

选 择

学习目标

- 声明boolean类型以及使用比较运算符书写布尔表达式 (3.2节)。
- 使用布尔表达式编写程序AdditionQuiz (3.3节)。
- 使用单向if语句实现选择控制 (3.4节)。
- 使用单向if语句编写游戏GuessBirthday的程序 (3.5节)。
- 使用双向if语句实现选择控制 (3.6节)。
- 使用嵌套的if语句实现选择控制 (3.7节)。
- 避免if语句中的常见错误 (3.8节)。
- 使用选择语句编程的不同种类的例子 (SubtractionQuiz、BMI、ComputeTax) (3.9~3.11节)。
- 使用Math.random()方法产生随机数 (3.9节)。
- 使用逻辑运算符(&&、||和!)对条件进行组合 (3.12节)。
- 使用带组合条件的选择语句进行编程 (LeapYear、Lottery) (3.13~3.14节)。
- 使用switch语句实现选择控制 (3.15节)。
- 使用条件运算符书写表达式 (3.16节)。
- 使用System.out.printf方法格式化输出 (3.17节)。
- 检查控制运算符优先级和结合方向的规则 (3.18节)。
- (GUI) 使用确定对话框获取用户的确认信息 (3.19节)。

3.1 引言

如果给程序清单2-2中的radius赋一个负值,程序就会打印一个非法的结果。如果半径是一个负值,是不希望程序计算面积的,那么该如何处理这种情况呢?

Java和所有高级程序设计语言一样也提供选择语句,以便在两个或更多可选择的流程中做出选择。可以用下面的选择语句来替换程序清单2-2中的第12~17行:

```
if (radius < 0)
    System.out.println("Incorrect input");
else {
    area = radius * radius * 3.14159;
    System.out.println("Area is " + area);
}
```

选择语句要用到条件。条件就是布尔表达式。本章首先介绍布尔类型、布尔值、布尔比较运算符和布尔表达式。

3.2 boolean数据类型

该如何比较两个值呢?例如:一个半径是大于0,等于0,还是小于0呢?如表3-1所示,Java提供六种比较运算符 (comparison operator) (也称为关系运算符 (relational operator)),用于两个值的比较(假设表中的半径值为5)。

注意 也可以对字符进行比较。对字符进行比较和对字符统一码进行比较是一样的。例如:因为'a'的统一码大于'A'的统一码,所以'a'大于'A'。参见附录B就可以找到字符的顺序。

表3-1 比较运算符

运 算 符	名 称	举 例	结 果
<	小于	radius<0	false
<=	小于等于	radius<=0	false
>	大于	radius>0	true
>=	大于等于	radius>=0	true
==	等于	radius==0	false
!=	不等于	radius!=0	true

警告 相等的比较运算符是两个等号 (==)，而不是一个等号 (=)，后者是指赋值运算符。

比较的结果是一个布尔值：true（真）或false（假）。例如，下面的语句显示true：

```
double radius = 1;
System.out.println(radius > 0);
```

具有布尔值的变量称为布尔变量（boolean variable），boolean数据类型用于声明布尔型变量。boolean型变量只可能是以下这两个值中的一个：true和false。例如，下述语句将true赋值给变量lightsOn：

```
boolean lightsOn = true;
```

true和false都是直接量，就像10这样的数字。它们都是保留字，不能用做程序中的标识符。




3.3 问题：一个简单的数学学习工具

假设希望开发一个程序，让一年级学生练习加法。程序随机产生两个一位整数：number1和number2，然后显示给学生“What is 7 + 9?”，如示例运行所示。当学生在输入对话框中输入答案之后，程序显示一个消息，表明答案是真的还是假的。

产生随机数的方法有很多种。现在，使用System.currentTimeMillis()%10产生第一个整数，使用System.currentTimeMillis()*7%10产生第二个整数。程序清单3-1给出该程序。第5~6行产生两个数：number1和number2。第14行获取从用户那里得到的答案。第18行使用布尔表达式number1 + number2 == answer给答案打分。

程序清单3-1 AdditionQuiz.java

```
1 import java.util.Scanner;
2
3 public class AdditionQuiz {
4     public static void main(String[] args) {
5         int number1 = (int)(System.currentTimeMillis() % 10);
6         int number2 = (int)(System.currentTimeMillis() * 7 % 10);
7
8         // Create a Scanner
9         Scanner input = new Scanner(System.in);
10
11         System.out.print(
12             "What is " + number1 + " + " + number2 + "? ");
13
14         int answer = input.nextInt();
15
16         System.out.println(
17             number1 + " + " + number2 + " = " + answer + " is " +
18             (number1 + number2 == answer));
19     }
20 }
```

What is 1 + 7? 8 <input type="button" value="Enter"/> 1 + 7 = 8 is true					
What is 4 + 8? 9 <input type="button" value="Enter"/> 4 + 8 = 9 is false					
line#	number1	number2	answer	output	
5	4				
6		8			
14			9		
16				4 + 8 = 9 is false	

3.4 if语句

前面的程序会显示像“6 + 2 = 7 is false”这样的消息。如果希望显示的消息是“6 + 2 = 7 is incorrect”，那么必须使用条件语句实现这个细微的改变。

本节介绍选择语句。Java有几种类型的选择语句：单向if语句、双向if语句、嵌套if语句、switch语句和条件表达式。

单向if语句

单向if语句是指当且仅当条件为true时执行一个动作。单向if语句的语法如下：

```
if (布尔表达式) {
    语句 (组);
}
```

执行的流程图如图3-1a所示。

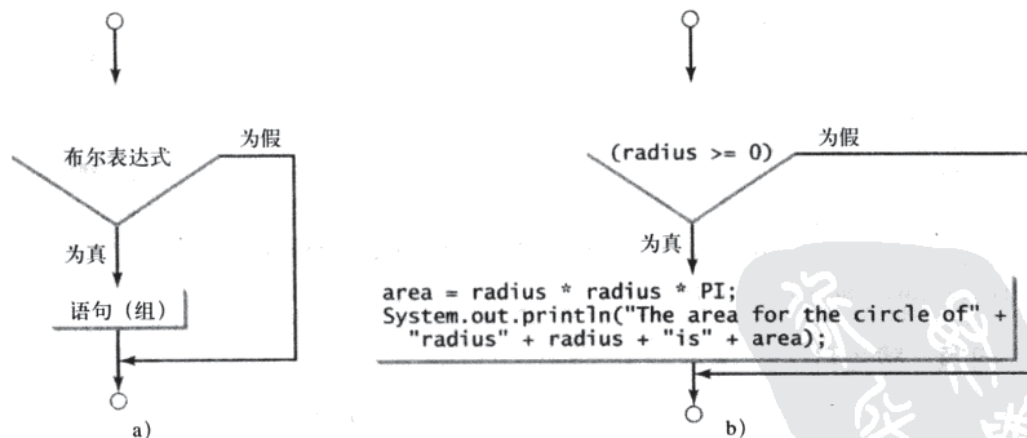


图3-1 if语句在布尔表达式计算结果为true的情况下执行语句组

如果布尔表达式计算的结果为true，则执行块内语句。作为例子，看看下面的代码：

```
if (radius >= 0) {
    area = radius * radius * PI;
    System.out.println("The area for the circle of radius " +
        radius + " is " + area);
}
```

上述语句的流程图参见图3-1b。如果半径radius的值大于等于0，则计算面积area并显示其结果；

否则，不执行块内的两条语句。

布尔表达式应该用括号括住。例如：下面图a中的代码是错误的。应该将它改为如图b所示。

```
if i > 0 {
    System.out.println("i is positive");
}
```

a) 错误的

```
if (i > 0) {
    System.out.println("i is positive");
}
```

b) 正确的

如果花括号内只有一条语句，则可以省略花括号。例如：下面两个语句是等价的。

```
if (i > 0) {
    System.out.println("i is positive");
}
```

a)

