc.nextInt();
        if(grade<60){ //多分支结构
            System.out.println("E");
        }
        else if(grade>=60&&grade<70){
            System.out.println("D");
        }
        else if(grade>=70&&grade<80){
            System.out.println("C");
        }
        else if(grade>=80&&grade<90){
```

```

System.out.println("B");
}
else if(grade>=90&&grade<=100){
System.out.println("A");
}
else{
System.out.println("分数输入有误! ");
}
}
}
}

```

- if分支结构代码小练习——汽车过红绿灯能否通行

```

package com.cxz.test_branch;
import java.util.Scanner;
public class test_branch {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("前方红绿灯是什么颜色: ");
String color = sc.next();
traffic(color);
}
public static void traffic(String color){
if(color.equals("红")){
System.out.println("请等红灯,前方停车等待! ");
}
else if(color.equals("绿")){
System.out.println("请过绿灯,可以通行! ");
}
else{
System.out.println("为了行车安全请等待绿灯亮起! ");
}
}
}
}

```

## ii. switch分支结构

- 通过比较值是否相等，来决定执行哪条分支

- 结构: **switch(表达式){**

**case 常量值1: 执行代码; break;**

**case 常量值2: 执行代码; break;**

**default: 执行代码; break;**

**}**

表达式结果匹配到了哪个常量值就执行对应的代码，执行完毕通过break跳出分支，default是表达式的值与所有常量值都不匹配时，才执行。

- 例子：

```
package com.cxz.branch;
import java.util.Scanner;
public class switch_demo {
    public static void main(String[] args) {
        test();
    }
    public static void test() {
        System.out.println("请输入你的性别：");
        Scanner sc = new Scanner(System.in);
        String sex = sc.next();
        switch(sex){
            case "男":
                System.out.println("剑来");
                break;
            case "女":
                System.out.println("霸道总裁爱上我");
                break;
            default:
                System.out.println("你输入的性别有误");
                break;
        }
    }
}
```

- switch适合一个一个的值比较，if适合区间比较
- switch的表达式类型只能是byte,short,int,char,String不能是其他类型
- case后的值只能是常量值(即字面量)，不能是变量
- 使用Switch时不要忘记加上break，否则会继续向下执行，直到遇到break为止，形成穿透

有的穿透性不是缺点，相邻的case干的是同一件事，就可以利用穿透性，减少代码量，提高效率。

```
public static void test2() {
    String week="周二";
    switch (week){
        case "周一":
            System.out.println("上班");
            break;
        case "周二":
```

```

    case "周三":
    case "周四":
        System.out.println("休息");
        break;
    case "周五":
    case "周六":
        System.out.println("玩");
    }
}

```

利用穿透性，周二三四干的事一件事就可以删除相同的代码，提高效率。

### 3. 循环结构

#### i. for循环

- 格式: **for(初始化语句;条件判断语句;更新语句){循环体语句}**

- ```

for(int i = 0; i < 10; i++){
    System.out.println(i);
}

```

循环打印i的值

- 求水仙花数：满足两个要求，是一个三位数，水仙花数的个位，十位，百位位的数字的立方之和等于这个三位数本身。

```

for(int i = 100; i < 1000; i++){
    int a=i/100;
    int b=i/10-a*10;
    int c=i-100a-10b;
    int sum=aaa+bbb+cc*c;
    if(sum==i){
        System.out.println(i);
    }
}

```

#### ii. while循环

- 格式: **while(条件判断语句){循环体语句;迭代语句;}**

- ```

int i=0; while(i<3){System.out.println("hello");i++;}

```

- 知道循环几次用for，不知道用while

- ```

public static void test() {
    double money=100000;
    double num=100000;
    int year=0;
    double rate=0.017;

```

```
while(num<2.0money){
    num=num(1+rate);
    year++;
}
System.out.println(year);
}
```

本金1万，每年利率1.7%，求得到大于2万至少需要多少年

- 一张0.1mm的纸折叠几次能超过8848860mm

```
public static void test2() {
    double a = 0.1;
    int i=0;
    while(a<8848860){
        a=a*2;
        i++;
    }
    System.out.println(i);
}
```

27次

### iii. do...while循环

- 格式：**do{循环体语句;迭代语句;}while(条件判断语句);**
- `int i=0; do{System.out.println("hello");i++;}while(i<3);`

- 特点：**一定会先执行一次，然后再进行判断**

### iv. 死循环

- 可以一直执行下去，如果没有干预不会停下来

```
for(;;){
    System.out.println("死循环");
}
while(true){
    System.out.println("死循环");
}
do{
    System.out.println("死循环");
}while(true);
```

- 应用：服务器程序，一直在跟用户交互等

### v. 循环嵌套

- 外部循环每循环一次，内部循环就全部执行完1轮
- 比如：

- ```

public static void test() {
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++) {
            System.out.print("*");
        }
        System.out.println();
    }
}

```

**println()是换行, print()是输出**

输出:

```

*****
*****
*****
*****
*****

```

- 九九乘法表的打印:

- ```

public static void test2() {
    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(); 打印换行
    }
}

```

对齐制表符, 制表符是\t, 要打印, 需要加 ""

- 输出:

