第 3 章

学生成绩分级

学习目标

- 理解分支结构的作用；
- 学会应用分支语句设计程序；
- 掌握多分支程序设计的方法。

3.1 示 例 程 序

第 2 章完成了一个有点实际意义的程序，显示了一个学生的姓名、年龄和成绩。下面继续来丰富这个程序，根据学生成绩不同，显示出学生是否通过考试。判断的条件为：

通过：成绩>=60
未通过：成绩<60

想显示学生考试结果，就需要根据学生的成绩进行判断，看成绩是否大于等于 60 分，是则通过，否则不通过。在程序 2.3 的基础上修改代码，结果如程序 3.1 所示。

【程序 3.1】 程序 StudentInfo.java。

```
/**
 * 程序功能:显示一个学生信息
 * 作者:XXX
 * 日期:XX 年 XX 月 XX 日
 */
import java.util. Scanner;
public class StudentInfo{
    /**
     * main 方法是 Java 主方法
     * 功能:显示一个学生信息
     * 参数:没有用到
     * 返回值:无
     */
    public static void main(String [] args){
        //name 学生姓名
        String name = "张三";
        //age 学生年龄
```

```
        int age = 23;
        //grade 学生成绩
        double grade;
        //是否通过考试
        String result = "通过";
        //输入学生成绩
        System.out.println("输入学生成绩:");
        Scanner sc = new Scanner(System.in);
        grade = sc.nextDouble();
        if(grade < 60 ){
            result = "不通过";
        }
        System.out.println("姓名:" + name);
        System.out.println("年龄:" + age);
        System.out.println("成绩:" + grade);
        System.out.println("考试结果:" + result);
    }
}
```

程序 3.1 的编译和运行结果如图 3.1 所示。

```
D:\program\unit3\3-1\1-1>javac StudentInfo.java

D:\program\unit3\3-1\1-1>java StudentInfo
输入学生成绩:
87.67
姓名：张三
年龄：23
成绩：87.67
考试结果：通过
```

图 3.1　程序 3.1 运行结果

输入学生成绩值为 87.67，保存到变量 grade 中，if 语句中的条件 grade ＜ 60 不成立，result 变量值不变，因此输出考试结果为“通过”。

3.2　相 关 知 识

3.2.1　基本语句

Java 语言的基本语句包括方法调用语句、表达式语句和复合语句。方法调用语句是调用某个类或者对象的方法。例如输入学生成绩时调用 Scanner 类对象 sc 的 nextDouble()方法的语句：

```
grade = sc.nextDouble();
```

表达式语句，是在表达式的末尾加上分号“;”，例如，上面例子中给一个字符串变量赋

值语句：

```
result = "不通过";
```

Java 语言允许使用一对大括号“{”和“}”把一条或者多条语句扩起来，组成一个代码块，称为复合语句。逻辑上一个代码块相当于一条语句，如在下面程序段中，条件成立时执行的语句块：

```
if(grade < 60 ){
    result = "不通过";
    System.out.println("成绩" + grade);
}
```

上面介绍的三种语句是 Java 的基本语句，这些基本语句可以组成不同的程序段，也可以作为其他复合语句的基本组成部分。

3.2.2　条件分支语句

语句 if 是 Java 语言中的条件分支语句，作用是根据不同条件来执行不同的操作，if 语句格式如下：

```
if(条件表达式){
    若干条语句
}
[ else{
    若干条语句
} ]
```

其中，条件表达式是一个逻辑表达式，结果是一个逻辑值，如果这个值为 true，则执行条件后面紧跟着的若干条语句。方括号[]里的 else 部分是可选的，如果有 else 部分，则在条件表达式为 false 的时候执行 else 后面的若干条语句。如果没有 else 部分，条件表达式为 false 时执行 if 语句的下一条语句。例如下面程序段就是一个没有 else 部分的 if 语句。

```
if(grade < 60 ){
    result = "不通过";
}
```

上面这个程序段是程序 3.1 中的一段，可以修改这段程序，增加 else 部分，修改后的程序码如下：

```
String result = "";
…
if(grade < 60 ){
```

```
        result = "不通过";
    }
    else{
        result = "通过";
    }
```

