

第四章 循环结构

课前回顾

1. 选择结构都有哪些

基本if选择结构、if-else选择结构、嵌套if-else选择结构、多重if-else选择结构、switch选择结构

2. 描述逻辑短路

逻辑与短路：在使用逻辑与衔接的多个条件中，如果其中一个条件为假，那么该条件之后的所有条件均得不到执行，从而形成逻辑与短路

逻辑或短路：在使用逻辑或衔接的多个条件中，如果其中一个条件为真，那么该条件之后的所有条件均得不到执行，从而形成逻辑与短路

3. switch选择结构支持哪些数据类型？有无注意事项？

byte int short char String Enum Byte Integer Short

需要注意的是，switch选择结构如果使用String，则需要满足 JDK 必须在 JDK 7 及以上

主要内容

• 程序调试	重点	
• while 循环	重点	难点
• do-while 循环	重点	难点
• for 循环	重点	难点
• break	重点	难点
• continue	重点	难点

课程目标

- 掌握程序调试
- 掌握while、do-while及for循环的使用
- 掌握break和continue流程控制

第一节 程序调试

1. 什么是程序调试

当程序出现问题时，我们希望程序能够暂停下来，然后通过我们操作使代码逐行执行，观察整个过程中变量的变化是否按照我们设计程序的思维变化，从而找问题并解决问题，这个过程称之为程序调试

2. 为什么需要程序调试

人无完人，考虑问题不可能面面俱到，尤其是在编写复杂程序的时候，如果程序运行出了问题，定位问题就是关键了，要定位问题就需要使用程序调试

3. 如何进行程序调试

首先在有效的代码上下断点

然后以DEBUG模式启动程序

程序启动后，通过操作使程序代码逐行执行，观察变量的值的变化，从而找出问题并解决问题

4. 程序调试演示

第二节 循环结构

1. 什么是循环

循环就是同样的事情反复做多次

2. 为什么要使用循环

思考如下代码存在什么问题？

```
1 public class Example1{
2
3     public static void main(String[] args){
4         System.out.println("Hello world");//1
5         System.out.println("Hello world");//2
6         System.out.println("Hello world");//3
7         System.out.println("Hello world");//4
8         System.out.println("Hello world");//5
9         System.out.println("Hello world");//6
10        System.out.println("Hello world");//7
11        System.out.println("Hello world");//8
12        System.out.println("Hello world");//9
13        System.out.println("Hello world");//10
14    }
15 }
```

重复的编码出现多次，在Java中，这样的情况我们称之为代码冗余。那么如何减少这种重复的冗余代码？这就需要使用到循环结构了。

```
1 public class Example2{
2
3     public static void main(String[] args){
4         int i=0;
5         while (i < 10){
6             System.out.println("Hello world");
7             i++;
8         }
9     }
10 }
```

对比之前的代码，冗余部分已经完全被去掉了，但是实现的功能却完全是一样的

3. 循环三要素

定义循环变量并赋初值

循环条件

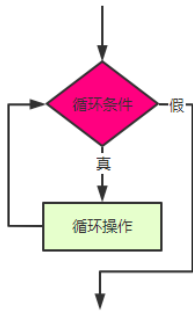
循环变量的更新

第三节 while循环

1. 语法

```
1 while(循环条件){  
2     //循环操作  
3 }
```

2. 执行流程图



3. 案例

打印0-9之间的所有整数

4. 代码实现

```
1 public class Example3 {  
2  
3     public static void main(String[] args) {  
4         //1. 定义循环变量并赋初值  
5         int i = 0;  
6         //2. 添加循环条件  
7         while(i < 10){  
8             System.out.println(i);  
9             //3. 循环变量更新  
10            ++i;  
11        }  
12    }  
13 }
```

演示 while 循环执行过程

断点调试

5. 总结

while 循环的特征就是先判断，后执行。如果一开始条件就不满足，那么 while 循环可能一次也不执行

练习

1. 在控制台输出100以内能够被3整除的所有整数。

```

1 public class Example4 {
2
3     public static void main(String[] args) {
4         int i = 0; //定义循环变量并赋上初值
5         while (i < 100){//循环条件
6             if(i % 3 == 0){//能够被3整除
7                 System.out.println(i);
8             }
9             i++;//循环变量的更新
10        }
11    }
12 }

