



条件语句能够使计算机程序具有“判断能力”，像人类的大脑一样分析问题，并根据某些表达式的值有选择地执行语句。Java 语言提供了两种条件语句：if 语句和 switch 语句。

5.1 if 语句

由 if 语句引导的选择结构有 if 结构、if-else 结构和 if-else-if 结构三种。

5.1.1 if 结构

if 结构流程示意图如图 5-1 所示，首先测试条件表达式，如果为 true 则执行语句组（包含一条或多条语句代码块），否则执行 if 语句组后面的语句。

提示：如果语句组只有一条语句，可以省略大括号，但从编程规范的角度考虑，建议不要省略大括号，否则会使程序的可读性变差。

if 结构语法格式如下：

```
if (条件表达式) {  
    语句组  
}
```

if 结构示例代码如下：

```
package exercise5_1_1;  
//5.1.1 if 结构  
import java.util.Scanner; // 导入 Scanner 类 ①  
  
public class Main {  
    public static void main(String args[] ) {  
        Scanner in = new Scanner(System.in); // 创建 Scanner 对象 ②
```

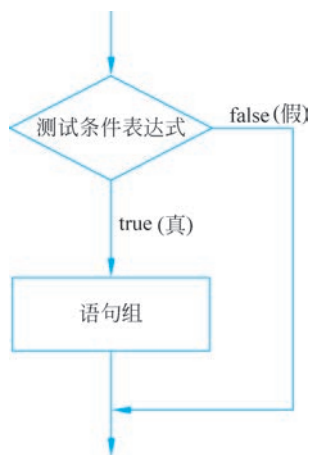


图 5-1 if 结构流程示意图



微课视频

```
System.out.println("请输入一个整数:");
int score = in.nextInt();           // 读取从键盘输入的字符串并转换为
                                   // int 类型数据 ③

if (score >= 85) {
    System.out.println("您真优秀!");
}
if (score < 60)                      ④
    System.out.println("您需要加倍努力!");

if ((score >= 60) && (score < 85)) {
    System.out.println("您的成绩还可以,仍需继续努力!");
}
}
}
```