等价

```
if (i > 0)
    System.out.println("i is positive");
```

b)


程序清单3-2给出一个程序，提示用户输入一个整数。如果该数字是5的倍数，打印HiFive。如果该数字能被2整除，打印HiEven。

程序清单3-2 SimpleIfDemo.java

```
1 import java.util.Scanner;
2
3 public class SimpleIfDemo {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6         System.out.println("Enter an integer: ");
7         int number = input.nextInt();
8
9         if (number % 5 == 0)
10            System.out.println("HiFive");
11
12        if (number % 2 == 0)
13            System.out.println("HiEven");
14    }
15 }
```

Enter an integer: 4 
HiEven



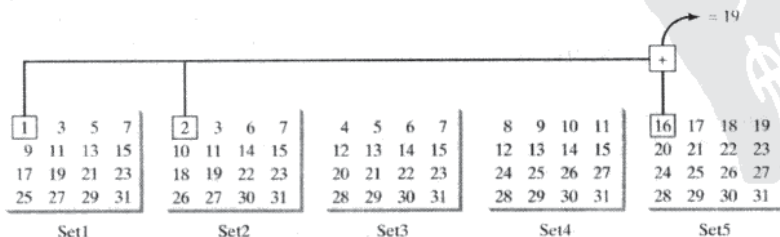
Enter an integer: 30 
HiFive
HiEven



程序提示用户输入一个整数（第7行），如果它能被5整除就显示HiFive（第9~10行），而如果它被2整除则显示HiEven（第12~13行）。

3.5 问题：猜生日

可以通过询问朋友5个问题，找到他出生在一个月的哪一天。每个问题都是在询问他的出生日是否是5个数字集合中的一个。



生日是出现这一天的每个集合的第一个数字的和。例如：如果生日是19，那么它会出现集合1、集合2和集合5中。这三个集合的第一个数字分别是1、2和16。它们的和就是19。

程序清单3-3给出程序，提示用户回答该天是否在集合1中（第41~47行），是否在集合2中（第50~56行），是否在集合3中（第59~65行），是否在集合4中（第68~74行），是否在集合5中（第77~83行）。如果这个数字在某个集合中，程序就将该集合的第一个数字加到day中去（第47、56、65、74、83行）。

程序清单3-3 GuessBirthday.java

```
1 import java.util.Scanner;
2
3 public class GuessBirthday {
4     public static void main(String[] args) {
5         String set1 =
6             " 1 3 5 7\n" +
7             " 9 11 13 15\n" +
8             "17 19 21 23\n" +
9             "25 27 29 31";
10
11         String set2 =
12             " 2 3 6 7\n" +
13             "10 11 14 15\n" +
14             "18 19 22 23\n" +
15             "26 27 30 31";
16
17         String set3 =
18             " 4 5 6 7\n" +
19             "12 13 14 15\n" +
20             "20 21 22 23\n" +
21             "28 29 30 31";
22
23         String set4 =
24             " 8 9 10 11\n" +
25             "12 13 14 15\n" +
26             "24 25 26 27\n" +
27             "28 29 30 31";
28
29         String set5 =
30             "16 17 18 19\n" +
31             "20 21 22 23\n" +
32             "24 25 26 27\n" +
33             "28 29 30 31";
34
35         int day = 0;
36
37         // Create a Scanner
38         Scanner input = new Scanner(System.in);
39
40         // Prompt the user to answer questions
41         System.out.print("Is your birthday in Set1?\n");
42         System.out.print(set1);
43         System.out.print("\nEnter 0 for No and 1 for Yes: ");
44         int answer = input.nextInt();
45
46         if (answer == 1)
47             day += 1;
48
49         // Prompt the user to answer questions
50         System.out.print("\nIs your birthday in Set2?\n");
51         System.out.print(set2);
52         System.out.print("\nEnter 0 for No and 1 for Yes: ");
53         answer = input.nextInt();
54
55         if (answer == 1)
56             day += 2;
```

60 • 第3章 选 择

```

57
58 // Prompt the user to answer questions
59 System.out.print("Is your birthday in Set3?\n");
60 System.out.print(set3);
61 System.out.print("\nEnter 0 for No and 1 for Yes: ");
62 answer = input.nextInt();
63
64 if (answer == 1)
65     day += 4;
66
67 // Prompt the user to answer questions
68 System.out.print("\nIs your birthday in Set4?\n");
69 System.out.print(set4);
70 System.out.print("\nEnter 0 for No and 1 for Yes: ");
71 answer = input.nextInt();
72
73 if (answer == 1)
74     day += 8;
75
76 // Prompt the user to answer questions
77 System.out.print("\nIs your birthday in Set5?\n");
78 System.out.print(set5);
79 System.out.print("\nEnter 0 for No and 1 for Yes: ");
80 answer = input.nextInt();
81
82 if (answer == 1)
83     day += 16;
84
85 System.out.println("\nYour birthday is " + day + "!");
86 }
87 }

```

Is your birthday in Set1?

1 3 5 7

9 11 13 15

17 19 21 23

25 27 29 31

Enter 0 for No and 1 for Yes: 1

Is your birthday in Set2?

2 3 6 7

10 11 14 15

18 19 22 23

26 27 30 31

Enter 0 for No and 1 for Yes: 1

Is your birthday in Set3?

4 5 6 7

12 13 14 15

20 21 22 23

28 29 30 31

Enter 0 for No and 1 for Yes: 0

Is your birthday in Set4?

8 9 10 11

12 13 14 15

24 25 26 27

28 29 30 31

Enter 0 for No and 1 for Yes: 0

Is your birthday in Set5?

16 17 18 19

20 21 22 23

24 25 26 27

28 29 30 31

Enter 0 for No and 1 for Yes: 1

Your birthday is 19

line#	day	answer	output
35	0		
44		1	
47	1		
53		1	
56	3		
62		0	
71		0	
80		1	
83	19		
Your birthday is 19			



这个游戏是很容易编程的。你可能很好奇想知道如何创建这个游戏。实际上，这个游戏背后的数学知识是非常简单的。这些数字不是随意组成一组的。它们放在五个集合中的方式是经过深思熟虑的。这五个集合的第一个数分别是1、2、4、8和16，它们分别对应二进制数的1、10、100、1000和10000。从1到31的十进制数最多用五个二进制数就可以表示，如图3-2a所示。假设它是 $b_5b_4b_3b_2b_1$ ，那么 $b_5b_4b_3b_2b_1 = b_5 \cdot 10000 + b_4 \cdot 1000 + b_3 \cdot 100 + b_2 \cdot 10 + b_1$ ，如图3-2b所示。如果某天的二进制数在 b_i 位为整数1，那么该数就该出现在Set*k*中。例如：数字19的二进制是10011，所以它就该出现在集合1、集合2和集合5中。它就是二进制数 $1+10+10000=10011$ 或者十进制数 $1+2+16=19$ 。数字31的二进制是11111，所以它就会出现在集合1、集合2、集合3、集合4和集合5中。它就是二进制数 $1+10+100+1000+10000=11111$ ，或是十进制数 $1+2+4+8+16=31$ 。

Decimal	Binary			
1	00001	b_5 0 0 0 0	10000	
2	00010	b_4 0 0 0	1000	
3	00011	b_3 0 0	100	
...		b_2 0	10	
19	10011	b_1	1	
...			10011	11111
31	11111	$b_5 b_4 b_3 b_2 b_1$	19	31

a)

b)