```

2. 水仙花数是指一个 3 位数，它每位上的数字的 3 次幂之和等于它本身（例如： $1^3 + 5^3 + 3^3 = 153$ ），水仙花数的取值范围在100~1000之间。请设计一道程序，求出所有的水仙花数并在控制台打印出来

```

1 public class Example5 {
2
3     public static void main(String[] args) {
4         int number = 100;
5         while(number < 1000){
6             //101 => 101 % 10
7             int ge = number % 10;
8             //111 => 111 / 10 = 11 % 10 = 1
9             int shi = number / 10 % 10;
10            // 501 => 501 / 100 = 5
11            int bai = number / 100;
12            if(ge * ge * ge + shi * shi * shi + bai * bai * bai ==
number){
13                System.out.println(number);
14            }
15            number++;
16        }
17
18        // 3 = 27                4 = 64
19        // 7 = 49 * 7 = 343      0 = 0
20        // 0 = 0                1 = 1        7 = 343
21    }
22 }

```

第四节 do-while 循环

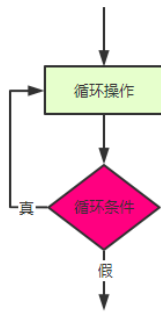
1. 语法

```

1 do {
2     //循环操作
3 } while(循环条件);

```

2. 执行流程图



3. 案例

从控制台录入学生的成绩并计算总成绩，输入0时退出

4. 代码实现

```
1 public class Example6 {
2
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int totalScore = 0; //默认总成绩为0
6         int score; //定义循环变量，但是没有赋初值
7         do{
8             System.out.println("请输入成绩: ");
9             score = sc.nextInt(); //第一次执行时，对循环变量score赋上初始值
10            // totalScore = totalScore + score;
11            totalScore += score;
12        } while (score != 0);
13        System.out.println("总成绩为: " + totalScore);
14    }
15 }
```

演示 do-while 循环执行过程

断点调试

练习

从控制台输入一个数字，如果该数字不能被7整除，则重新输入，直到输入的数字能够被7整除为止。

```
1 public class Example7 {
2
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int number; //定义循环变量，但没有赋初值
6         do{
7             System.out.println("请输入一个整数: ");
8             number = sc.nextInt();
9         } while(number % 7 != 0);
10        System.out.println(number);
11    }
12 }
```

5. 总结

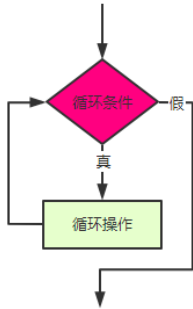
do-while 循环的特征就是先执行，后判断。do-while 循环最少会执行一次

第五节 for 循环

1. 语法

```
1  for(定义循环变量并赋初值;循环条件;循环变量的更新){  
2      //循环操作  
3  }
```

2. 执行流程图



3. 案例

求1~10的累加和

4. 代码实现

```
1  public class Example8 {  
2  
3      public static void main(String[] args) {  
4          int total = 0;  
5          int i = 1;  
6          while(i <= 10){  
7              total += i;  
8              i++;  
9          }  
10         System.out.println(total);  
11  
12         int sum = 0; //和, 默认为0  
13         for(int m = 1; m <= 10; m++){ //变量m的作用范围仅限于整个for循环结构  
14             sum += m;  
15         }  
16         System.out.println(sum);  
17     }  
18 }
```

[演示 for 循环执行过程](#)

[断点调试](#)

[练习](#)

1. 求6的阶乘 ($6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6$)

```

1 public class Example9 {
2
3     public static void main(String[] args) {
4         int result = 1;
5         for(int i=1; i<=6; i++){
6             result = result * i;
7             // result *= i;
8         }
9         System.out.println(result);
10    }
11 }

```

2. 求100以内既能被2整除又能被9整除的所有整数的和

```

1 public class Example10 {
2
3     public static void main(String[] args) {
4         int sum = 0;
5         for(int i=1; i<100; i++){
6             if((i % 2 == 0) && (i % 9 == 0)){
7                 sum = sum + i;
8                 // sum += i;
9             }
10        }
11        System.out.println(sum);
12    }
13 }