上述程序运行到代码第②行会挂起,等待用户输入,如图 5-2 所示。输入内容后按 Enter 键,程序将继续运行,如图 5-3 所示。

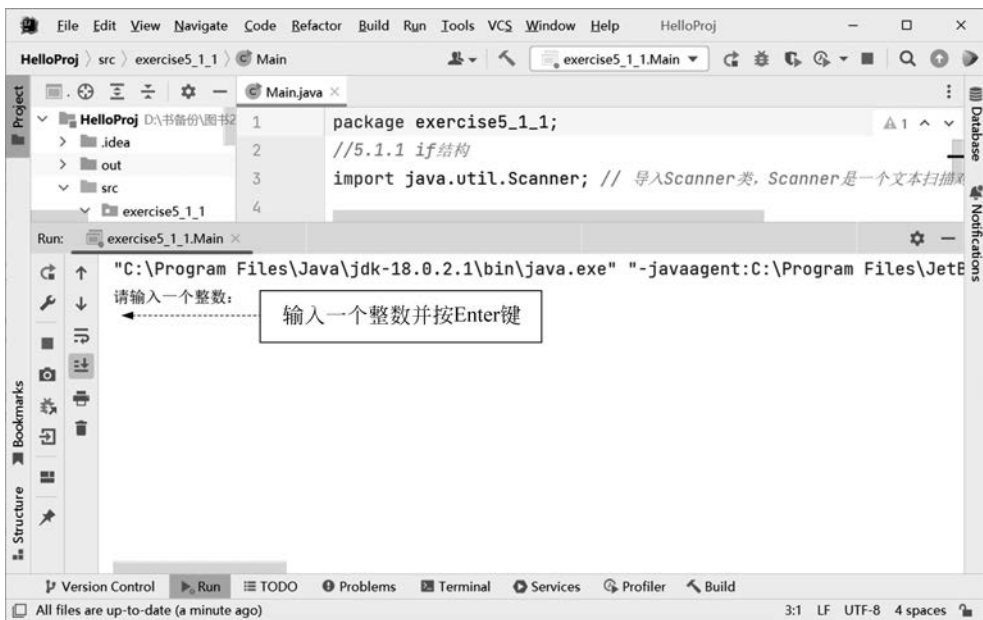


图 5-2 程序挂起等待用户输入

上述代码第①行导入 Scanner 类,该类是一个文本扫描对象。

上述代码第②行创建 Scanner 对象时,需要参数 System.in, System.in 是标准输入流,默认是键盘。

上述代码第③行 in.nextInt() 方法是从键盘读取字符串并转换为 int 类型数据。

另外,代码第④行的 if 语句中的语句组只有一条语句,省略大括号。

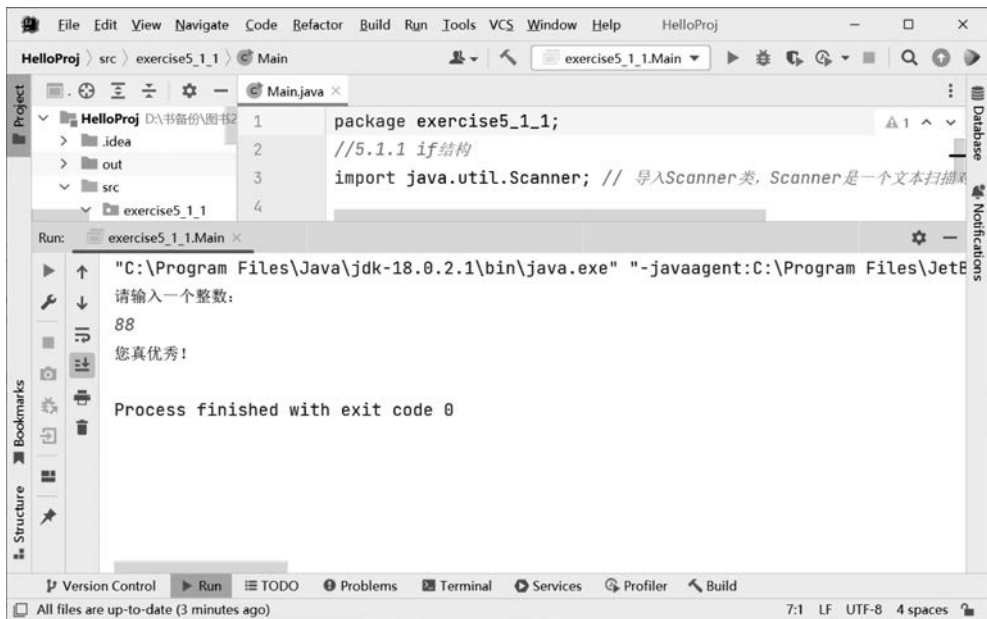


图 5-3 程序继续执行

5.1.2 if-else 结构

if-else 结构流程示意图如图 5-4 所示,首先测试条件表达式,如果值为 true,则执行语句组 1; 如果条件表达式值为 false,则忽略语句组 1,直接执行语句组 2,然后继续执行后面的语句。

if-else 结构语法格式如下:

```
if (条件表达式) {
    语句组 1
} else {
    语句组 2
}
```

if-else 结构示例代码如下:

```
package exercise5_1_2;
//5.1.2 if-else 结构
import java.util.Scanner;
```

```
public class Main {
    public static void main(String args[] ) {
        Scanner in = new Scanner(System.in); // 创建 Scanner 对象
        System.out.println("请输入一个整数:");
        int score = in.nextInt();           // 读取文本并转换为 int 类型
```

```
// 导入 Scanner 类,Scanner 是一个文本扫描对象
```

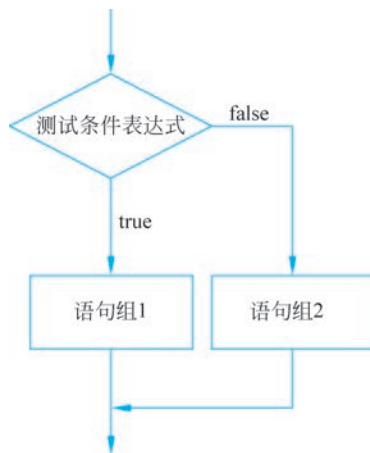


图 5-4 if-else 结构流程示意图



微课视频

```
    if (score < 60) {  
        System.out.println("不及格");  
    } else {  
        System.out.println("及格");  
    }  
}
```