修改后的程序，与原来的程序段功能完全一样，读者可以想想为什么是一样的，可以自己多找几个成绩来试试，看看两段程序的运行结果是否一致。

3.2.3 多分支语句

如果有多个条件需要判断，则需要使用多分支条件语句。多分支 if 语句格式如下：

```
if(条件表达式 1){
    若干条语句 1
}
else if(条件表达式 2){
    若干条语句 2
}
…
else if(条件表达式 n){
    若干条语句 n
}
[else{
    若干条语句
}]
```

与条件分支语句相同，所有的多分支语句中条件表达式都是一个逻辑表达式，结果为 true 或者 false。如果条件表达式 1 的结果为 true，执行后面的若干条语句 1，执行完结束；否则判断条件表达式 2，如果条件表达式 2 的结果为 true，则执行后面的若干条语句 2，执行完结束；否则继续判断后面的条件表达式。如果所有条件表达式全部为 false，则执行最后 else 部分的语句。

另外，还有一种多分支 switch 语句使用相对较少，除非应用中特别适合使用该语句，一般情况下尽量使用多分支 if 语句来实现。如果使用对象的状态常量作为条件，也可以考虑使用多态技术来实现，对象多态技术参见第 14 章。

3.3 训 练 程 序

参照程序 3.1 增加新功能。根据学生的成绩进行分级，级别包括优秀、良好、及格和不及格四个等级，等级与成绩对应关系如下：

优秀：成绩>= 90
良好：成绩>=75 且成绩< 90

及格:成绩>=60 且成绩< 75
不及格:成绩<60

要求从键盘读入学生成绩,显示学生成绩等级。

3.3.1　程序分析

由于学生成绩分成多个级别,因此适合使用多分支语句来实现。上面列出的判断方法就是不同的条件,等级就是执行结果。程序思路如下:

先判断学生成绩是否大于等于 90,如果是则等级为优秀
否则再判断成绩是否大于等于 75,如果是则等级为良好
否则再判断成绩是否大于等于 60,如果是则等级为及格
否则学生成绩等级为不及格

3.3.2　参考程序

根据上面的分析过程,参考程序 3.1 把这个分析过程转化成多分支语句,得到程序结果如程序 3.2 所示。

【程序 3.2】 多分支程序 StudentInfo.java。

```
/**
 * 程序功能:显示一个学生信息
 * 作者:XXX
 * 日期:XX 年 XX 月 XX 日
 */
import java.util.Scanner;
public class StudentInfo{
    /**
     * main 方法是 Java 主方法
     * 功能:显示一个学生信息
     * 参数:没有用到
     * 返回值:无
     */
    public static void main(String [] args){
        //name 学生姓名
        String name = "张三";
        //age 学生年龄
        int age = 23;
        //grade 学生成绩
        double grade = 0;
        //成绩级别
        String result = "";
        System.out.println("请输入学生成绩:");
        Scanner sc = new Scanner(System.in);
        grade = sc.nextDouble();
```

```
            if(grade >= 90 ){
                result = "优秀";
            }
            else if(grade<90 && grade >=75){
                result="良好";
            }
            else if(grade<75 && grade>=60){
                result="及格";
            }
            else{
                result="不及格";
            }
            System.out.println("姓名:" + name);
            System.out.println("年龄:" + age);
            System.out.println("成绩:" + grade);
            System.out.println("成绩级别:" + result);
        }
    }
```

程序 3.2 的运行结果如图 3.2 所示。

```
D:\program\unit3\3-3\3-1>javac StudentInfo.java

D:\program\unit3\3-3\3-1>java StudentInfo
请输入学生成绩：
67.5
姓名：张三
年龄：23
成绩：67.5
成绩级别：及格
```

图 3.2　程序 3.2 运行结果

输入学生成绩 67.5，根据多分支语句中的判断条件，条件“grade＜75 && grade＞＝60”成立，执行该条件对应的语句，将 result 变量赋值为“及格”，多分支语句结束。程序最后输出成绩级别为“及格”。有兴趣的读者可以继续改进这个程序，增加输入成绩在 0～100 范围内的判断。

