

循环结构

Author: zhangzhang

Version: 1.0.0

- 一、引言
 - 1.1 生活中的循环
- 二、循环的概念
 - 2.1 程序中的循环
- 三、while循环
 - 3.1 while循环
 - 3.2 循环的组成
 - 3.3 初始与迭代
 - 3.4 while的特点
 - 3.5 课堂案例
- 四、do while循环
 - 4.1 do while循环
 - 4.2 do while的应用场景
- 五、for循环
 - 5.1 for循环
 - 5.2 for的特点
 - 5.3 课堂案例
- 六、流程控制关键字
 - 6.1 流程控制break
 - 6.2 流程控制continue
- 七、嵌套循环
 - 7.1 嵌套循环
 - 7.2 课堂案例

一、引言

1.1 生活中的循环

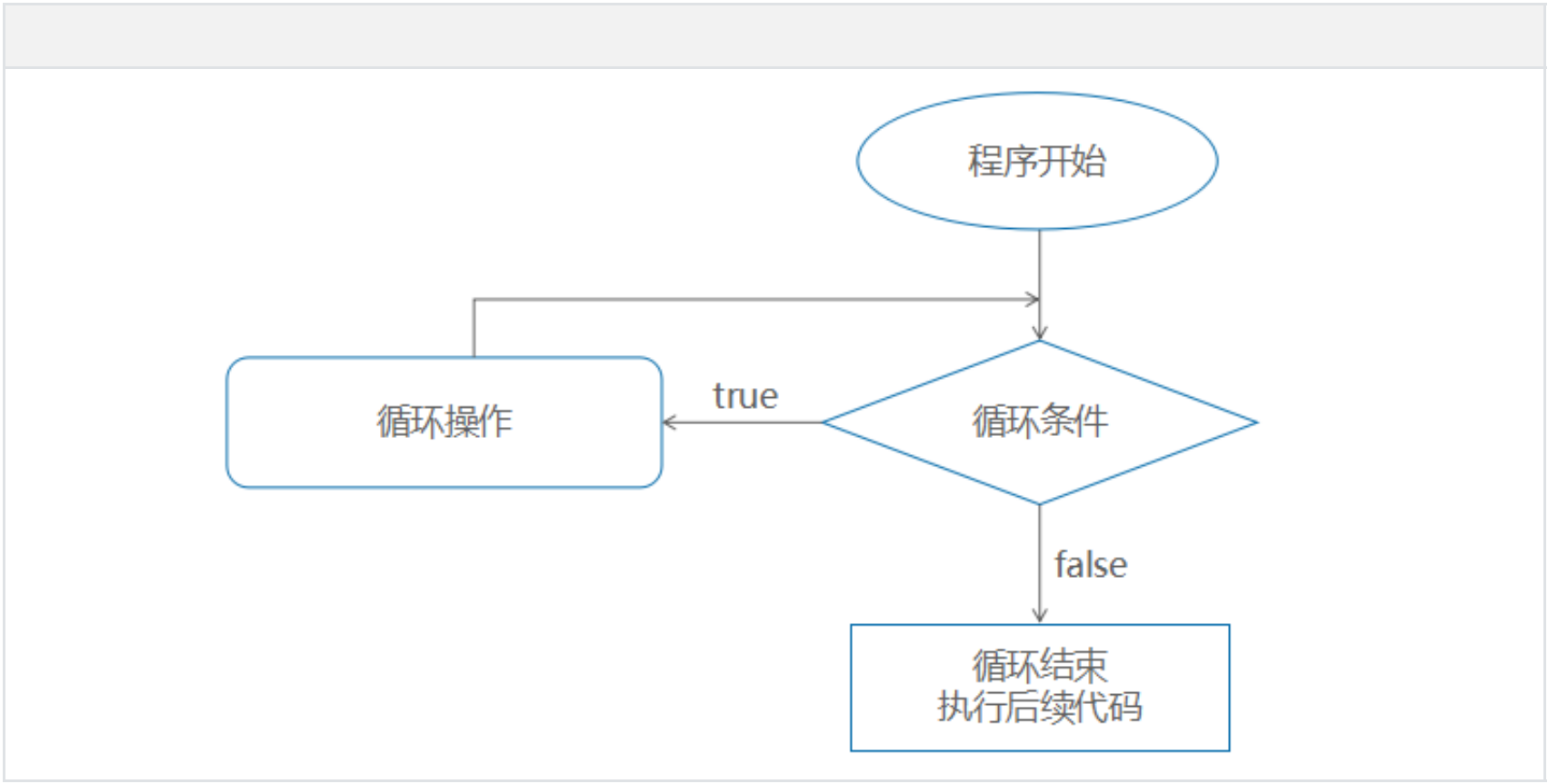


在单圈400米的跑道上完成10000米长跑，当跑完25圈时，满足条件，循环结束。

二、循环的概念

2.1 程序中的循环

概念：通过某个条件，重复的执行一段逻辑代码。



三、while循环

3.1 while循环

语法：

```
while(布尔表达式){  
    //逻辑代码 （循环操作）  
}
```

执行流程：

- 先对布尔表达式进行判断，结果为true，则执行逻辑代码。
- 本次执行完毕后，再次进行判断，结果仍旧为true，则再次执行逻辑代码。
- 直至布尔表达式的结果为false时，才会退出循环结构，执行后续代码。

3.2 循环的组成

需求：打印100遍“HelloWorld”。

```
package demo;  
  
public class TestWhile {  
    public static void main(String[] args) {  
        int i = 1;  
        while(i <= 100){  
            System.out.println("HelloWorld");  
            i++;  
        }  
    }  
}
```

1.初始部分：用以判断的变量

2.循环条件：决定是否继续循环的依据

3.循环操作：单次执行的逻辑代码或任务

4.迭代部分：控制循环条件改变的增量

注意：循环有四部分组成。

3.3 初始与迭代

以下代码，哪一行是初始部分？哪一行是迭代部分？

```
1 public class TestWhile {
2     public static void main(String[] args) {
3         int a = 1;
4         int b = 1;
5         int c = 1;
6         while(c <= 100){
7             System.out.println("HelloWorld");
8             a++;
9             b++;
10            c++;
11        }
12    }
13 }
```

被循环条件判断的变量，
才是初始部分和迭代部分！

3.4 while的特点

```
public class TestWhile {
    public static void main(String[] args) {
        int i = 1000;

