

程序流程控制

同其他程序设计语言一样,Java 使用选择结构和循环结构确定控制流程。Java 流程控制语句有 3 种:条件语句、循环语句和转移语句。

3.1 块作用域

块作用域,也称复合语句,是由花括号({})括起来的若干 Java 语句组成的。块确定了变量的作用域,一个块可以嵌套在另一个块中。

注意: 不能在嵌套的两个块中声明同名的变量。

示例 3-1

```
public static void main(String[] args) {  
    int n;  
    {  
        int b;  
    }  
}
```

3.2 条件语句



条件语句

条件语句提供程序判断能力,其可使部分程序根据某些表达式的值被有选择性地执行。Java 提供两种条件语句:if 语句和 switch 语句。

3.2.1 if 语句

在 Java 中,if 语句是构造分支选择结构的基本语句。根据 if 语句的语法形式可分为 3 种类型:单分支选择结构、双分支选择结构、嵌套 if 语句多分支结构。

1. 单分支选择结构

单分支选择结构的语法格式如下:

if (判断条件)
语句;

与大多数程序设计语言一样,Java 希望在某个条件为真时执行多条语句。此时需要使用块语句实现。

```
if (判断条件) {  
    语句 1;  
    语句 2;  
    ...  
}
```

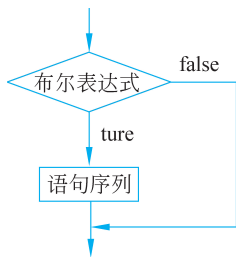


图 3-1 单分支选择结构执行流程

其执行流程如图 3-1 所示。

如果布尔表达式为 true 就执行语句序列,否则执行 if 结构后面的语句。其中:

(1) 布尔表达式。必要参数,表示最后返回的结果必须是一个布尔值。既可使用一个布尔变量或常量,也可使用关系运算符或布尔运算符结合操作数组成的表达式。

(2) 语句序列。可选参数,可是一条或多条语句。当布尔表达式的值为 true 时执行语句序列;若语句序列仅有一条语句,可省略语句中的 {}。

程序示例 3-1 chapter3 /chapter3_1.java

```
package com.cumtb;  
import java.util.Scanner;  
public class chapter3_1 {  
    public static void main(String[] args) {  
        Scanner inScanner = new Scanner(System.in);  
        int score = inScanner.nextInt();  
        if (score >= 60) {  
            System.out.println("及格");  
        }  
        if (score < 60) {  
            System.out.println("不及格");  
        }  
    }  
}
```

2. 双分支选择结构

双分支选择结构的语法格式如下:

```
if (判断条件) {  
    语句 1;  
}  
else {  
    语句 2;  
}
```

其执行流程如图 3-2 所示。

如果布尔表达式的值为 true 时,程序执行语句 1,否则程序执行语句 2。因此,在双分支选择结构中,每个时刻只能执行一个分支。

程序示例 3-2 chapter3 /chapter3_2.java

```
package com.cumtb;
import java.util.Scanner;
public class chapter3_2 {
    public static void main(String[] args) {
        Scanner inScanner = new Scanner(System.in);
        int score = inScanner.nextInt();
        if (score >= 60) {
            System.out.println("及格");
        } else {
            System.out.println("不及格");
        }
    }
}
```

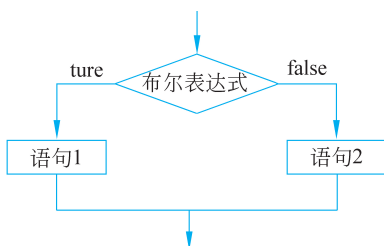


图 3-2 多分支选择结构执行流程

3. 嵌套 if 语句多分支结构

在实际问题中,一个简单的条件无法满足复杂问题的设计,需要有若干条件来决定执行若干不同的操作。Java 提供嵌套 if 语句来实现,其语法格式如下:

```
if (判断条件){
    语句 1;
}
else if (判断条件){
    语句 2;
}
else
    语句 3;
```

