第3章

Java 程序流程控制

流程控制是所有编程语言的基本功能，主要是指控制程序中各语句的执行顺序。在 Java 语言中，最主要的流程控制方式是结构化程序设计中规定的 3 种基本控制结构。Java 程序通过控制语句来控制方法的执行流程。本章主要介绍选择结构、循环结构和 Java 方法，并通过猜数游戏，学习程序流程控制和 Java 方法的应用。

通过本章的学习，达到以下目标。

- 理解逻辑值，能运用关系表达式和逻辑表达式实现真假判断。
- 能使用 if 语句、switch 语句编写分支结构程序。
- 能运用分支结构等编写打折计价、显示星座、判断成绩等级应用程序。
- 学会使用 for、while 和 do-while 循环语句，理解递归调用方法。
- 能运用循环结构编写计划累加、阶乘以及乘法表等应用程序。
- 学会定义方法和调用方法，理解变量和字段的作用域。
- 能编写方法来计算圆、矩形的面积和周长。

3.1 程序基本控制结构

面向对象程序结构有 3 种：顺序结构、分支结构和循环结构。顺序结构按从上到下的顺序逐条执行语句，上一条语句执行完毕，接着执行下一条语句，直到程序结束。

3.2 选择结构

选择结构也称分支结构，一般由两个分支组成，程序流程图如图 3-1 所示。根据条件是否成立，选择执行一个分支的语句。各分支中的语句可以是多条语句组成的代码块，也可以是不包含语句的空语句（有一个分支是空语句的，称为单分支结构），如图 3-2 所示。

3.2.1 if 选择结构

if 选择结构是单分支结构，程序流程图如图 3-2 所示，其语法形式如下所示。

```
if(条件表达式){
  代码块
}
```

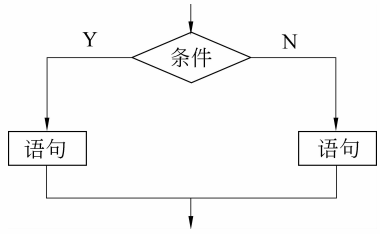

图 3-1　分支结构

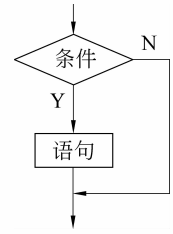

图 3-2　单分支结构

说明：条件表达式必须是一个布尔表达式。如果条件中的值为 true，执行代码块，否则跳过。

条件表达式是返回逻辑值的关系表达式或逻辑表达式，必须用圆括号括起来。代码块可以是一条语句，也可以是多条语句，必须用大括号括起来。

【例 3-1】 比较数的大小。

```
public class Test{
    public static void main(String[] args){
          int a =2;
          if(a <5){
              System.out.print("a 小于 5");
          }
    }
}
```

程序输出结果：

```
a 小于 5
```

3.2.2　if-else 选择结构

最常用的分支语句是 if-else，程序流程图如图 3-2 所示，其语法形式如下。

```
if(条件表达式){
    代码块 1
}else{
    代码块 2
}
```

说明：如果条件表达式中的值为 true，执行代码块 1，否则执行代码块 2。

【例 3-2】 选择结构的应用例子。

```
public class Test{
    public static void main(String[] args){
        int a=5;
        if(a<3){
            System.out.print("a 小于 b");
        }else{
```

```
            System.out.print("a 大于 b");
        }
    }
}
```

程序输出结果：

```
a 大于 b
```

3.2.3 多重 if 选择结构

多重 if 选择结构语法形式如下。

```
if(条件表达式 1){
    代码块 1
}else if(条件表达式 2){
    代码块 2
}else{
    代码块 3
}
```

说明：若需要判断的条件是连续的区间，使用多重 if 选择结构有很大优势。else if 块可以有多个，取决于程序的需要。如果条件表达式 1 为 true，执行代码块 1，否则执行 else if 块，判断条件表达式 2；为 true 时，执行代码块 2，否则执行代码块 3，以此类推。当条件满足某个 else if 块时，程序余下的部分将不再执行，直接跳出 if 块。

【例 3-3】 比较数的大小。

```
public class Test{
    public static void main(String[] args){
        int a=5;
        if(a>2){
            System.out.print("a");
        }else if(a>3){
            System.out.print("b");
        }else if(a>4){
            System.out.print("c");
        }
    }
}
```

程序输出结果：

```
a
```

【例 3-4】 编写程序实现下述内容：某超市搞促销活动，购买商品总价 2000 元以上，打 8 折；总价 1000～2000 元，打 8.5 折；总价 500～1000 元，打 9 折；总价不到 500 元，不打折。

```
import java.util.Scanner;
public class Example4{
    public static void main(String[]args){
        double price,discount,discPrice;
        Scanner scan=new Scanner(System.in);
        System.out.println("====打折计价====");
        System.out.println("购买商品 2000 元以上,8 折优惠");
        System.out.println("购买商品 1000-2000 元,8.5 折优惠");
        System.out.println("购买商品 500-1000 元,9 折优惠");
        System.out.print("请输入购买商品的价格:");
        price=scan.nextDouble();
        if(price>=2000){ discount=0.8;}
        else  if(price>=1000){ discount=0.85;}
        else  if(price>=500){ discount=0.9;}
        else  if(price>0){ discount=1;}
        else{
            System.out.println("输入数据有问题");
            return;
        }
        discPrice=price * discount;
        System.out.printf("%.2f 折,折扣价为$%.2f",discount,discPrice);
    }
}
```

程序说明：该程序嵌套了 4 层 if 语句,因而有 5 个分支,即有 5 种不同的运行路线。多次运行程序,其中 4 次的运行结果如图 3-3 所示。

图 3-3　打折计价程序 4 次运行结果

3.2.4　嵌套 if 选择结构

嵌套 if 选择结构语法形式如下。

```
if(条件表达式 1){
    if(条件表达式 2){
        代码块 1
```

```
    }else{
        代码块 2
    }
}else{
    代码块 3
}
```

说明：该结构其实就是在 if 选择结构里嵌入 if 选择结构。条件表达式 1 为 false 时，执行代码块 3，否则执行内部 if 选择结构。也就是说，要执行代码块 1，必须满足条件表达式 1 及条件表达式 2。

【例 3-5】 比较数的大小。

```
public class Test{
    public static void main(String[] args){
        int a=3;
        if(a>2){
            if(a !=3){
                System.out.print("a!=3");              //代码块 1
            }else{
                System.out.print("a=3");               //代码块 2
            }
        }else{
            System.out.print("a>2");                   //代码块 3
        }
    }
}
```

程序输出结果：

```
a= 3
```

3.2.5 switch 选择结构

switch 选择结构语法格式如下。

```
switch(表达式){
    case 常量 1:
        代码块 1;
        break;
    case 常量 2:
        代码块 2;
        break;
    default:
        代码块 3;
        break;
}
```