        while(i <= 100){
            System.out.println("HelloWorld");
        }

        System.out.println("程序结束");
    }
}
```

首次判断不满足，则一次都不会执行。
(执行次数：0~n次)

特点：首次即有入口条件，先判断、再执行，适用于循环次数明确的情况。

```
public class TestWhile{

    public static void main(String[] args){

        //打印100遍HelloWorld

        int i = 1;

        while( i <= 10 ){
            System.out.println("HelloWorld");
            i++;
        }

        System.out.println("程序结束...");

    }
}
```

3.5 课堂案例

计算 1 + 2 + 3 + 4 + 5 ... + 98 + 99 + 100 的总和。

```
public class TestGetSum{

    public static void main(String[] args){

        //求1~100之间所有数字的总和

        //1.循环100次的问题

        //2.循环的过程中进行求和

        int i = 1;//初始部分

        int sum = 0;
```

```
while(i <= 100){ //循环条件

    sum = sum + i; //循环操作

    i++; //迭代部分
}

System.out.println("总和为: " + sum);

System.out.println("程序结束...");

}

}
```

计算 1 ~ 100 之间，所有偶数的和。

```
public class TestGetSum2{

    public static void main(String[] args){

        //求1~100之间所有偶数的和

        //方案一： 获取所有的偶数，相加求和

        //方案二： 依旧获取1~100之间的每一个数字，进行偶数的判断，满足条件之后，相加求和

        int i = 1;

        int sum = 0;

        while( i <= 100 ){

            if( i % 2 != 0 ){ //判断偶数
                //求和
                sum = sum + i;
            }

            i++;
        }

        System.out.println("偶数的和: " + sum);

    }

}
```

四、do while循环

4.1 do while循环

```
语法：
do{
    逻辑代码（循环操作）
}while(布尔表达式);
```

```
执行流程：



- 先执行一次循环操作之后，再进行布尔表达式的判断。
- 如果结果为true，则再次执行循环操作。
- 如果结果为false，才会退出循环结构，执行后续代码。