上述代码与 5.1.1 节类似,这里不再赘述。

5.1.3 if-else-if 结构

如果有多个分支,则可以使用 if-else-if 结构,它的流程示意图如图 5-5 所示。if-else-if 结构实际上是 if-else 结构的多层嵌套,特点是在多个分支中只执行一个语句组,而其他分支都不执行,所以这种结构可以用于有多种判断结果的分支中。

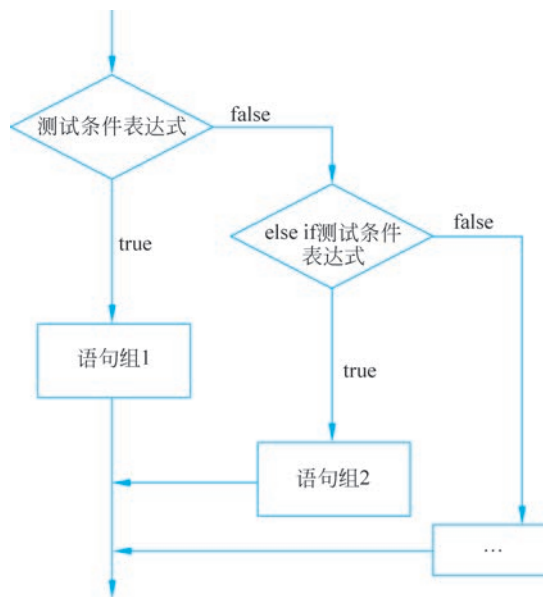


图 5-5 if-else-if 结构流程示意图

if-else-if 结构语法格式如下:

```
if (条件表达式 1) {  
    语句组 1  
} else if (条件表达式 2) {  
    语句组 2  
} else if (条件表达式 3) {  
    语句组 3  
} ...  
} else if (条件表达式 n) {  
    语句组 n  
} else {
```



微课视频

```
    语句组 n + 1
}
```

if-else-if 结构示例代码如下：

```
package exercise5_1_3;
//5.1.3 if - else - if 结构

import java.util.Scanner; // 导入 Scanner 类,Scanner 是一个文本扫描对象
public class Main {
    public static void main(String args[] ) {
        Scanner in = new Scanner(System.in); // 创建 Scanner 对象
        System.out.println("请输入一个整数:");
        int score = in.nextInt(); // 读取文本并转换为 int 类型
        char grade;
        if (score >= 90) {
            grade = 'A';
        } else if (score >= 80) {
            grade = 'B';
        } else if (score >= 70) {
            grade = 'C';
        } else if (score >= 60) {
            grade = 'D';
        } else {
            grade = 'F';
        }
        System.out.println("分数等级:" + grade);
    }
}
```

上述代码与 5.1.1 节类似,这里不再赘述。

5.2 多分支语句

事实上,如果分支很多,那么 if-else-if 结构使用起来也很麻烦,这时可以使用 switch 语句,它提供多分支程序结构语句。

5.2.1 switch 语句

最早的 switch 语句是从 C 和 C++ 语言继承而来的。下面先介绍 switch 语句的基本语法结构,如下所示:

```
switch (表达式) {
    case 值 1:
        语句组 1
    case 值 2:
        语句组 2
    case 值 3:
        语句组 3
```



```
        :
    case 值 n:
        语句组 n
    default:
        语句组 n+1
}
```

其中, default 语句可以省略。switch 语句中“表达式”计算结果只能是如下几种类型:

- (1) byte、short、char 和 int 类型。
- (2) Byte、Short、Character 和 Integer 等包装类。
- (3) String 类型。
- (4) 枚举类型。

Java 中有 8 个包装类对应 Java 中 8 种基本数据类型。有关包装类将在后面章节详细介绍。

当程序执行到 switch 语句时,先计算条件表达式的值,假设值为 A,然后将 A 与第 1 个 case 语句中的值 1 进行匹配,如果匹配则执行“语句组 1”,执行完成后不跳出 switch,只有遇到 break 才跳出 switch。如果 A 没有与第 1 个 case 语句匹配,则与第 2 个 case 语句进行匹配,如果匹配则执行“语句组 2”,以此类推,直到执行“语句组 n”。如果所有 case 语句都没有执行,就执行 default 的“语句组 n+1”,这时才跳出 switch。