图3-2 a) 从1到31的数字可以用5位二进制数表示；

b) 通过添加二进制数1、10、100、1000或者10000得到5位二进制数

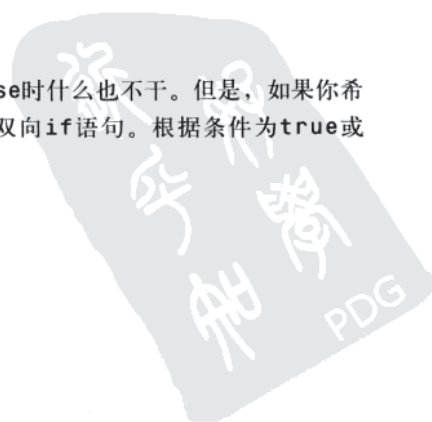
3.6 双向if语句

当指定条件为true时单向if语句执行一个操作。而当条件为false时什么也不干。但是，如果你希望在条件为false时也能执行一些动作，该怎么办呢？你可以使用双向if语句。根据条件为true或false，双向if语句可以指定不同的操作。

下面是双向if语句的语法：

```
if (布尔表达式) {
    布尔表达式为真时执行的语句(组);
}
else{
    布尔表达式为假时执行的语句(组);
}
```

语句的流程图如图3-3所示。



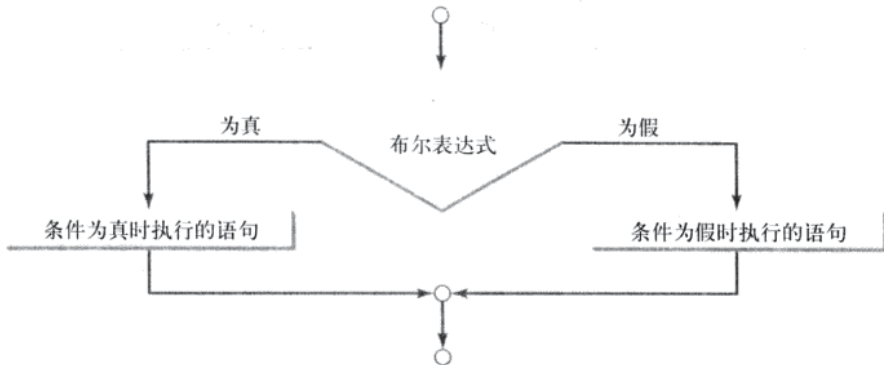


图3-3 若布尔表达式计算结果为true, if...else语句运行true情况下的语句组;
否则, 运行false情况下的语句组

如果布尔表达式的计算结果为true, 则执行条件为true时该执行的语句; 否则, 执行条件为false时该执行的语句。例如, 考虑下面的代码:

```

if (radius >= 0) {
    area = radius * radius * PI;
    System.out.println("The area for the circle of radius " +
        radius + " is " + area);
}
else {
    System.out.println("Negative input");
}
  
```

若radius>=0为true, 则计算并显示area; 如果radius>=0为false, 则打印信息"Negative input"。

通常, 如果花括号中只有一条语句, 那就可以省略花括号。因此, 前例中用于括住语句System.out.println("Negative input")的花括号可以省略。

这里还有另外一个使用if...else语句的例子。这个例子检测一个数是奇数还是偶数, 如下所示:

```

if (number % 2 == 0)
    System.out.println(number + " is even.");
else
    System.out.println(number + " is odd.");
  
```

3.7 嵌套的if语句

if或if...else语句中的语句可以是任意合法的Java语句, 甚至可以是其他的if或if...else语句。外层if语句里的内层if语句称为是嵌套的(nested)。内层if语句还可以包含其他的if语句; 事实上, 对嵌套的深度没有限制。例如, 下面就是一个嵌套的if语句:

```

if (i > k) {
    if (j > k)
        System.out.println("i and j are greater than k");
}
else
    System.out.println("i is less than or equal to k");
  
```

语句if(j>k)被嵌套在语句if(i>k)内。

嵌套的if语句可用于实现多重选择。例如: 图3-4a中所给出的语句使用了多重选择, 根据分数给变量grade赋一个用字母表示的级别。

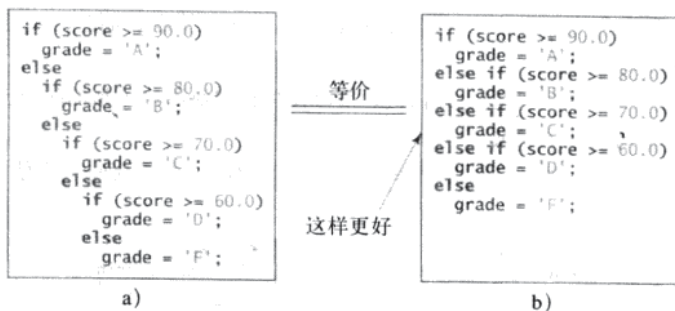
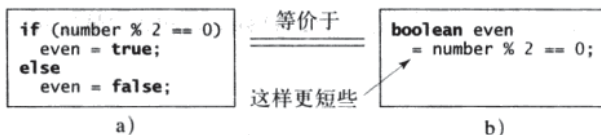


图3-4 推荐使用如图3-4b所示的多选if语句格式

这个if语句的执行过程如下。测试第一个条件 ($\text{score} \geq 90.0$)。如果它为true, 级别就变成 'A'。如果它为false, 就测试第二个条件 ($\text{score} \geq 80.0$)。如果第二个条件为true, 级别就变成 'B'。如果第二个条件为false, 则会继续测试第三个和剩余的条件 (如果有必要的话), 直到遇到满足的条件, 或者所有条件都为false。如果所有条件都为false, 级别就变成 'F'。注意: 只有在前面的所有条件都为false时才测试下一个条件。

图3-4a中的if语句等价于图3-4b中的if语句。事实上, 图3-4b是推荐使用的多重选择if语句的书写风格。这种风格可以避免深度缩进, 并使程序容易阅读。

提示 编程新手通常会用下面图a中的代码, 将一个测试条件赋值给一个布尔变量。



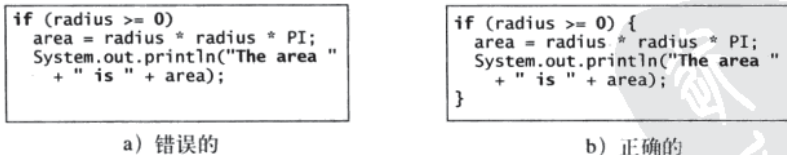
可以通过将这个测试值直接赋值给该变量来简化代码, 如图b中所示。

3.8 选择语句中的常见错误

以下错误是编程新手经常会犯的错误。

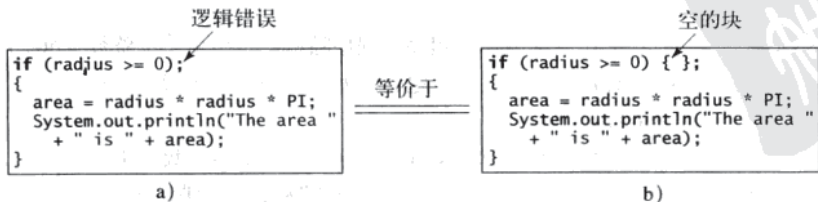
常见错误1: 忘记必要的括号

如果块中只有一条语句, 就可以忽略花括号。但是, 当需要用花括号将多条语句括在一起时, 忘记花括号是一个常见的程序设计错误。如果通过在没有花括号的if语句中添加一条新语句来修改代码, 就必须插入花括号。例如: 下面图a中的代码是错误的。应该用花括号将多个语句放在一起, 如图b所示。