```

```
public class TestDoWhile{

    public static void main(String[] args){

        //打印100遍“HelloWorld”

        int i = 1;

        do{
            System.out.println("HelloWorld" + i);
            i++;
        }while( i <= 100);

        System.out.println("程序结束...");

    }

}
```

4.2 do while的应用场景

需求：检查学生作业完成情况，输入教师评语，决定学生是否需要抄写代码。

```
public class TestDowhile {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        char answer;//记录老师当次的评语
        do{
            System.out.println("抄写一遍作业代码.....");

            System.out.println("请输入老师评语：");
            answer = input.next().charAt(0);//输入y，表示及格，退出循环；否则继续循环
        }while(answer != 'y');

        System.out.println("作业已完成");
    }
}
```

应用场景：适用于循环次数不明确的情况。

```
import java.util.Scanner;

public class TestDowhile2{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);

        //循环操作：学生抄写代码、教师输入评语

        char answer = 'y';//赋予初始值

        do{
            System.out.println("抄写一遍...");

            System.out.println("请教师输入评语：");

            answer = input.next().charAt(0);//控制台获取'y'或者'n'
        }while( answer != 'y' );

        System.out.println("程序结束...");

    }
}
```

五、for循环

5.1 for循环

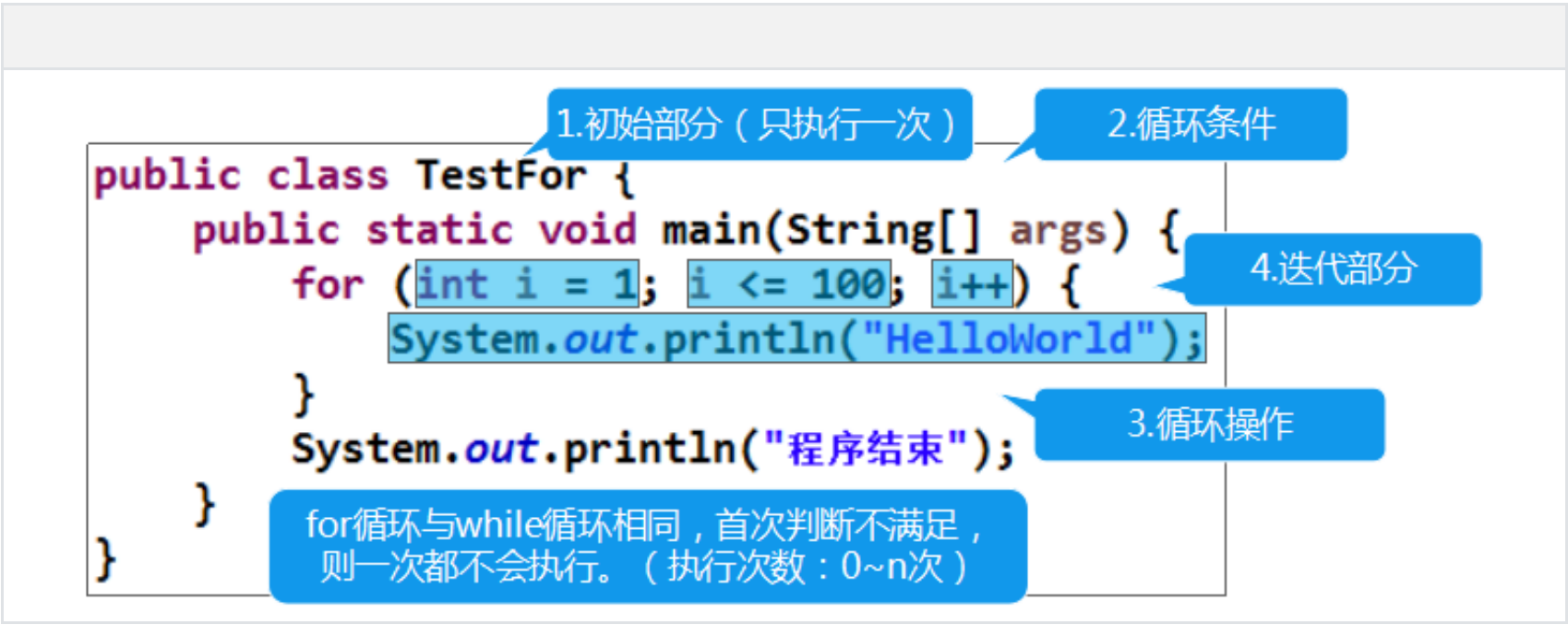
语法：
for(初始部分 ; 循环条件 ; 迭代部分){
 //循环操作
}

执行流程：

- 首次执行初始部分（仅一次）。
- 对布尔表达式进行判断，结果为true，则执行逻辑代码。
- 本次执行完毕后，执行迭代部分，再次判断，结果仍旧为true，则再次执行逻辑代码。
- 直至布尔表达式的结果为false时，才会退出循环结构，执行后续代码。