3.3.3　进阶训练

在上面的程序中，学生信息和教师信息都是直接将键盘输入的数据赋给变量，没有经过处理。但程序设计时，许多变量的值输入后并不直接使用，而要经过有效性判断。如果是合法数据进行处理，不合法数据给出提示。

【程序 3.3】 计算圆面积进阶 2。在程序 2.8 的基础上，从键盘输入圆的半径值，如果圆的半径大于等于 0，则计算出圆的面积并输出；否则打印提示：“圆半径错误!”。

程序设计思路分析：

(1) 定义 double 类型的变量 r 和 area，分别表示圆的半径和面积；

(2) 从键盘输入半径 r 的值；

(3) 使用分支语句，判断 r>=0 是否为真；

(4) 为真则使用圆面积公式(r×r×π)计算出面积的值，并赋值给变量 area；将 area 的值输出；

(5) 否则输出字符串“圆半径错误!”。

读者可以根据前面给出的设计思路，自己编写这个程序，并编译、运行程序，查看结果是否正确。

3.4　拓 展 知 识

3.4.1　分支语句讨论

从上面的例子可以看出分支语句的格式比较简单，但是在实际应用中还是有很多需要注意的问题，恰当处理这些问题可以提高代码的可读性和质量。下面讲解三个常见问题的处理方法。

第一个问题，如果分支中只有一条语句是否可以看作一个语句块。分支语句根据条件不同执行不同的分支，每个分支可以是一条语句也可以是一个语句块，前面的例子中每个分支都看作一个语句块，使用大括号括了起来，例如下面代码段：

```
if(grade < 60 ){
    result = "不通过";
}
else{
    result = "通过";
}
```

在这个代码段中，每个分支只有一条语句，但是仍然用大括号括起来。从语法上讲，上面代码段的两个分支可以不用大括号，写成如下形式：

```
if(grade < 60 ) result = "不通过";
else result = "通过";
```

从程序设计规范角度来讲，建议把每个分支作为一个语句块处理，这样增加程序可读性，降低出错可能性。

第二个问题是多个 if 语句相互配合的问题。在实际的程序设计中，分支语句的每个分支可能也是 if 分支语句，例如下面程序段：

```
double grade = 70;
if(grade >= 60)
    if(grade >=90)
        System.out.println("优秀");
```

```
        else
            System.out.println("什么等级？");
```

分析上面的程序段，里面的 else 语句和哪个 if 相匹配呢？刚开始看到这样的程序是很难一下子说清楚的。按照语法上讲，else 应该和离它最近的且没有配对的 if 相匹配。实际程序设计中尽量避免出现这样容易产生歧义的语句。处理方法很简单，一种方法是按照前面说的每个分支都作为语句块处理，加上大括号；还有一种方法就是使用多分支 if 语句实现。感兴趣的读者可以自己尝试使用这两种方法完成这个程序。

第三个常见的问题是，对于多分支 if 语句，应该按照什么次序来排列这些分支。常见的处理方法有两种：第一种方法就是程序 3.2 给出的方法，按照问题给出的先后顺序来排列。程序 3.2 中就是按照等级优秀、良好、及格和不及格的次序进行处理，这样做方便用户阅读和理解程序。对于问题中没有明显次序的多分支程序，建议可以按照出现的频率把经常使用的分支排在前面，这样可以提高程序处理效率。

3.4.2 数据合法性检查

在编写 Java 程序过程中，经常使用输入语句从键盘输入数据，例如程序 3.2 中就使用输入语句输入学生成绩。程序 3.2 能够正确运行并得到结果，但这个程序是否还存在问题？

如果输入的数据不是双精度数，而是一个字符串，结果会如何？下面运行程序 3.2，输入数据 1a2，运行结果如图 3.3 所示。

```
D:\program\unit3\3-3\3-1>java StudentInfo
请输入学生成绩：
1a2
Exception in thread "main" java.util.InputMismatchException
        at java.util.Scanner.throwFor(Scanner.java:840)
        at java.util.Scanner.next(Scanner.java:1461)
        at java.util.Scanner.nextDouble(Scanner.java:2387)
        at StudentInfo.main(StudentInfo.java:25)
```