```

1*1=1
1*2=2   2*2=4
1*3=3   2*3=6   3*3=9
1*4=4   2*4=8   3*4=12  4*4=16
1*5=5   2*5=10  3*5=15  4*5=20  5*5=25
1*6=6   2*6=12  3*6=18  4*6=24  5*6=30  6*6=36
1*7=7   2*7=14  3*7=21  4*7=28  5*7=35  6*7=42  7*7=49
1*8=8   2*8=16  3*8=24  4*8=32  5*8=40  6*8=48  7*8=56  8*8=64
1*9=9   2*9=18  3*9=27  4*9=36  5*9=45  6*9=54  7*9=63  8*9=72  9*9=81

```

#### 4. break和continue

- break: 跳出循环, 结束循环。

- continue：跳出本次循环，继续下一次循环。
- **break只能用于结束所在循环，或者所在Switch分支**
- **continue只能在循环中进行使用**
- break实例：

```
public static void test() {  
    for(int i=0;i<10;i++){  
        if(i==5){  
            break; //跳出循环，结束循环  
        }  
        System.out.println(i);  
    }  
}
```

输出：0 1 2 3 4

- continue实例：
- ```
public static void test1() {  
    for(int i=0;i<10;i++){  
        if(i==5){  
            continue; //跳出本次循环，继续下一次循环  
        }  
        System.out.println(i);  
    }  
}
```

输出：0 1 2 3 4 6 7 8 9

## 二、实战小项目

### 1. 简单计数器

- 目标：设计一个可以实现基本数学运算的计数器程序
- 要求：用户输入两个数字和一个运算符，显示运算，计算结果

```
package com.cxz.test;  
import java.util.Scanner;  
public class test {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("请输入第一个数字：");  
        double a = sc.nextDouble();  
        System.out.println("请输入第二个数字：");  
        double b = sc.nextDouble();  
        System.out.println("请输入运算符：");
```

```

String op = sc.next();
double sum = 0;
sum=test(a,b,op);
System.out.println("结果为: " + sum);
}

public static double test(double a,double b,String op) {
double sum=0;
if(b==0 && op.equals("/")){
System.out.println("除数不能为0, 请重新输入! ");
}
else {
switch (op) {
case "+":
sum = a + b;
break;
case "-":
sum = a - b;
break;
case "*":
sum = a * b;
break;
case "/":
sum = a / b;
break;
default:
System.out.println("运算符输入有误! ");
break;
}*
}
return sum;
}
}

```

考虑除法分母不为0的情况, 如果分母为0, 则提示用户重新输入。

## 2. 猜数字

- **随机数的生成 (生成1-100之间的随机整数):**

**(int)(Math.random()\*100)+1;**

- Math.random(): 返回[0.0,1.0)之间的随机小数  
 (int)(Math.random()\*100): 返回[0,100)之间的随机整数  
 (int)(Math.random()\*100)+1: 返回[1,100]之间的随机整数

- **另一种random方式：导包，new对象，生成随机数**

- **import java.util.Random;**  
**Random r = new Random();**  
**int number=r.nextInt(100): 返回[0,100)之间的随机整数**

要生成65-91之间的随机数，也就是生成0-26之间的随机数再+65，即  
**r.nextInt(27)+65即可**

- 目标：随机生成一个1-100之间的随机整数，让用户输入一个数字，猜大，提示数字太大，猜小，提示数字太小，猜中，结束游戏。

- ```
package com.cxz.test;
import java.util.Scanner;
public class text_02 {
    public static void main(String[] args) {
        test();
    }
    public static void test(){
        //随机生成1-100的数字
        int num = (int)(Math.random()*100+1);
        System.out.println("请您猜测一下生成的数字(1-100)范围内: ");
        Scanner sc = new Scanner(System.in);
        int a = -1;
        while(a!=num){
            a = sc.nextInt();
            if(a<num){
                System.out.println("您猜的数字太小了! ");
            }
            else if(a>num){
                System.out.println("您猜的数字太大了! ");
            }
        }
        System.out.println("恭喜您猜对了! ");
    }
}
```

### 3. 验证码

- 目标：开发一个程序，生成一个指定位数的验证码，每位可以是数字，大小写字母。

- ```
package com.cxz.test;
public class test_03 {
    public static void main(String[] args) {
        System.out.println(test(6));
    }
}
```

```

public static String test(int n) {
    String code = ""; 先声明空字符串, 表示需要返回这个字符串
    for(int i=0;i<n;i++){ 循环几次表示几位验证码
        int type=(int)(Math.random()*3); 由于(int)故取值范围为[0,2]
        switch(type){ 通过Switch, type类型为0,1,2分别对应取小写字母, 大写字母, 数字
            case 0:
                code+=(char)(Math.random()*26+'a'); 取小写字母
                break;
            case 1:
                code+=(char)(Math.random()*26+'A'); 取大写字母
                break;
            case 2:
                code+=(int)(Math.random()*10); 取数字
                break;
        }
    }
    return code;
}
}

```

#### 4. 找素数

- 目标: 开发一个程序, 判断101-200之间有多少个素数, 并输出所有素数

```

package com.cxz.test;
public class test_04 {
    public static void main(String[] args) {
        int count=0;
        for(int i=101;i<=200;i++){
            count=test(i,count);
        }
        System.out.println("101-200之中共有"+ count + "个素数");
    }
    public static int test(int number,int count){
        for(int i=2;i<=Math.sqrt(number);i++){ 判断number是否为素数,即能否被从2到
        number这个数的开平方之间的数整除
        if(number%i==0){
            count++;
            System.out.println(number);
            return count;
        }
    }
    return count;
}

```

```
}  
}
```