其执行流程如图 3-3 所示。

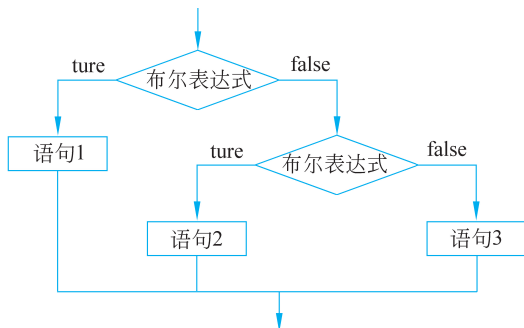


图 3-3 嵌套 if 语句多分支结构执行流程

如图 3-3 所示,嵌套 if 语句可实现多个执行分支。需要注意的是,虽然图 3-3 中只有 3 个执行分支,但嵌套 if 结构可以嵌套至 $n(n \geq 1)$ 层。

程序示例 3-3 chapter3 /chapter3_3.java

```
package com.cumtb;
import java.util.Scanner;
public class chapter3_3 {
    public static void main(String[] args) {
        Scanner inScanner = new Scanner(System.in);
        int score = inScanner.nextInt();
        if (score >= 90) {
            System.out.println("优秀");
        } else if (score >= 80) {
            System.out.println("良好");
        } else if (score >= 70) {
            System.out.println("中等");
        } else if (score >= 60) {
            System.out.println("及格");
        } else {
            System.out.println("不及格");
        }
    }
}
```

使用嵌套 if 语句的注意事项如下。

(1) Java 编译器将 else 与离它最近的 if 组合在一起,除非使用花括号才能指定不同的匹配方式。

(2) 在嵌套 if 语句中,当每个语句块包含多条语句时,必须使用花括号括起来,使其构成一条复合语句,否则导致语法错误或输出错误。

程序示例 3-4 chapter3 /chapter3_4.java

```
package com.cumtb;
import java.util.Scanner;
public class chapter3_4 {
    public static void main(String[] args) {
        Scanner inScanner = new Scanner(System.in);
        int score = inScanner.nextInt();
        if (score > 60)
            if (score > 70) {
                System.out.println("良好");
            }
        else {
            System.out.println("不及格");
        }
    }
}
```

当程序示例 3-4 执行时,当 $\text{score} > 60$ 且 $\text{score} \leq 70$ 时,输出“不及格”信息,而当 $\text{score} < 60$ 时没有信息输出。显然,程序执行的结果与预期结果不相符。

3.2.2 switch 语句

虽然使用嵌套 if 语句能够实现从多分支中选择一个分支执行操作,但当嵌套层数较多时,程序可读性大大降低。

Java 提供 switch 语句提供多分支程序结构语句,其可根据表达式的值来执行多个操作中的一个。switch 语句的语法格式如下:

```
switch(表达式) {  
    case 1: 语句 1; break;  
    case 2: 语句 2; break;  
    ...  
    case n: 语句 n; break;  
    [default: 语句 n+1; ]  
}
```

其执行流程如图 3-4 所示。

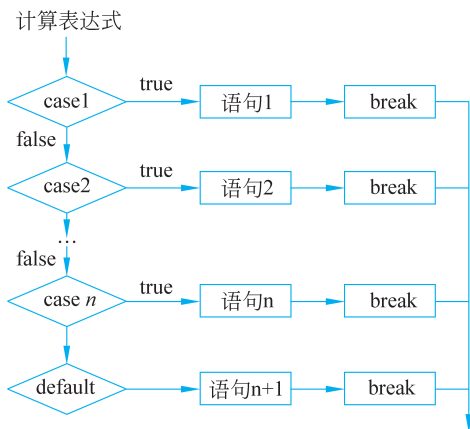


图 3-4 switch 语句执行流程

使用 switch 语句的注意事项如下。

- (1) 表达式类型可以是 char、byte、short 和 int,不允许使用 float 类型和 long 类型。
- (2) case 后面的值必须是与表达式类型相同的常量,且它们之间的值各不相同。
case 后面语句无须使用花括号。
- (3) default 语句可删除。
- (4) 当表达式的值与某个 case 的常量值匹配成功后,就执行此 case 后的语句。
- (5) 若删除 case 语句后的 break 语句,则执行完第一个匹配 case 语句后,会继续执行其余的 case 语句(无须匹配),直到 break 为止。

程序示例 3-5 chapter3 /chapter3_5.java

```
package com.cumtb;  
import java.util.Scanner;  
public class chapter3_5 {  
    public static void main(String[] args) {  
        Scanner inScanner = new Scanner(System.in);  
        int score = inScanner.nextInt();  
        int grade = score / 10;  
        switch (grade) {  
            case 10:  
            case 9:  
                System.out.println("优秀");  
                break;  
            case 8:  
            case 7:  
                System.out.println("良好");  
                break;  
            case 6:  
                System.out.println("及格");  
                break;  
            default:  
                System.out.println("不及格");  
                break;  
        }  
    }  
}
```