图 3.3 程序 3.2 运行结果

可以看到程序出现了错误，出错的原因是因为输入的数据不正确，系统提示抛出一个 InputMismatchException 类异常，有关异常知识在第 17 章中进行介绍。

如果编写的程序是交付给用户使用的实际软件，那么无法保证用户每次的输入数据都符合程序的需要。这时候程序就需要先对用户输入的数据进行合法性检查，只有通过检查，格式正确的数据程序才能进行下一步处理，否则提示用户输入数据格式不正确。改进后的程序如程序 3.4 所示。

【程序 3.4】 合法性检查程序 StudentInfo.java。

```
/**
 * 程序功能：显示一个学生信息
 * 作者：xxx
 * 日期：xx 年 xx 月 xx 日
 */
```

```
import java.util.Scanner;
import java.util.InputMismatchException;
public class StudentInfo{
    /**
     * main 方法是 Java 主方法
     * 功能:显示一个学生信息
     * 参数:没有用到
     * 返回值:无
     */
    public static void main(String [] args){
        //name 学生姓名
        String name = "张三";
        //age 学生年龄
        int age = 23;
        //grade 学生成绩
        double grade = 0;
        //是否通过考试
        String result = "通过";
        try{
            Scanner sc = new Scanner(System.in);
            grade = sc.nextDouble();
        }
        catch(InputMismatchException ime){
            System.out.println("输入数据格式不正确");
        }
        if(grade < 60 ){
            result = "不通过";
        }
        System.out.println("姓名:" + name);
        System.out.println("年龄:" + age);
        System.out.println("成绩:" + grade);
        System.out.println("考试结果:" + result);
    }
}
```

程序 3.4 的运行结果如图 3.4 所示。

```
D:\program\unit3\3-4\4-1>javac StudentInfo.java

D:\program\unit3\3-4\4-1>java StudentInfo
1a2
输入数据格式不正确
姓名：张三
年龄：23
成绩：0.0
考试结果：不通过
```

图 3.4　程序 3.4 运行结果

程序中增加了异常处理，用来处理输入数据格式不正确这种异常的情况。有关异常处理详见第17章。

最后做个简单的总结，本章学习了分支结构关键字：if、else，同时涉及到switch、case，但没有详细说明，另外，goto语句已经废弃不再进行讲解，表3-1中分别用粗体和斜体表示上述关键字。其余关键字将在后续章节讲解。

表3-1 Java关键字

abstract	do	implements	protected	this
assert***	**else**	import	public	throw
break	enum****	instanceof	return	throws
case	extends	interface	static	transient
catch	final	native	strictfp**	try
class	finally	new	super	void
const*	for	package	*switch*	volatile
continue	*goto**	private	synchronized	while
default	**if**			

3.5 实做程序

1. 请根据错误提示找出下列程序中存在错误并分析原因。

（1）程序Test.java，根据成绩判断考试是否通过，考试不通过则输出成绩，通过则不输出，运行结果如下。

```
public class Test {
    public static void main(String[] args){
        double grade = 70;
        if(grade >= 60)
            System.out.println("考试通过");
        else
            System.out.println("考试成绩:" + grade);
            System.out.println("考试不通过");
    }
}
```

```
D:\program\unit3\3-5\5-1\1>javac Test.java

D:\program\unit3\3-5\5-1\1>java Test
考试通过
考试不通过
```

（2）程序Test.java，程序编译报错。

```
public class Test {
```

```
    public static void main(String[] args){
        int flag;
        flag = 0;
        if(flag){
            System.out.println("标志不为 0");
        }
    }
}
```

```
D:\program\unit3\3-5\5-1\2>javac Test.java
Test.java:5: 不兼容的类型
找到:  int
需要:  boolean
                if(flag){
                   ^
1 错误
```