常见错误2: 在if行出现错误的分号

如下面的图a中所示, 在if行加上了一个分号, 这是一个常见错误。

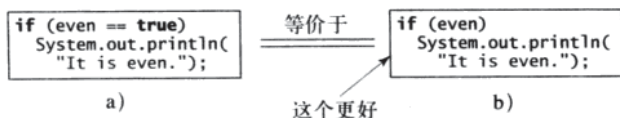


这个错误是很难发现的，因为它既不是编译错误也不是运行错误，而是一个逻辑错误。图a中的代码等价于一个带空块的图b中的代码。

当使用下行块风格时，经常会出现这个错误。所以使用行尾块风格可帮助防止出现此类错误。

常见错误3：对布尔值的冗余测试

为了检测测试条件中的布尔型变量是true还是false，像图a中的代码这样使用相等比较运算符是多余的：



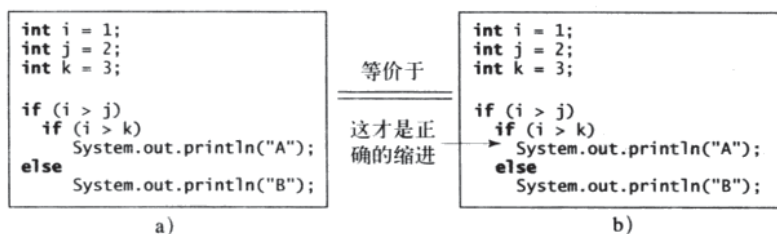
比较好的替代方法就是直接测试布尔变量，如图b所示。这么做的另一个原因就是避免出现难以发现的错误。使用=运算符而不是==运算符去比较测试条件中的两项是否相等是一个常见错误。它可能会导致出现下面的错误语句：

```
if (even = true)
    System.out.println("It is even.");
```

这条语句没有语法错误。它给even赋值true，这样even永远都是true。

常见错误4：悬空else出现的歧义

下面图a中的代码有两个if子句和一个else子句。那么，哪个if子句和这个else匹配呢？这里的缩进表明else子句匹配第一个if子句。但是，else实际匹配的是第二个if子句。这种现象就称为悬空else歧义（dangling-else ambiguity）。在同一个块中，else总是和离它最近的if子句匹配。这样，图a中的语句就等价于图b中的语句。



由于 $(i > j)$ 为假，所以图a和图b中的语句不打印任何东西。为强制这个else匹配第一个if子句，必须添加一对花括号。

```
int i = 1, j = 2, k = 3;

if (i > j) {
    if (i > k)
        System.out.println("A");
}
else
    System.out.println("B");
```

这条语句会打印出B。

3.9 问题：一个改进的数学学习工具

假设你想开发一个一年级学生练习减法的程序。程序随机产生两个一位整数：number1和number2，且满足 $\text{number1} \geq \text{number2}$ 。程序向学生显示问题，例如，“What is 9-2?”。当学生输入答案之后，程序会显示一个消息表明该答案是否正确。

前面的程序使用`Systems.currentTimeMillis()`产生两个随机数。更好的方法是使用`Math`类中的`random()`方法。调用这个方法会返回一个双精度的随机值d且满足 $0.0 \leq d < 1.0$ 。这样，`(int)`

(`Math.random()*10`) 会返回一个随机的一位整数 (即0到9之间的数)。

程序可能如下工作:

- 1) 产生两个一位整数`number1`和`number2`。
- 2) 如果`number1 < number2`, 交换`number1`和`number2`。
- 3) 提示学生回答 “what is number1-number2?”。
- 4) 检查学生的答案并且显示该答案是否正确。


完整的程序如程序清单3-4所示。

程序清单3-4 **SubtractionQuiz.java**


```

1 import java.util.Scanner;
2
3 public class SubtractionQuiz {
4     public static void main(String[] args) {
5         // 1. Generate two random single-digit integers
6         int number1 = (int)(Math.random() * 10);
7         int number2 = (int)(Math.random() * 10);
8
9         // 2. If number1 < number2, swap number1 with number2
10        if (number1 < number2) {
11            int temp = number1;
12            number1 = number2;
13            number2 = temp;
14        }
15
16        // 3. Prompt the student to answer "What is number1 - number2?"
17        System.out.print
18            ("What is " + number1 + " - " + number2 + "? ");
19        Scanner input = new Scanner(System.in);
20        int answer = input.nextInt();
21
22        // 4. Grade the answer and display the result
23        if (number1 - number2 == answer)
24            System.out.println("You are correct!");
25        else
26            System.out.println("Your answer is wrong\n" + number1 + " - "
27                               + number2 + " should be " + (number1 - number2));
28    }
29 }

```

What is 6 - 6? 0 
You are correct!



What is 9 - 2? 5 
Your answer is wrong
9 - 2 should be 7



line#	number1	number2	temp	answer	output
6	2				
7		9			
11			2		
12	9				
13		2			
20				5	
26					Your answer is wrong 9 - 2 should be 7



PDG

为了交换变量number1和number2,首先要使用一个临时变量temp (第11行) 存储number1的值。将number2的值赋值给number1 (第12行),然后将temp的值赋给number2 (第13行)。

3.10 问题: 计算身体质量指数

身体质量指数 (BMI) 是关于体重指标的健康测量。将以千克为单位的体重除以以米为单位身高的平方,就得到BMI的值。针对16岁及以上年龄的人群,他们的BMI值的说明如右表所示:

编写程序,提示用户输入以英镑为单位的体重,以及以英尺为单位的身高,然后显示BMI。注意: 一磅是0.45359237 千克,而一英尺是0.0254米。程序清单3-5给出这个程序。



程序清单3-5 ComputeBMI.java

```

1 import java.util.Scanner;
2
3 public class ComputeAndInterpretBMI {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6
7         // Prompt the user to enter weight in pounds
8         System.out.print("Enter weight in pounds: ");
9         double weight = input.nextDouble();
10
11        // Prompt the user to enter height in inches
12        System.out.print("Enter height in inches: ");
13        double height = input.nextDouble();
14
15        final double KILOGRAMS_PER_POUND = 0.45359237; // Constant
16        final double METERS_PER_INCH = 0.0254; // Constant
17
18        // Compute BMI
19        double weightInKilograms = weight * KILOGRAMS_PER_POUND;
20        double heightInMeters = height * METERS_PER_INCH;
21        double bmi = weightInKilograms /
22            (heightInMeters * heightInMeters);
23
24        // Display result
25        System.out.println("Your BMI is " + bmi);
26        if (bmi < 16)
27            System.out.println("You are seriously underweight");
28        else if (bmi < 18)
29            System.out.println("You are underweight");
30        else if (bmi < 24)
31            System.out.println("You are normal weight");
32        else if (bmi < 29)
33            System.out.println("You are overweight");
34        else if (bmi < 35)
35            System.out.println("You are seriously overweight");
36        else
37            System.out.println("You are gravely overweight");
38    }
39 }

```

BMI	说 明
16以下	严重偏轻
16~18	偏轻
18~24	正常体重
24~29	超重
20~35	严重超重
35以上	非常严重超重

Enter weight in pounds: 146 
Enter height in inches: 70 
Your BMI is 20.948603801493316
You are normal weight



line#	weight	height	WeightInKilograms	heightInMeters	bmi	output
9	146					
13		70				
19			66.22448602			
20				1.778		
21					20.9486	
25						Your BMI is 20.95
31						You are normal weight

第15~16行定义两个常量KILOGRAMS_PER_POUND和METERS_PER_INCH。这里使用常量可以使程序易于阅读。

3.11 问题：计算税款

美国国家联邦个人收入所得税是基于纳税人登记的身份和可征税收入计算的。纳税人登记的身份有四种：单身纳税人、已婚共同纳税人、已婚单独纳税人和家庭户主纳税人。税率会随年变化。表3-2给出2009年的税率。也就是说，如果你是单身纳税人，可征税收入为10 000美元，那么可征税收入的前8350美元的税率为10%，而剩下的1650美元的税率为15%。所以，你该付的税金为1082.5美元。