```

5. 总结

for 循环的特征是先判断，后执行；如果一开始条件就不满足，那么 for 循环可能一次也不执行。循环次数确定的情况下，通常使用 for 循环；循环次数不确定的情况下通常使用 while 循环和 do-while 循环

第六节 流程控制

1. break关键字

应用场景

break只能应用于 while 循环、do-while 循环、for 循环和 switch 选择结构

作用

break 应用在循环结构中时，表示终止 break 所在的循环，执行循环结构下面的代码，通常与 if 选择结构配合使用

break 应用在 switch 选择结构时，表示终止 break 所在的 switch 选择结构

案例

获取一个10以内的随机整数，然后从控制台输入一个整数，如果输入的整数与随机整数不相等，则重新输入，直到输入的整数与随机整数相等为止

代码实现

```

1 public class Example11 {
2
3     public static void main(String[] args) {
4         //Math.random(); 表示的意思是随机获取一个0~1之间的随机浮点数，能够取到0，但是
        取不到1
5         //[0,1)
6         double random = Math.random();// [0,1)浮点数
7         //    int number = (int)(random * 10); //[0, 10)浮点数
8         double number = random * 10; //[0, 10)浮点数
9         //[10,20) random * 10 + 10
10        int randomNumber = (int)number;
11        Scanner sc = new Scanner(System.in);
12        //    int inputNumber; //定义循环变量，没有赋初值
13        //    do{
14        //        System.out.println("请输入一个0~10之间的整数: ");
15        //        inputNumber = sc.nextInt();
16        //    }while (randomNumber != inputNumber);
17        while(true){//死循环
18            System.out.println("请输入一个0~10之间的整数: ");
19            int inputNumber = sc.nextInt();
20            if(randomNumber == inputNumber){
21                break;//循环中使用break表示终止break所在的循环
22            }
23        }
24    }
25 }

```

练习

从控制台输入一个数字，判断该数字是否是素数（素数的特征：只能被1和本身整除，如素数3 只能被1和3整除）。要求使用 break 实现

```

1 public class Example12 {
2
3     public static void main(String[] args) {
4         // 5  2 3 4
5         // 6  2 3 4 5
6         // 7  2 3 4 5 6
7         Scanner sc = new Scanner(System.in);
8         System.out.println("请输入一个整数: ");
9         int number = sc.nextInt();
10        boolean isPrime = true; //任何数都默认是素数
11        for(int i=2; i<number; i++){
12            //只要2~number-1范围内，任意一个数能够被number整除，
13            //则说明该数不是素数
14            if(number % i == 0){
15                isPrime = false;
16                break;
17            }
18        }
19        if(isPrime){
20            System.out.println(number + "是素数");
21        } else {
22            System.out.println(number + "是和数");
23        }
24    }
25 }

```


2. continue关键字

应用场景

continue只能应用在循环结构中（while循环、do-while循环和for循环）

作用

表示跳过本次循环，进入下一次循环，通常与if选择结构配合使用

案例

打印1~10之间的所有偶数

代码实现

```
1 public class Example13 {
2
3     public static void main(String[] args) {
4         for(int i=1; i<=10; i++){
5             //         if(i % 2 == 0){
6             //             System.out.println(i);
7             //         }
8             //当i是奇数时，跳过本次循环，直接进入下一次循环
9             if(i % 2 == 1) continue; //如果if语句后面只有一条语句，那么{}可以省略
10            System.out.println(i);
11        }
12
13        int m = 1;
14        while (m <= 10){
15            if(m % 2 == 1) {
16                m++;
17                continue;
18            }
19            System.out.println(m);
20            m++;
21        }
22    }
23 }
```

练习

从控制台录入一位学生的成绩，如果成绩低于60分，则将输入的成绩加5分，知道成绩及格为止。要求使用 continue 实现。

```
1 public class Example14 {
2
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         System.out.println("请输入学生成绩: ");
6         int score = sc.nextInt();
7         if(score < 60){
8             while (true){
9                 score += 5;
10                if(score < 60) continue;
11                else break;
12            }
13        }
14    }
15 }
```

```
12         }
13     }
14 }
15 }
```