1. 表达式计算结果是 int 类型示例

```
package exercise5_2_1;
//5.2.1 switch 语句
//1. 表达式计算结果是 int 类型示例

import java.util.Scanner; // 导入 Scanner 类,Scanner 是一个文本扫描对象

public class Main_1 {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in); // 创建 Scanner 对象
        System.out.println("请输入≤100 的一个整数:");
        int score = in.nextInt(); // 读取文本并转换为 int 类型
        String grade;

        switch (score / 10) {
            case 10, 9: // 9 和 10 是同一个分支,常量值用逗号分隔
                grade = "A";
                break;
            case 8:
                grade = "B";
                break;
            case 7:
                grade = "C";
                break;
        }
    }
}
```

```

        case 6:
            grade = "D";
            break;
        case 5:
            grade = "E";
            break;
        default:
            grade = "未知";
    }
    System.out.println("分数等级:" + grade);
}
}

```

2. 表达式计算结果是 String 类型示例

```

package exercise5_2_1;
//5.2.1 switch 语句
//2. 表达式计算结果是 String 类型示例
import java.util.Scanner;
public class Main_2 {
    public static void main(String args[] ) {
        Scanner in = new Scanner(System.in); // 创建 Scanner 对象
        System.out.println("请输入级别:");
        String level = in.next();           // 读取文本
        String desc = "";
        switch (level) {
            case "优":
                desc = "90 分以上";
                break;
            case "良":
                desc = "80~89 分";
                break;
            case "中":
                desc = "60~79 分";
                break;
            case "差":
                desc = "低于 60 分";
                break;
            default:
                desc = "无法判断";
        }
        System.out.println(desc);
    }
}

```

上述示例运行时,用户通过键盘输入“优”“良”“中”和“差”等字符,然后执行 switch 语句选择返回结果,如图 5-6 所示。

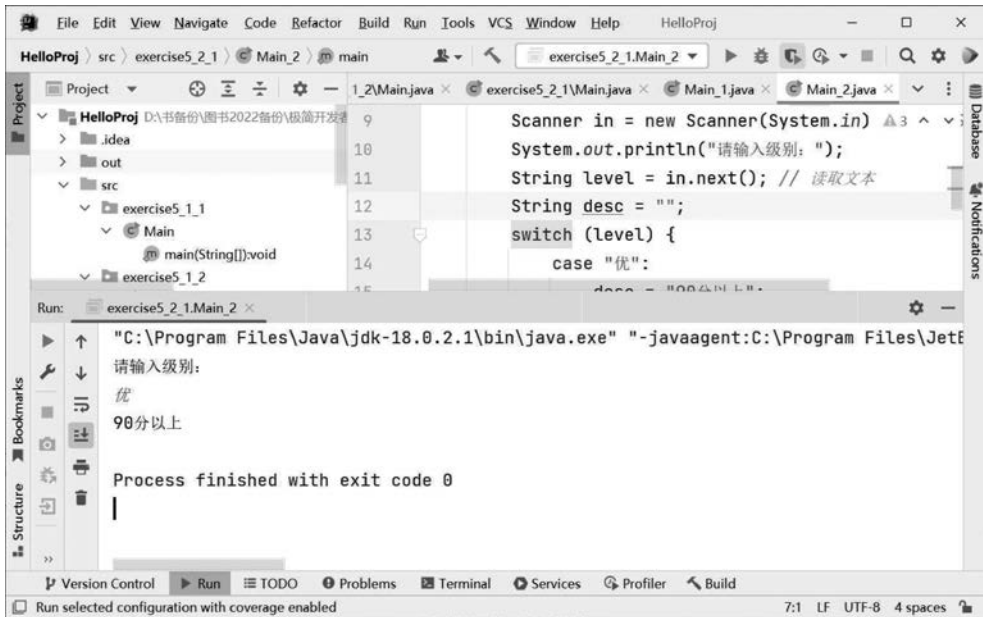


图 5-6 运行结果



微课视频

5.2.2 switch 表达式

如果希望根据不同的分支选择返回单个值,则可以使用 switch 表达式。Java 14 推出了 switch 表达式,它是在 case 后面使用箭头运算符(->)替代 break 语句。使用箭头符号(->),每个 case 执行完成后结束 switch 表达式;其次,case 后面可以有多个常量,常量之间用逗号(,)分隔。