表3-2 2009年美国国家联邦个人收入所得税税率表

税率	单身纳税人	已婚共同纳税人或证实的鳏寡	已婚单独纳税人	家庭户主纳税人
10%	\$0~\$8350	\$0~\$16 700	\$0~\$8350	\$0~\$11 950
15%	\$8351~\$33 950	\$16 701~\$67 900	\$8351~\$33 950	\$11 951~\$45 500
25%	\$33 951~\$52 250	\$67 901~\$137 050	\$33 951~\$68 525	\$45 501~\$117 450
28%	\$82 251~\$171 550	\$137 051~\$208 850	\$68 525~\$104 425	\$117 451~\$190 200
33%	\$171 551~\$372 950	\$208 851~\$372 950	\$104 426~\$186 475	\$190 201~\$372 950
35%	\$372 951+	\$372 951+	\$186 476+	\$372 951+

你将要编写一个程序来计算个人收入税。程序应该提示用户输入登记的身份以及可征税收入，然后计算出税款。输入0表示单身纳税人，1表示已婚共同纳税人，2为已婚单独纳税人，3为家庭户主纳税人。

程序要计算基于登记身份的可征税收入。登记的身份可以使用if语句来决定，如下所示：

```
if (status == 0) {
    // Compute tax for single filers
}
else if (status == 1) {
    // Compute tax for married filing jointly
}
else if (status == 2) {
    // Compute tax for married filing separately
}
else if (status == 3) {
    // Compute tax for head of household
}
else {
    // Display wrong status
}
```

对每个登记的身份都有六种税率。每个税率应用于某个可征税收入范围内。例如：对于有可征税收

入400 000美元的单身登记人来说, 8350美元的税率是10%, 从8350到33 950之间税率为15%, 从33 950到82 250之间税率是25%, 从82 250到171 550之间税率是28%, 从171 550到372 950之间税率是33%而从372 950到400 000之间税率是35%。

程序清单3-6给出计算单身纳税人税款的解决方案, 完整的解决方案留作练习。

程序清单3-6 ComputeTax.java

```

1 import java.util.Scanner;
2
3 public class ComputeTax {
4     public static void main(String[] args) {
5         // Create a Scanner
6         Scanner input = new Scanner(System.in);
7
8         // Prompt the user to enter filing status
9         System.out.print(
10             "(0-single filer, 1-married jointly,\n" +
11             "2-married separately, 3-head of household)\n" +
12             "Enter the filing status: ");
13         int status = input.nextInt();
14
15         // Prompt the user to enter taxable income
16         System.out.print("Enter the taxable income: ");
17         double income = input.nextDouble();
18
19         // Compute tax
20         double tax = 0;
21
22         if (status == 0) { // Compute tax for single filers
23             if (income <= 8350)
24                 tax = income * 0.10;
25             else if (income <= 33950)
26                 tax = 8350 * 0.10 + (income - 8350) * 0.15;
27             else if (income <= 82250)
28                 tax = 8350 * 0.10 + (33950 - 8350) * 0.15 +
29                     (income - 33950) * 0.25;
30             else if (income <= 171550)
31                 tax = 8350 * 0.10 + (33950 - 8350) * 0.15 +
32                     (82250 - 33950) * 0.25 + (income - 82250) * 0.28;
33             else if (income <= 372950)
34                 tax = 8350 * 0.10 + (33950 - 8350) * 0.15 +
35                     (82250 - 33950) * 0.25 + (171550 - 82250) * 0.28 +
36                     (income - 171550) * 0.33;
37             else
38                 tax = 8350 * 0.10 + (33950 - 8350) * 0.15 +
39                     (82250 - 33950) * 0.25 + (171550 - 82250) * 0.28 +
40                     (372950 - 171550) * 0.33 + (income - 372950) * 0.35;
41         }
42         else if (status == 1) { // Compute tax for married file jointly
43             // Left as exercise
44         }
45         else if (status == 2) { // Compute tax for married separately
46             // Left as exercise
47         }
48         else if (status == 3) { // Compute tax for head of household
49             // Left as exercise
50         }
51         else {
52             System.out.println("Error: invalid status");
53             System.exit(0);
54         }
55
56         // Display the result
57         System.out.println("Tax is " + (int)(tax * 100) / 100.0);

```



```
58 }
59 }
```

```
(0-single filer, 1-married jointly,
2-married separately, 3-head of household)
Enter the filing status: 0
Enter the taxable income: 400000
Tax is 117683.5
```



line#	status	income	tax	output
13	0			
17		400000		
20			0	
38			117683.5	
57				Tax is 117683.5



这个程序接收纳税人身份和可征税收入。多重选择if语句（第22、42、45、48和51行）判断登记人的身份，并根据登记身份计算税款。

`System.exit(0)`（第53行）是在`System`类中定义的。调用这个方法可以终止程序。参数0表明程序正常结束。

赋初始值0给tax（第20行）。因为所有其他给tax赋值的语句都在if语句中，所以，如果tax没有初值，就会出现一个语法错误。编译器认为这些语句不可能执行，因此会报告一个语法错误。

为了测试程序，应该提供覆盖所有情况的输入。对这个程序而言，输入应该涵盖所有的身份（0、1、2、3）。针对每一种身份，应对6个范围中的每种情况测试税款。这样，总共会有24种情况。

提示 对所有的程序都应该先编写小量代码然后进行测试，之后再继续添加更多的代码。这个过程称为递进式开发和测试（incremental development and testing）。这个方法使得调试变得更加容易，因为错误很可能就你刚刚添加进去的新代码中。

3.12 逻辑运算符

有时候，是否执行一条语句是由几个条件的组合来决定的。可以使用逻辑运算符组合这些条件。逻辑运算符（logical operator）也称为布尔运算符（boolean operator），是对布尔值进行的运算，它会创建新的布尔值。表3-3列出了布尔运算符清单。表3-4定义了非运算符（!）。非运算符（!）对true取反是false，而false取反之后则是true。表3-5定义了与运算符（&&）。当且仅当两个操作数都为true时，这两个布尔型操作数的与（&&）为true。表3-6定义了或运算符（||），当至少有一个操作数为true时，两个布尔型操作数的或（||）为true。表3-7定义了异或运算符（^）。当且仅当两个操作数具有不同的布尔值时，两个布尔型操作数的异或（^）才为true。

表3-3 布尔运算符

运 算 符	名 称	说 明
!	非	逻辑非
&&	与	逻辑与
	或	逻辑或
^	异或	逻辑异或

表3-4 运算符!的真值表

操作数p	非的结果!p	举例 (假设age=24, gender='F')
true	false	!(age>18)为false, 因为 (age>18) 为true
false	true	!(gender=='M')为true, 因为 (gender=='M') 为false

表3-5 运算符&&的真值表

操作数p1	操作数p2	与的结果p1&p2	举例 (假设age=24, gender='F')
false	false	false	(age>18) && (gender=='F') 为true, 因为 (age>18) 和 (gender=='F') 都为true
false	true	false	
true	false	false	(age>18) && (gender!='F') 为false, 因为 (gender!='F') 为false
true	true	true	

表3-6 或运算符||的真值表

操作数p1	操作数p2	或的结果p1 p2	举例 (假设age=24, gender='F')
false	false	false	(age>34) (gender=='F') 为true, 因为 (gender=='F') 为true
false	true	true	
true	false	true	(age>34) (gender=='M') 为false, 因为 (age>34) 和 (gender=='M') 都是false
true	true	true	

表3-7 异或运算符^的真值表

操作数p1	操作数p2	异或的结果p1^p2	举例 (假设age=24, gender='F')
false	false	false	(age>34) ^ (gender=='F') 为true, 因为 (age>34) 为false而 (gender=='F') 为true
false	true	true	
true	false	true	(age>34) ^ (gender=='M') 为false, 因为 (age>34) 和 (gender=='M') 都为false
true	true	false	

