

循环结构

Java Platform Standard Edition

郑春光

课程目标

CONTENTS

ITEMS **1** 循环的概念

ITEMS **2** while循环

ITEMS **3** do while循环

ITEMS **4** for循环

ITEMS **5** 流程控制关键字

ITEMS **6** 嵌套循环

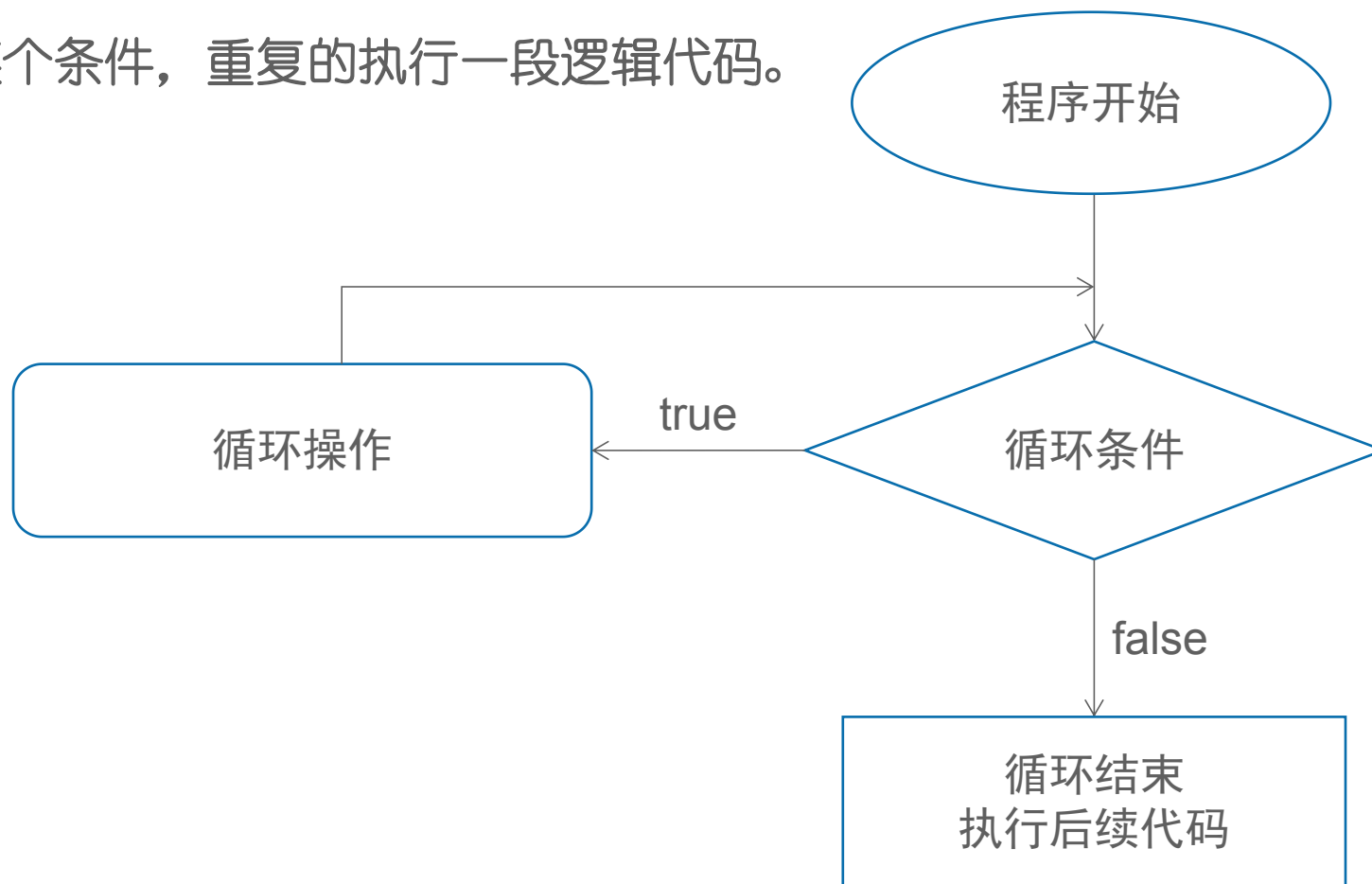
生活中的循环



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。

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。



- 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。

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。

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！

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。

- 计算 $1 + 2 + 3 + 4 + 5 \dots + 98 + 99 + 100$ 的总和。
- 计算 $1 \sim 100$ 之间，所有偶数的和。

- 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。

do while的特点

- 需求：打印100遍 “HelloWorld”。

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

直接执行循环操作，
首次没有入口条件。

判断满足，则循环。
不满足，则至少执行一次。
(执行次数：1~n次)

- 特点：首次没有入口条件，**先执行，再判断。**

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。

- 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。

for的特点

- 需求：打印100遍“HelloWorld”。

```
public class TestFor {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 100; i++) {  
            System.out.println("HelloWorld");  
        }  
        System.out.println("程序结束");  
    }  
}
```

1.初始部分（只执行一次）

2.循环条件

4.迭代部分

3.循环操作

for循环与while循环相同，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。

- 控制台输入整数 n ，计算 n 的阶乘 ($n!$)。例： $1 * 2 * 3 \dots * n$
- 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。

- 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");  
    }  
}
```

循环执行中，遇到break，
则退出整个循环结构。

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

- 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");  
    }  
}
```

循环执行中，遇到continue，
则跳过此次，进入下一次循环。

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

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：

嵌套循环

- 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

- 打印等腰三角形。
- 打印菱形。

课后作业

- 根据以下图片提示，完成相应功能。（要求使用：switch、do while）

```
=====欢迎使用ATM自动银行服务=====
1:开户 2:存款 3:取款 4:转帐 5:查询余额 6:修改密码 0:退出
=====
请输入操作编号: 1
---执行开户功能---
```

```
=====欢迎使用ATM自动银行服务=====
1:开户 2:存款 3:取款 4:转帐 5:查询余额 6:修改密码 0:退出
=====
请输入操作编号: 2
---执行存款功能---
```

```
=====欢迎使用ATM自动银行服务=====
1:开户 2:存款 3:取款 4:转帐 5:查询余额 6:修改密码 0:退出
=====
请输入操作编号: 0
---执行退出功能---
```

```
=====欢迎使用ATM自动银行服务=====
1:开户 2:存款 3:取款 4:转帐 5:查询余额 6:修改密码 0:退出
=====
请输入操作编号: 7
---输入错误，请重新输入---
=====欢迎使用ATM自动银行服务=====
1:开户 2:存款 3:取款 4:转帐 5:查询余额 6:修改密码 0:退出
=====
请输入操作编号: -1
---输入错误，请重新输入---
=====欢迎使用ATM自动银行服务=====
1:开户 2:存款 3:取款 4:转帐 5:查询余额 6:修改密码 0:退出
=====
请输入操作编号: 0
---执行退出功能---
```

- 循环的概念：
 - 通过某个条件，重复的执行一段逻辑代码。
- while循环：
 - `while(){ }`
- do while循环：
 - `do{ }while;`
- for循环：
 - `for(初始 ; 条件; 迭代) {操作;}`
- 流程控制关键字：
 - `break`、`continue`
- 嵌套循环：
 - 在一个完整的循环结构中，嵌套另一个完整的循环结构。