2. 从键盘输入两个数，按照由大到小顺序输出。要点提示：

(1) 方法一：比较 a 和 b 的大小，如果 a＞b 则先输出 a 再输出 b，否则先输出 b 再输出 a；

(2) 方法二：比较 a 和 b 的大小，如果 a＜b 则将 a 和 b 的值交换，最后依次输出 a 和 b。

3. 从键盘输入三个数，输出最大数。要点提示：

(1) 定义一个变量 max 保存最大值；

(2) 假定第一个最大，把它放在 max 中；

(3) 比较第二个数是否比第一个数大，如果是就把它放在 max 中；

(4) 同样方法比较第三个数，最后 max 中的值即为最大值。

4. 设计程序显示桌子的信息，包括桌子的类型(长方形、正方形、圆形)、腿数、高度和面积，其中面积是通过根据桌子类型不同而输入不同的数据来计算得出：长方形：输入长和宽；正方形：输入边长；圆形：输入半径。要点提示：

(1) 桌子的形状可以使用一个整数变量来表示，例如用整数 1～3 分别代表长方形、方形、圆形；

(2) 先输入桌子的类型，再根据类型输入不同的数据，最后计算面积。

5. 下面程序编译和运行结果都是正确的，分析程序还有哪些不足，如何改进？

```
public class Test {
    public static void main(String[] args){
        double grade = 50;
        if(grade >= 60){
        }
        else{
            System.out.println("考试不通过");
        }
    }
}
```

要点提示：

（1）分支语句中需要处理的分支按照重要性从前到后排列；

（2）不提倡使用空分支。

6. 输入整数 n，判断 n 是否可以同时被 3 和 7 整除，输出判断的结果。要点提示，判断 n 是否整除 3，可以使用表达式：n % 3 == 0。

7. 输入三个整数 a、b、c，将三个数按从小到大的顺序输出。要点提示：

（1）方法一：直接比较 a、b、c 三个数，根据比较的结果输出；

（2）方法二：先比较 a 和 b，如果 a>b 则交换 a、b，再与 c 比较大小。

8. 已知 x 和 y 的关系是一个分段函数，编写程序，输入 x 的值，计算输出 y 的值。

$$y=\begin{cases}10+2x, & x<0\\ 1, & x=0\\ 5+x, & x>0\end{cases}$$

9. PM2.5 是指大气中直径小于或等于 2.5 微米的颗粒物，也称为可入肺颗粒物，它含有大量的有毒、有害物质且在大气中的停留时间长、输送距离远，因而对人体健康和大气环境质量的影响很大。我们用空气中 PM2.5 的浓度反映空气污染的程度。编写空气质量预报程序，输入 PM2.5 的值，显示空气质量的分级结果。空气质量的等级标准如下表。

PM2.5 浓度（μg/m³）	等　　级
0～35	优
35～75	良
75～115	轻度污染
115～150	中度污染
150～250	重度污染
250 以上	严重污染

10. 编写程序，输入自己的身高和体重，根据公式计算身体质量指数 BMI 值，并输出所对应的分类。BMI 的计算公式：BMI＝体重（kg）÷身高²（m²）。

国内 BMI 值（kg/m²）	分　　类
<18.5	偏瘦
18.5～24	正常
24～28	偏胖
>=28	肥胖

11. 购物时如果买的商品数量比较多，商家往往会打相应的折扣。购买瓶装水时，如果购买数量为 1～5 瓶，按原价结账；如果购买数量为 6～10 瓶，结账时打 9.5 折；如果购买 11～23 瓶，结账时打 9 折；如果购买 24 瓶及以上，结账时打 8.5 折。编写程序，输入购买数量和单价，计算并输出付款金额，计算结果保留小数点后两位。

12. 实做程序 2.6 和 2.7 实现了摄氏温度和华氏温度的相互转换。现在我们把这两个程序合并成一个完整的温度转换程序，如果输入摄氏温度则转换为华氏温度，如果输入华氏温度则转换为摄氏温度，输出转换结果。要点提示：

（1）设置一个标志位用来判断进行哪种转换，例如值为 1 和 2，从键盘输入标志位的值；

（2）使用分支结构，将标志位的取值作为条件表达式，根据标志位的值不同执行不同的分支；

（3）两个分支中分别编写摄氏温度转换华氏温度、华氏温度转换摄氏温度的代码。