程序清单3-7给出的程序检验一个数是否能被2和3整除, 是被2还是3整除, 是否只能被2或3两者中的一个整除。

程序清单3-7 TestBooleanOperators.java

```

1 import java.util.Scanner;
2
3 public class TestBooleanOperators {
4     public static void main(String[] args) {
5         // Create a Scanner
6         Scanner input = new Scanner(System.in);
7
8         // Receive an input
9         System.out.print("Enter an integer: ");
10        int number = input.nextInt();
11    }

```

```

12 System.out.println("Is " + number +
13     "\n\tdivisible by 2 and 3? " +
14     (number % 2 == 0 && number % 3 == 0)
15     + "\n\tdivisible by 2 or 3? " +
16     (number % 2 == 0 || number % 3 == 0) +
17     "\n\tdivisible by 2 or 3, but not both? "
18     + (number % 2 == 0 ^ number % 3 == 0));
19 }
20 }

```

```

Enter an integer: 18
Is 18
    divisible by 2 and 3? true
    divisible by 2 or 3? true
    divisible by 2 or 3, but not both? false

```



在第12~18行对子串进行链接形成一个长的字符串。三个\n字符将字符串显示在四行中。(number%2==0&&number%3==0) (第14行) 检验一个数是否能被2和3整除。(number%2==0||number%3==0) (第16行) 检验一个数是否能被2或3整除。(number%2==0^ number%3==0) (第20行) 检验一个数是否能被2或3整除但又不能同时被这两者整除。

警告 从数学的角度看, 表达式

```
1 <= numberOfDaysInAMonth <= 31
```

是正确的。但是, 在Java中它是错的, 因为1<=numberOfDaysInAMonth得到的是一个布尔值的结果, 它是不能和31进行比较的。下面的两个操作数 (一个布尔值和一个数值) 是不兼容的。正确的Java表达式是:

```
(1 <= numberOfDaysInAMonth) && (numberOfDaysInAMonth <= 31)
```

注意 如前一章所示, 一个char型值可以转换为int型值, 反之亦然。但是, 一个布尔型值不能转换为其他类型的值, 其他类型的值也不能转换为布尔类型值。

注意 德模佛定理是以印度出生的英国数学家和逻辑学家奥古斯都·德·模佛来命名的 (1806—1871), 这个定理可以用来简化表达式。定义表述如下:

!(condition1 && condition2)和!condition1 || ! condition2是一样的。

!(condition1 || condition2)和!condition1 && ! condition2是一样的。

例如:

!(n == 2 || n==3)和n!= 2 && n!= 3是一样的。

!(n % 2 == 0 && n % 3 == 0)和n % 2 != 0 || n % 3 != 0是一样的。

如果运算符&&的操作数之一为false, 那么表达式就是false; 如果运算符||的操作数之一为true, 那么表达式就是true。Java利用这些特性来提高这些运算符的效率。当计算p1&&p2时, Java先计算p1, 如果p1为true再计算p2; 如果p1为false, 则不再计算p2。当计算p1||p2时, Java先计算p1, 如果p1为false再计算p2; 如果p1为true, 则不再计算p2。因此, &&又称为条件与 (conditional AND) 运算符或短路与 (short-circuit AND) 运算符, 而||称为条件或 (conditional OR) 运算符或短路或 (short-circuit OR) 运算符。

3.13 问题: 判定闰年

如果某年可以被4整除而不能被100整除, 或者可以被400整除, 那么这一年就是闰年 (leap year)。所以, 可以使用下面的布尔表达式判定某年是否为闰年:

```
// A leap year is divisible by 4
boolean isLeapYear = (year % 4 == 0);

// A leap year is divisible by 4 but not by 100
isLeapYear = isLeapYear && (year % 100 != 0);

// A leap year is divisible by 4 but not by 100 or divisible by 400
isLeapYear = isLeapYear || (year % 400 == 0);
```


或者可以将这些表达式组合在一起，如下所示：


```
isLeapYear = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
```

程序清单3-8给出的程序让用户输入一个年份，然后判断它是否是闰年。

程序清单3-8 LeapYear.java

```
1 import java.util.Scanner;
2
3 public class LeapYear {
4     public static void main(String[] args) {
5         // Create a Scanner
6         Scanner input = new Scanner(System.in);
7         System.out.print("Enter a year: ");
8         int year = input.nextInt();
9
10        // Check if the year is a leap year
11        boolean isLeapYear =
12            (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
13
14        // Display the result
15        System.out.println(year + " is a leap year? " + isLeapYear);
16    }
17 }
```

Enter a year: 2008 
 2008 is a leap year? true

Enter a year: 2002 
 2002 is a leap year? false



3.14 问题：彩票

假设你想开发一个玩彩票的游戏，程序随机地产生一个两位数的彩票，提示用户输入一个两位数，然后按照下面的规则判定用户是否能赢：

- 1) 如果用户的输入数匹配彩票的实际顺序，奖金为10 000美金。
- 2) 如果用户输入的所有数字匹配彩票的所有数字，奖金为3000美金。
- 3) 如果用户输入的一个数字匹配彩票的一个数字，奖金为1000美金。

完整的程序如程序清单3-9所示。

程序清单3-9 Lottery.java


```
1 import java.util.Scanner;
2
3 public class Lottery {
4     public static void main(String[] args) {
5         // Generate a lottery
6         int lottery = (int)(Math.random() * 100);
7
8         // Prompt the user to enter a guess
9         Scanner input = new Scanner(System.in);
10        System.out.print("Enter your lottery pick (two digits): ");
11        int guess = input.nextInt();
12    }
```




```

13 // Get digits from lottery
14 int lotteryDigit1 = lottery / 10;
15 int lotteryDigit2 = lottery % 10;
16
17 // Get digits from guess
18 int guessDigit1 = guess / 10;
19 int guessDigit2 = guess % 10;
20
21 System.out.println("The lottery number is " + lottery);
22
23 // Check the guess
24 if (guess == lottery)
25     System.out.println("Exact match: you win $10,000");
26 else if (guessDigit2 == lotteryDigit1
27         && guessDigit1 == lotteryDigit2)
28     System.out.println("Match all digits: you win $3,000");
29 else if (guessDigit1 == lotteryDigit1
30         || guessDigit1 == lotteryDigit2
31         || guessDigit2 == lotteryDigit1
32         || guessDigit2 == lotteryDigit2)
33     System.out.println("Match one digit: you win $1,000");
34 else
35     System.out.println("Sorry, no match");
36 }
37 }

```

Enter your lottery pick (two digits): 45 
 The lottery number is 12
 Sorry, no match



Enter your lottery pick: 23 
 The lottery number is 34
 Match one digit: you win \$1,000



line#	6	11	14	15	18	19	33
variable							
lottery	34						
guess		23					
lotteryDigit1			3				
lotteryDigit2				4			
guessDigit1					2		
guessDigit2						3	
output							Match one digit: you win \$1,000



程序使用`random()`方法(第6行)创建一个彩票,然后提示用户输入他自己的猜测值(第11行)。注意,因为`guess`是一个两位数,所以`guess%10`能得到`guess`的最后一位数,而`guess/10`能得到`guess`的第一位数(第18~19行)。

程序按照以下顺序检测猜测值和彩票:

- 1) 首先检测猜测值是否精确匹配彩票(第24行)。
- 2) 如果没有匹配,就检测猜测数的逆序是否匹配彩票(第26~27行)。
- 3) 如果还不匹配,就检测是否有一个数字在彩票中(第29~32行)。
- 4) 如果还没有,就表明都不匹配。

