

第五章 多重循环

课前回顾

1. 程序调试的步骤是什么？

首先在有效代码中下断点，然后使用F8单步执行，观察变量的值的变化是否按照我们设计思维变化，从而找出问题，并解决问题

2. 循环结构有几种，分别有什么特点？

while循环、do-while循环、for循环。while循环和for循环都是先判断后执行，如果条件一开始就不满足，那么while循环和for循环可能一次都不执行。do-while循环是先执行再判断，因此，do-while循环至少都会执行一次

3. 循环三要素分别是什么？

定义循环变量并赋初值，循环操作，循环变量的更新

4. 循环中break和continue有什么区别？

break在循环中的作用就是终止break所在的循环，执行循环下面的代码。通常与if选择结构配合使用

continue在循环中的作用就是跳过本次循环，进入下一次循环。通常与if选择结构配合使用

主要内容

- 二重循环 **重点**
- Java 中的标号（标签） **难点**
- **熟悉**

课程目标

- 掌握多重循环的使用
- 熟悉 Java 中的标号

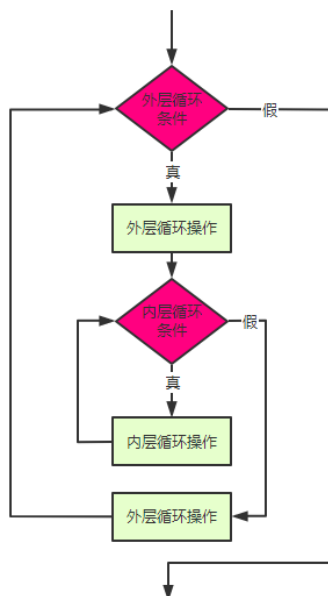
第一节 二重循环

1. 什么是二重循环

二重循环就是一个循环结构中又包含另外一个循环结构

```
1  while(外层循环条件){
2      //外层循环操作
3      while(内层循环条件){
4          //内层循环操作
5      }
6      //外层循环操作
7  }
8  while(外层循环条件){
9      //外层循环操作
10     for(循环变量初始化;内层循环条件;循环变量更新){
11         //内层循环操作
12     }
13     //外层循环操作
14 }
```

2. 执行流程图



3. 应用场景

打印乘法表

```

1  1x1=1
2  1x2=2  2x2=4
3  1x3=3  2x3=6  3x3=9
4  1x4=4  2x4=8  3x4=12  4x4=16
5  1x5=5  2x5=10  3x5=15  4x5=20  5x5=25
6  1x6=6  2x6=12  3x6=18  4x6=24  5x6=30  6x6=36
7  1x7=7  2x7=14  3x7=21  4x7=28  5x7=35  6x7=42  7x7=49
8  1x8=8  2x8=16  3x8=24  4x8=32  5x8=40  6x8=48  7x8=56  8x8=64
9  1x9=9  2x9=18  3x9=27  4x9=36  5x9=45  6x9=54  7x9=63  8x9=72  9x9=81
  
```

分析

- 乘法表需要打印9行
- 每一行中的列数都是跟行号一致

代码实现

```

1  public class Example1 {
2
3      public static void main(String[] args) {
4          for(int i=1; i<=9; i++){//打印9行
5              for(int j=1; j<=i; j++){//j<=i表示列数的最大值就是行号
6                  //print表示在同一行中打印，也就是不会换行
7                  System.out.print( j + " x " + i + " = " + i*j + "\t");
8              }
9              System.out.println();
10         }
11     }
12 }
  
```

打印矩形

```
1 *****
2 *****
3 *****
4 *****
```

分析

- a. 矩形一共打印4行
- b. 每一行都有10列

代码实现

```
1 public class Example2 {
2
3     public static void main(String[] args) {
4         for(int i=0; i<4; i++){//外层循环控制行数
5             for(int j=0; j<10; j++){//内层循环控制列数
6                 system.out.print("*");
7             }
8             System.out.println();
9         }
10    }
11 }
```

思考如何打印空心矩形?

```
1 *****
2 *&&&&&&&*
3 *&&&&&&&*
4 *****
```

分析

- a. 矩形一共打印4行
- b. 每一行都有10列
- c. 矩形的第一行和最后一行都是 '*', 第一列和最后一列也是 '*'

代码实现

```
1 public class Example3 {
2
3     public static void main(String[] args) {
4         for(int i=0; i<4; i++){
5             for(int j=0; j<10; j++){
6                 if(i == 0 || i == 3 || j == 0 || j == 9){
7                     System.out.print("*");
8                 } else {
9                     System.out.print(" ");
10                }
11            }
12            System.out.println();
13        }
14    }
```

```
14     }
15 }
```

打印三角形

```
1  &&&&*
2  &&&***
3  &&*****
4  &*****
5  *****
```

分析

- 三角形一共5行
- 三角形的第一行的列数与三角形的行数一致
- 三角形的每一行的空白数量 = 三角形的总行数 - 行号
- 三角形的每一行的'*'数量 = (行号 - 1) * 2 + 1

代码实现

```
1  public class Example4 {
2
3      public static void main(String[] args) {
4          int totalLines = 25; //总行数
5          for(int i=1; i<=totalLines; i++){
6              int whiteSpaceCount = totalLines - i; //空白数量 = 三角形的总行数 -
行号
7              for(int m=0; m<whiteSpaceCount; m++){
8                  System.out.print(" ");
9              }
10             int starCount = (i - 1) * 2 + 1; // '*'数量 = (行号 - 1) * 2 + 1
11             for(int n=0; n<starCount; n++){
12                 System.out.print("*");
13             }
14             System.out.println();
15         }
16     }
17 }
```