5.2 for的特点

需求：打印100遍“HelloWorld”。



特点：首次即有入口条件，先判断、再执行，适用于循环次数明确的情况。

```
public class TestFor{

    public static void main(String[] args){

        //打印100遍HelloWorld

        //    初始    条件    迭代
        for(int i = 1 ; i <= 100 ; i++){
            System.out.println("HelloWorld" + i);//操作
        }

    }

}
```

5.3 课堂案例

控制台输入整数n，计算n的阶乘（!n）。例：1*2*3... * n

```
import java.util.Scanner;

public class TestFor2{

    public static void main(String[] args){
        //阶乘  例如：6的阶乘  1*2*3*4*5*6

        Scanner input = new Scanner(System.in);

        System.out.println("请输入一个需要计算阶乘的值：");

        int target = input.nextInt(); //接收一个目标值，计算该值的阶乘

        long sum = 1; //注意：int类型的sum，最大计算到12的阶乘，如果计算13的阶乘，积会超过int的取值，不精确。

        for(int i = 1; i <= target; i++){

            //相乘求积
            sum = sum * i;
        }

        System.out.println("阶乘的积为： " + sum);
    }

}
```

现有1个班级的5名同学，请通过控制台输入5名同学的分数，并计算平均分。

```
import java.util.Scanner;

public class TestFor3{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);

        //计算5位同学的平均分

        double sum = 0.0;//总和

        for(int i = 1; i <= 5; i++){
```

```
        //1.循环控制台录入分数
        System.out.println("请输入第" + i + "位同学的成绩: ");

        double score = input.nextDouble();

        //2.累加总和
        sum = sum + score;

    }

    double avg = sum / 5;

    System.out.println("平均分: " + avg);

}

}
```

六、流程控制关键字

6.1 流程控制break

break：终止、跳出switch、循环结构。

```
public class TestBreak {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            if(i == 5){
                break;
            }
            System.out.println("当前循环次数: " + i);
        }
        System.out.println("循环结束");
    }
}
```

运行结果：
当前循环次数：1
当前循环次数：2
当前循环次数：3
当前循环次数：4
循环结束

```
import java.util.Scanner;

public class TestBreak{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);

        //控制台录入5位同学成绩，如果有任何一位同学的成绩产生非法数据（不满足0~100之间的数字）时，将直接退出整合循环操作

        double sum = 0.0;

        boolean flag = true;//合法

        for(int i = 1; i <= 5; i++){

            System.out.println("请输入第" + i + "位同学的成绩: ");

            double score = input.nextDouble();

            if(score < 0 || score > 100.0){
                flag = false;//非法数据
                break;
            }

            sum = sum + score;

        }

        if( flag == true ){ //根据flag标记决定是否需要计算和输出平均分
            double avg = sum / 5;

            System.out.println("平均分: " + avg);
        }else{
            System.out.println("非法数据，请重新运行程序计算平均分");
        }

    }

}
```


6.2 流程控制continue

continue：结束本次、进入下一次循环。

```
public class TestContinue {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            if(i == 5){
                continue;
            }
            System.out.println("当前循环次数: " + i);
        }
        System.out.println("循环结束");
    }
}
```

运行结果：
当前循环次数：1
当前循环次数：2
当前循环次数：3
当前循环次数：4
当前循环次数：6
当前循环次数：7
当前循环次数：8
当前循环次数：9
当前循环次数：10
循环结束