3.15 switch语句

程序清单3-6中的if语句是根据单独的一个true或false条件做出选择的。根据变量status的值,会有四种计算税金的情况。为了全面考虑所有的情况,需要使用嵌套的if语句。过多地使用嵌套的if语句会使程序很难阅读。Java提供switch语句来有效地处理多重条件的问题。可以使用下述switch语句替换程序清单3-6中的嵌套if语句:

```
switch (status) {
    case 0: compute taxes for single filers;
            break;
    case 1: compute taxes for married filing jointly;
            break;
    case 2: compute taxes for married filing separately;
            break;
    case 3: compute taxes for head of household;
            break;
    default: System.out.println("Errors: invalid status");
            System.exit(0);
}
```

上面的switch语句的流程图如图3-5所示。

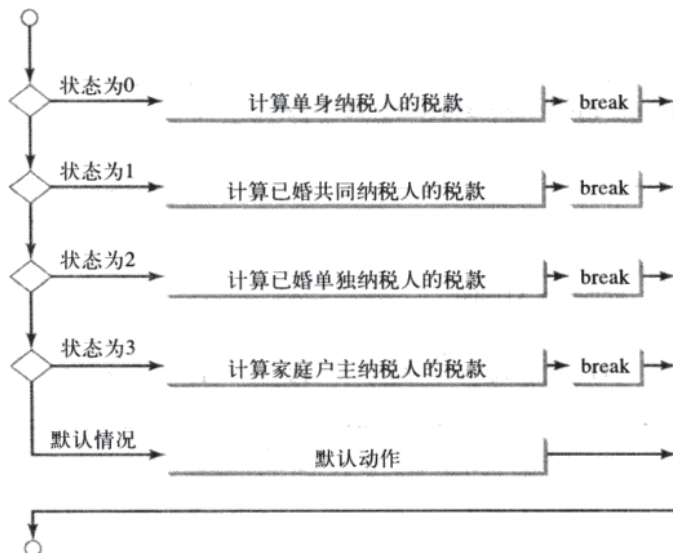


图3-5 switch语句检验所有的情况并执行匹配条件时的语句

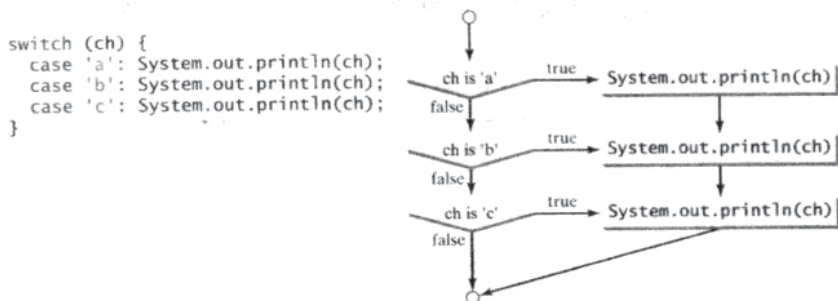
这条语句依次检查status是否能匹配0、1、2或3。如果匹配,就计算相应的税金;如果不匹配,就显示一条消息。下面是switch语句的完整语法:

```
switch (switch表达式) {
    case 值1: 语句 (组) 1;
            break;
    case 值2: 语句 (组) 2;
            break;
    ...
    case 值N: 语句 (组) N;
            break;
    default: 默认情况下执行的语句 (组)
}
```

switch语句遵从下述规则:

- 1) switch表达式必须能计算出一个char、byte、short或int型值，并且必须总是要用括号括住。
- 2) value1, ..., valueN必须与switch表达式的值具有相同的数据类型。注意：value1, ..., valueN都是常量表达式，也就是说这里的表达式是不能包含变量的，例如，不允许出现1+x。
- 3) 当switch表达式的值与case语句的值相匹配时，执行从该case开始的语句，直到遇到一个break语句或到达该switch语句的末端。
- 4) 关键字break是可选的。break语句会立即终止整个switch语句。
- 5) 默认情况（default）是可选的，用来指定没有一个指定的case与switch表达式匹配时应该采取的操作。
- 6) case语句是顺序检测的，但是，这些case的出现顺序（包括默认情况）是不重要的。然而，良好的编程风格是将情况按照逻辑顺序排列，并把默认情况放在最后。

警告 不要忘记在需要的时候使用break语句。一旦匹配其中一个case，就从匹配的case处开始执行，直到遇到break语句或到达switch语句的末端。这种现象称为向下贯通行为（fall-through behavior）。例如，如果ch是'a'，下面的代码将字母a打印三次：



提示 为了避免程序设计错误，提高代码的可维护性，如果故意省略break，在case子句后添加注释是一个好的做法。

3.16 条件表达式

有时可能需要给有特定条件限制的变量赋值。例如：下面的语句在x大于0时给y赋值1；当x小于等于0时给y赋值-1。

```
if (x > 0)
    y = 1;
else
    y = -1;
```

在这个例子中，还可以选择使用如下的条件表达式，也能达到同样的效果。

```
y = (x > 0) ? 1 : -1;
```

条件表达式是一种完全不同的风格，在语句中没有明确出现if。该语法如下所示：

boolean-expression ? expression1 : expression2; (布尔表达式? 表达式1: 表达式2)

如果布尔表达式的值为true，则条件表达式的结果为表达式expression1；否则，结果为表达式expression2。

假设希望将num1和num2中较大的数赋值给max，可以利用条件表达式，只需编写一条语句：

```
max = (num1 > num2) ? num1 : num2;
```

另外举一个例子，如果num是偶数，下面的语句就显示信息“num is even”；否则显示“num is odd”：

```
System.out.println((num % 2 == 0) ? "num is even" : "num is odd");
```

注意 符号?和:在条件表达式中同时出现。它们构成一种条件运算符，因为操作数有三个，所以称为三目运算符 (ternary operator)。它是Java中唯一的三目运算符。

3.17 格式化控制台输出

如果希望显示浮点值小数点后两位，那么可以如下编写代码：

```
double x = 2.0 / 3;
System.out.println("x is " + (int)(x * 100) / 100.0);
```

x is 0.66



但是，完成这个任务更好的方法是使用printf方法格式化输出。调用这个方法的话语是：

```
System.out.printf(format, item1, item2, ..., itemk)
```

这里的format是指一个子串和格式标识符构成的字符串。

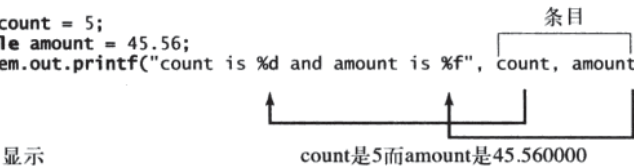
格式标识符指定每个条目应该如何显示。这里的条目可以是数值、字符、布尔值或字符串。一个标识符是以百分号(%)开头的转换码。表3-8列出了一些常用的简单标识符。

表3-8 常用的标识符

标 识 符	输 出	举 例
%b	布尔值	true或false
%c	字符	'a'
%d	十进制整数	200
%f	浮点数	45.460000
%e	标准科学记数法形式的数	4.556000e+01
%s	字符串	"Java is cool"

下面是一个例子：

```
int count = 5;
double amount = 45.56;
System.out.printf("count is %d and amount is %f", count, amount);
```



条目与标识符必须在次序、数量和类型上匹配。例如：count的标识符应该是%d，而amount的标识符应该是%f。默认情况下，浮点值显示小数点后6位。可以在标识符中指定宽度和精度，如表3-9中的例子所示。

表3-9 指定宽度和精度的例子

举 例	输 出
%5c	输出字符并在这个字符条目前面加4个空格
%6b	输出布尔值，在false前加一个空格，在true前加两个空格
%5d	输出整数条目，宽度至少为5。如果该条目的数字位数小于5，就在前面加空格。如果该条目的位数大于5，则自动增加宽度
%10.2f	输出的浮点数宽度至少为10，包括小数点和小数点后两位。这样，给小数点前分配了7位。如果该项小数点前的位数小于7，就在数字前面加空格。如果该项小数点前的位数大于7，则自动增加宽度
%10.2e	输出的浮点条目的宽度至少为10，包括小数点、小数点后两位和指数部分。如果按科学记数法显示的数字小于10，就给数前加空格
%12s	输出的字符串宽度至少为12。如果该字符串条目小于12个字符，就在该条目前加空格。如果该字符串条目多于12个字符，则自动增加宽度