思考如何打印倒三角形？

```
1  *****
2  &*****
3  &&*****
4  &&&***
5  &&&&*
```

分析

- 三角形一共5行
- 三角形最后一行的空白数量 = 行号 - 1
- 最后一行的'*'数量 = (总行数 - 行号) * 2 + 1

代码实现

```
1 public class Example5 {
2
3     public static void main(String[] args) {
4         int totalLines = 15;
5         for(int i=1; i<=totalLines; i++){
6             int whiteSpaceCount = i - 1; //空白数量 = 行号 - 1
7             for(int m = 0; m<whiteSpaceCount; m++){
8                 System.out.print(" ");
9             }
10            int starCount = (totalLines - i) * 2 + 1; // '*'数量 = (总行数 -
            行号) * 2 + 1
11            for(int n=0; n<starCount; n++){
12                System.out.print("*");
13            }
14            System.out.println();
15        }
16    }
17 }
```

打印菱形

```
1      *
2     ***
3    *****
4   *****
5  &*****
6 &&***
7 &&&*
```

分析

- a. 菱形的总行数一定是奇数
- b. 上三角形 （包含对称轴在内）

代码实现

```
1 public class Example6 {
2
3     public static void main(String[] args) {
4         int totalLines = 27;
5         //先打印上三角形
6         int topLines = totalLines / 2 + 1; //上三角形总行数
7         for(int i=1; i<=topLines; i++){
8             int whiteSpaceCount = topLines - i; //空白数量 = 上三角形的总行数 -
            行号
9             for(int m=0; m<whiteSpaceCount; m++){
10                System.out.print(" ");
11            }
12            int starCount = (i - 1) * 2 + 1; // '*'数量 = (行号 - 1) * 2 + 1
13            for(int n=0; n<starCount; n++){
14                System.out.print("*");
15            }
16            System.out.println();
17        }
```

```

18      //再打印下三角形
19      int downLines = totalLines - topLines; //下三角形总行数
20      for(int i=1; i<=downLines; i++){
21          //每一行的空白数量与行号相同
22          for(int m=0; m<i; m++){
23              system.out.print(" ");
24          }
25          //每一行的 '*' 数量 = (downLines - 行号) * 2 + 1
26          int starCount = (downLines - i) * 2 + 1;
27          for(int n=0; n<starCount; n++){
28              system.out.print("*");
29          }
30          system.out.println();
31      }
32  }
33  }

```

4. 练习

求2~100内的所有质数

分析

- 求质数的范围2~100
- 判断这个范围内的每一个数是否是质数
- 质数的特征：一个数除了1和它本身之外，不能被任何数整除。换言之，就是从2开始，到这个数-1为止，没有任何一个数能够被这个数整除。

代码实现

```

1  public class Example7 {
2
3      public static void main(String[] args) {
4          for(int i=2; i<=100; i++){
5              if(i == 2) {
6                  system.out.println(i + "是质数");
7              } else {
8                  boolean isPrime = true; //默认 i 是质数
9                  for(int j=2; j<i; j++){
10                     //如果一个数模上另外一个数结果为零，则表示这个数能够被另一个数整除
11                     if(i % j == 0){ //如果 i 能够被任意一个 j 整除，则说明 i 不是
质数
12                         isPrime = false;
13                         break;
14                     }
15                 }
16                 if(isPrime){
17                     system.out.println(i + "是质数");
18                 }
19             }
20         }
21     }
22 }

```

第二节 Java 中的标号（标签 label）

1. 语法规则

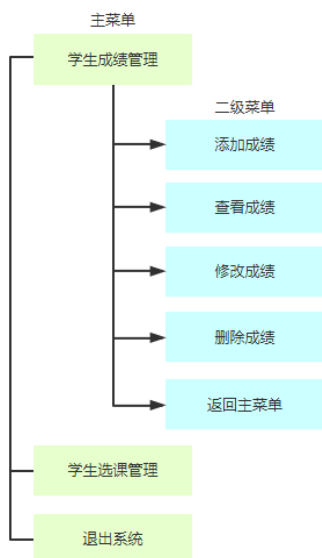
1 | 标号名称：循环结构

2. 作用

标号的作用就是给代码添加一个标记，方便后面使用。通常应用在循环结构中，与break语句配合使用

3. 应用场景

有如下菜单：



实现其中返回主菜单的功能

4. 代码实现

```
1 public class Example8 {
2
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         while (true){
6             System.out.println("=====");
7             System.out.println("1.学生成绩管理");
8             System.out.println("2.学生选课管理");
9             System.out.println("3.退出系统");
10            System.out.println("=====");
11            System.out.println("请选择菜单编号: ");
12            int menuNo = sc.nextInt();
13            if(menuNo == 1){
14                childMenu:while(true){
15                    System.out.println("*****");
16                    System.out.println("1.添加成绩");
17                    System.out.println("2.查看成绩");
18                    System.out.println("3.修改成绩");
19                    System.out.println("4.删除成绩");
20                    System.out.println("5.返回主菜单");
21                    System.out.println("*****");
22                    System.out.println("请选择菜单编号: ");
23                    int number = sc.nextInt();
24                    switch (number){
25                        case 1:
```

```
26         System.out.println("你选择添加成绩");
27         break;
28     case 2:
29         System.out.println("你选择查看成绩");
30         break;
31     case 3:
32         System.out.println("你选择修改成绩");
33         break;
34     case 4:
35         System.out.println("你选择删除成绩");
36         break;
37     case 5:
38         System.out.println("你选择返回主菜单");
39         break childMenu; //java中的标号，可以理解为一个代码
    的标记
40     }
41 }
42 } else if(menuNo == 2){
43
44 } else {
45     System.out.println("感谢使用本人开发的系统");
46     break; //终止break所在的循环
47 }
48 }
49 }
50 }
```