循环语句



3.3 循环语句

循环语句能够使程序代码重复执行。Java 支持 3 种循环结构类型：while、do...while 和 for。上述循环结构各具特点，循环结构之间也可相互嵌套，用户可根据不同的程序设计需求来决定使用何种循环结构或循环嵌套结构。

3.3.1 while 语句

while 语句是一种先判断的循环结构，其语法格式如下：

```
while(循环条件) {  
    循环体语句块  
}
```

其执行流程如图 3-5 所示。

使用 while 语句的注意事项如下。

(1) while 循环没有初始化语句，循环次数不可知，只要

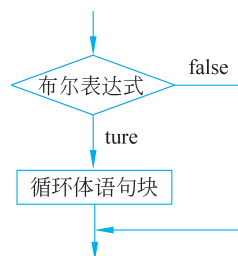


图 3-5 while 语句执行流程

循环条件满足,循环就会一直进行。

(2) while 循环条件语句中只能写一个表达式,且是布尔表达式。

(3) 如果循环体中需要循环变量,必须在 while 语句之前对循环变量进行初始化。

程序示例 3-6 chapter3 /chapter3_6.java

```
package com.cumtb;
import java.util.Scanner;
public class chapter3_6 {
    public static void main(String[] args) {
        Scanner inScanner = new Scanner(System.in);
        System.out.println("请输入数值:");
        int i = 1;
        while (i <= 3) {
            int score = inScanner.nextInt();
            System.out.println("第"+i+"次输出值:"+score);
            i++;
        }
    }
}
```

3.3.2 do...while 语句

do...while 语句的使用与 while 语句相似,但是,do...while 语句是事后判断的循环结构,其语法格式如下:

```
do{
    循环体语句块
}while(循环条件);
```

其执行流程如图 3-6 所示。

使用 do...while 语句时需要注意: do...while 语句没有初始化语句,循环次数不可知,无论循环条件是否满足,都会执行一次循环体,然后再判断循环条件。

程序示例 3-7 chapter3 /chapter3_7.java

```
package com.cumtb;
import java.util.Scanner;
public class chapter3_7 {
    public static void main(String[] args) {
        Scanner inScanner = new Scanner(System.in);
        System.out.println("请输入数值:");
        int i = 1;
        do {
            int score = inScanner.nextInt();
            System.out.println("第"+i+"次输出值:"+score);
        }
```

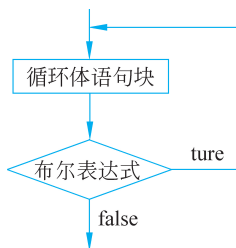


图 3-6 do...while 语句执行流程

```

        i++;
    } while (i <= 3);
}

```

3.3.3 for 语句

for 语句是应用最广泛、功能最强的循环语句,其语法格式如下:

```

for (表达式 1; 表达式 2; 表达式 3) {
    循环体语句块
}

```

其中,表达式 1 指初值表达式;表达式 2 指布尔表达式;表达式 3 指循环过程表达式。其执行流程如图 3-7 所示。

使用 for 语句的注意事项如下。

(1) for 语句中的 3 个表达式都可省略,但两个分号不能省略。

(2) 在 for 语句中省略任何一个表达式时,应将其写入程序中的其他位置,否则会出现死循环的问题。

(3) 在 for 语句中,表达式 1 和表达式 3 可使用逗号运算符,是 Java 中唯一使用逗号运算符的位置。

程序示例 3-8 chapter3 /chapter3_8.java

```

package com.cumt.b;

import java.util.Scanner;

public class chapter3_8 {
    public static void main(String[] args) {
        Scanner inScanner = new Scanner(System.in);
        System.out.println("请输入数值: ");
        for (int i = 1; i <= 3; i++) {
            int score = inScanner.nextInt();
            System.out.println("第"+i+"次输出值:"+score);
        }
    }
}