循环执行中，遇到continue，
则跳过此次，进入下一次循环。

```
import java.util.Scanner;

public class TestContinue{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);

        //控制台录入5位同学成绩，如果有任何一位同学的成绩产生非法数据（不满足0~100之间的数字）时，结束本次的统计，进入下一次循环当中

        double sum = 0.0;

        for(int i = 1; i <= 5; ){
            System.out.println("请输入第" + i + "位同学的成绩: ");

            double score = input.nextDouble();

            if(score < 0 || score >100.0){
                //非法数据
                continue;
            }

            sum = sum + score;

            i++;
        }

        double avg = sum / 5;

        System.out.println("平均分: " + avg);
    }
}
```

七、嵌套循环

7.1 嵌套循环

概念：在一个完整的循环结构中，嵌套另一个完整的循环结构。

需求：打印3行5颗星。

```
public class TestNestFor {
    public static void main(String[] args) {
        for (int i = 1; i <= 3; i++) {
            for (int j = 1; j <= 5; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

外层控制行数

内层控制列数

运行结果：


```
public class TestNestedFor{

    public static void main(String[] args){

        /*

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

        */

        //不采用此种方式
        //System.out.println("*****");

        //冗余（重复）代码，避免重复，采取复用代码的形式

        //将内部的代码，重复3次
        for(int i = 1 ; i <= 3 ; i++){// i = 4

            //在1行中打印5颗星
            for(int j = 1 ; j <= 5 ; j++){//
                System.out.print("*");
            }
            System.out.println();

        }
        //程序流程到达此处

        /*

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

        */

    }
}
```

需求：计算3个班，每个班5名同学的平均成绩。

```
public class TestNestFor {
    public static void main(String[] args) {
        for (int i = 1; i <= 3; i++) {
            for (int j = 1; j <= 5; j++) {
                //计算一个班5名同学的平均成绩
            }
        }
    }
}
```

外层控制循环次数

内层控制单次循环操作

```
import java.util.Scanner;

public class TestNestedFor2{

    public static void main(String[] args){

        Scanner input = new Scanner(System.in);

        for(int k = 1 ; k <= 3 ; k++){

            System.out.println("---第"+k+"个班---");

            //求一个班，5位同学的平均成绩
            double sum = 0.0;

            for(int i = 1 ; i <= 5 ; i++){

                System.out.println("请输入第" + i + "位同学的成绩：");

                double score = input.nextDouble();
```

```
        sum += score;

    }

    double avg = sum / 5;

    System.out.println("第" + k + "班5位同学的平均分: " + avg);

}

}

}
```

7.2 课堂案例

打印直角三角形。

```
public class TestNestedFor3{

    public static void main(String[] args){

        //打印直角三角形

        /*

        *      j <= 1
        **     j <= 2
        ***    j <= 3
        ****   j <= 4
        ***** j <= 5

        -----

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

        */

        //外层控制行数
        for(int i = 1 ; i <= 5 ; i++){ // i = 6

            //内层控制列数
            for(int j = 1 ; j <= i ; j++){ //
                System.out.print("*");
            }
            System.out.println();

        }

        System.out.println("程序结束");

    }

}
```

打印等腰三角形。

```
public class TestNestedFor4{

    public static void main(String[] args){

        //打印等腰三角形

        // #####
        // #####
        // #*****
        // #*****
        // #*****

        int rows = 5;

        //1.打印倒的直角三角形
        for(int i = 1 ; i <= rows ; i++){ //外层打印5行

            //打印4行的直角三角形
            for(int j = rows - 1 ; j >= i ; j--){ // j = 4 ; j >= 1
                //被满足4次
                System.out.print("#");
            }

            System.out.println();

        }

}
```

```
//2.打印正的直角三角形（每次递进数值上 *2-1 ）
//  *
//  ***
//  *****
//  *********
//  *********

for(int i = 1 ; i <= rows ; i++){ // i = 1

    for(int j = 1 ; j <= i*2-1 ; j++){
        System.out.print("*");
    }

    System.out.println();
}


//整合

//  ####*
//  #####*
//  #####*
//  #####*
//  #####*
//  #####*

for(int i = 1 ; i <= rows ; i++){ // i = 2

    for(int j = rows - 1 ; j >= i ; j--){ // j = 4 ; 4 >= 2
        System.out.print(" ");
    }

    //程序当前的执行流程位置
    for(int j = 1 ; j <= i*2-1 ; j++){ // j = 1 ; j <= 3
        System.out.print("*");
    }

    System.out.println();
}

}

}
```