 **提示:** 表达式可以出现在赋值符号(=)的右边,它会返回一个计算结果。

示例代码如下:

```
package exercise5_2_2;
//5.2.2 switch 表达式

import java.util.Scanner; // 导入 Scanner 类,Scanner 是一个文本扫描对象

public class Main {
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in); // 创建 Scanner 对象
        System.out.println("请输入一个小于或等于 100 的整数:");
        int score = in.nextInt(); // 读取文本并转换为 int 类型
        // 声明变量 grade 接收 switch 表达式返回的结果
        String grade = switch (score / 10) {
```

```
        case 10, 9 -> "优";
        case 8 -> "良";
        case 7, 6 -> "中";
        case 1, 2, 3, 4, 5 -> "差";
        default -> "未知";
    };
    System.out.println("Grade = " + grade);
}
}
```

上述代码根据用户输入的整数选择分支,并将结果返回给变量 grade,运行结果如图 5-7 所示。

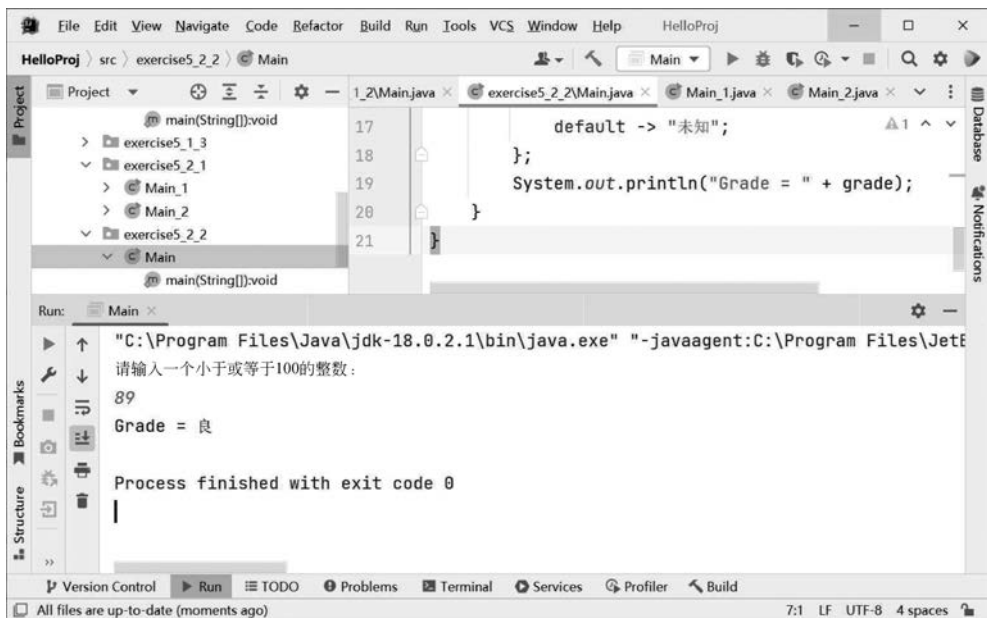


图 5-7 运行结果

5.3 动手练一练

1. 选择题

- (1) switch 语句中“表达式”的计算结果是如下哪些类型? ()
- A. byte,short,char 和 int 类型
 - B. Byte,Short,Character 和 Integer 包装类
 - C. String 类型
 - D. 枚举类型

(2) 下列语句序列执行后, ch1 的值是()。

```
char ch1 = 'A', ch2 = 'W';  
if (ch1 + 2 < ch2) ++ch1;
```

A. 'A'

B. 'B'

C. 'C'

D. B

2. 判断题

(1) switch 语句中每一个 case 语句,后面必须加上 break 语句。()

(2) if 语句可以替代 switch 语句。()

(3) if 语句中的语句组只有一条语句时,不能省略大括号。()