```

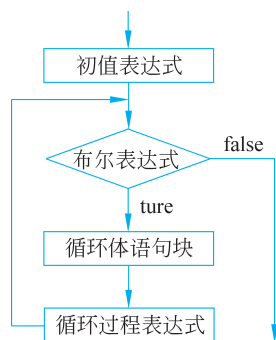


图 3-7 for 语句执行流程



3.4 转移语句



转移语句

Java 提供 4 种转移语句: break、continue、return 和 throw。转移语句完成改变程序执行流程的功能。本节重点介绍 break 和 continue 语句的使用,return 和 throw 语句的使用将在 11.7 节中详细介绍。

3.4.1 break 语句

break 语句有两种形式：不带标签和带标签。break 语句虽然可以单独使用，但在通常情况下，其主要用于 switch 语句和循环语句中，控制程序的执行流程的转移。

1. 不带标签

不带标签的 break 语句使程序跳出所在层的循环体，其语法格式如下：

break;

其中，break 是关键字。

程序示例 3-9 chapter3 /chapter3_9.java

```
package com.cumtb;
public class chapter3_9 {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i++) {
            for (int j = 5; j > 0; j--) {
                if (j == i) {
                    break;
                }
                System.out.printf("(i,j)=(%d,%d)", i, j);
                System.out.println();
            }
        }
    }
}
```

2. 带标签

带标签的 break 语句使程序跳至标签指示的循环体，其语法格式如下：

break label;

其中，break 是关键字，label 是用户定义的标号。

程序示例 3-10 chapter3 /chapter3_10.java

```
package com.cumtb;
public class chapter3_10 {
    public static void main(String[] args) {
        label: for (int i = 0; i < 3; i++) {
            for (int j = 3; j > 0; j--) {
                if (j == i) {
                    break label;
                }
            }
            System.out.printf("(i,j)=(%d,%d)", i, j);
        }
    }
}
```

```

        System.out.println();
    }
}
}
}

```

通过分析程序示例 3-9 和程序示例 3-10 的运行结果发现：不带标签的 break 语句执行 6 次，带标签的 break 语句执行 5 次。因此在程序设计中，添加标签对于多层嵌套循环而言，可以提高程序的执行效率。

3.4.2 continue 语句

continue 语句用于结束本次循环，跳过循环体中尚未执行的语句，进行终止条件的判断来决定是否继续循环。在循环体中使用 continue 语句有两种形式：不带标签和带标签。

1. 不带标签

不带标签的 continue 语句终止本次循环，循环体中 continue 之后的语句不再执行，接着进行下一次循环，其语法格式如下：

continue;

其中，continue 是关键字。

程序示例 3-11 chapter3 /chapter3_11.java

```

package com.cumtb;

public class chapter3_11 {
    public static void main(String[] args) {
        for (int i = 0; i < 3; i++) {
            for (int j = 3; j > 0; j--) {
                if (j == i) {
                    continue;
                }
                System.out.printf("(i,j)=(%d,%d)", i, j);
                System.out.println();
            }
        }
    }
}

```

2. 带标签

带标签的 continue 语句终止本次循环，当前循环体中 continue 之后的语句不再执行，而跳转至指定标签位置，进行执行标签指定的循环，其语法格式如下：

continue label;

其中,continue 是关键字,label 是用户定义的标号。

程序示例 3-12 chapter3 /chapter3_12.java

```
package com.cumtb;
public class chapter3_12 {
    public static void main(String[] args) {
        label: for (int i = 0; i < 3; i++) {
            for (int j = 3; j > 0; j--) {
                if (j == i) {
                    continue label;
                }
                System.out.printf("(i,j)=(%d,%d)", i, j);
                System.out.println();
            }
        }
    }
}
```

3.5 课后习题

1. Java 程序中有哪些基本流程? 分别对应哪些语句?
2. Java 中如何直接跳出当前的多重循环?
3. switch 语句中,case 语句后常量表达式值是什么类型的,可以是实数吗?
4. if 后面的复合语句块只有一条语句,是否能省略花括号?
5. 请找出下面程序的错误,并说明运行该程序会出现什么现象。

```
public class GetX{
    public static viod main() {
        while(x == 5) {
            System.out.println("x 的值为 5");
        }
    }
}
```

6. 下面表达式不能()作为 while 语句的表达式。
A. $m > 0$ B. $m++ > 0$ C. $m = 0$ D. $m > 10 \& \& \text{true}$
7. 在 Java 中常用的循环语句有几种? 分别进行说明。
8. 简要说明 break 语句和 continue 语句在程序中的作用。
9. 编写程序,用 for...each 遍历数组 [9,5,6,3,1],并输出结果。