可以使用printf方法改写在本节开始部分出现的，用于显示浮点值小数点后两位的代码，如下所示：

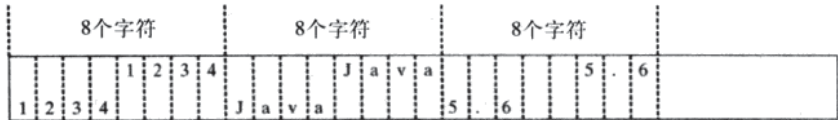
```
double x = 2.0 / 3;
System.out.printf("x is %4.2f", x);
display      x is 0.67
```



默认情况下，输出是右对齐的。可以在标识符中放一个负号（-），表明该条目在特定区域中的输出是左对齐的。例如，以下语句：

```
System.out.printf("%8d%8s%8.1f\n", 1234, "Java", 5.6);
System.out.printf("%-8d%-8s%-8.1f\n", 1234, "Java", 5.6);
```

显示



警告 条目与标识符必须在类型上严格匹配。对应于标识符%f或%e的条目必须是浮点型值，例如：是40.0而不是40。因此，int型变量不能匹配%f或%e。

提示 使用符号%来标记标识符，要在格式字符串里输出直接量%，就要使用%%。

3.18 运算符的优先级和结合方向

运算符的优先级和结合方向决定了运算符的计算顺序。假设有这样一个表达式：

```
3 + 4 * 4 > 5 * (4 + 3) - 1
```

它的值是多少呢？这些运算符的执行顺序是什么呢？

从数学角度看，应该首先计算括号中的表达式（括号可以嵌套，在嵌套的情况下，先计算里层括号中的表达式）。当计算没有括号的表达式时，运算符会依照优先级规则和结合规则进行运算。

优先级规则定义了运算符的先后次序，如表3-10所示，它包含了目前所学的所有运算符。它们从上到下按优先级递减的方式排列。优先级相同的运算符排在同一行。（Java运算符及其优先级的完整列表参见附录C。）

如果相邻运算符的优先级相同，则结合方向（associativity）决定它们的执行顺序。除了赋值运算符之外，所有的二元运算符都是左结合的（left-associative）。例如，由于+和-的优先级相同并且都是左结合的，所以表达式：

$$a - b + c - d \quad \text{等价于} \quad ((a - b) + c) - d$$

表3-10 运算符优先级表

优 先 级	运 算 符
最高级	var++和var--（后置运算符） +、-（一元加号和一元减号）、++var、--var（前置运算符） (type)（类型转换） ！（非） *、/、%（乘法、除法和求余运算） +、-（二元加法和减法） <、<=、>、>=（比较运算符）

(续)

优 先 级	运 算 符
	<code>==</code> 、 <code>!=</code> (相等运算符)
	<code>^</code> (异或)
	<code>&&</code> (条件与)
	<code> </code> (条件或)
最低级	<code>=</code> 、 <code>+=</code> 、 <code>-=</code> 、 <code>*=</code> 、 <code>/=</code> 、 <code>%=</code> (赋值运算符)

赋值运算符是右结合的 (right-associative)。因此, 表达式:

$$a = b += c = 5 \quad \text{等价于} \quad a = (b += (c = 5))$$

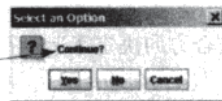
假设赋值前`a`, `b`和`c`都是1, 在计算表达式之后, `a`变成6, `b`变成6, 而`c`变成5。注意: 左结合对赋值运算符而言是没有什么意义的。

注意 Java有自己内部计算表达式的方式。Java计算的结果和它对应的算术计算是一样的。感兴趣的读者可以参考补充材料III.B, 以获得更多关于Java的后台是如何计算表达式的讨论。

3.19 (GUI) 确认对话框

你已经使用过`showMessageDialog`来显示一个消息对话框, 用`showInputDialog`来显示一个输入对话框。有时候, 使用确认对话框来回答问题是非常有用的。可以使用下面的语句创建一个确认对话框:

```
int option =
JOptionPane.showConfirmDialog
(null, "Continue");
```



当点击按钮时, 这个方法会返回一个选择值。点击`Yes`按钮返回的值是`JOptionPane.YES_OPTION(0)`, 点击`No`按钮返回的值是`JOptionPane.NO_OPTION(1)`, 点击`Cancel`按钮返回的值是`JOptionPane.CANCEL_OPTION(2)`。

可以使用确认对话框改写程序清单3-3中猜生日的程序, 如程序清单3-10所示。图3-6给出要猜出的生日是19时, 程序的一个运行示例。

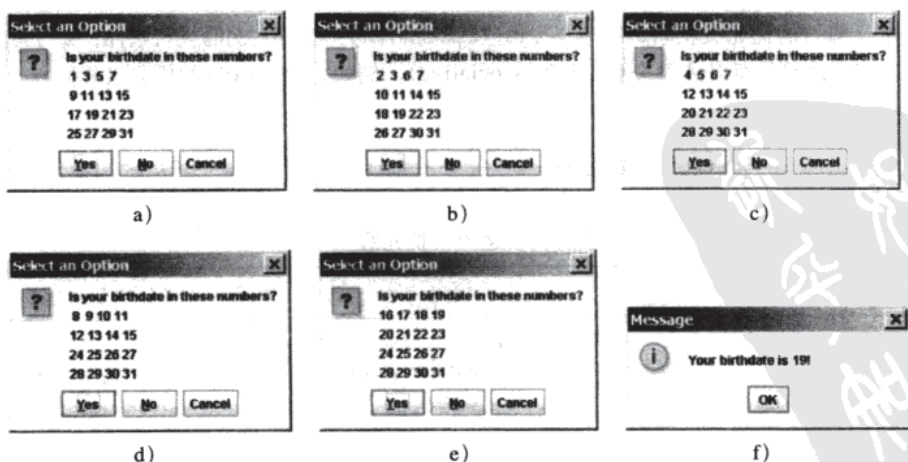


图3-6 在a中点击Yes, 在b中点击Yes, 在c中点击No, 在d中点击No, 在e中点击Yes

程序清单3-10 GuessBirthdayUsingConfirmationDialog.java

```
1 import javax.swing.JOptionPane;
2
3 public class GuessBirthdayUsingConfirmationDialog {
4     public static void main(String[] args) {
5         String set1 =
6             " 1 3 5 7\n" +
7             " 9 11 13 15\n" +
8             "17 19 21 23\n" +
9             "25 27 29 31";
10
11         String set2 =
12             " 2 3 6 7\n" +
13             "10 11 14 15\n" +
14             "18 19 22 23\n" +
15             "26 27 30 31";
16
17         String set3 =
18             " 4 5 6 7\n" +
19             "12 13 14 15\n" +
20             "20 21 22 23\n" +
21             "28 29 30 31";
22
23         String set4 =
24             " 8 9 10 11\n" +
25             "12 13 14 15\n" +
26             "24 25 26 27\n" +
27             "28 29 30 31";
28
29         String set5 =
30             "16 17 18 19\n" +
31             "20 21 22 23\n" +
32             "24 25 26 27\n" +
33             "28 29 30 31";
34
35         int day = 0;
36
37         // Prompt the user to answer questions
38         int answer = JOptionPane.showConfirmDialog(null,
39             "Is your birthday in these numbers?\n" + set1);
40
41         if (answer == JOptionPane.YES_OPTION)
42             day += 1;
43
44         answer = JOptionPane.showConfirmDialog(null,
45             "Is your birthday in these numbers?\n" + set2);
46
47         if (answer == JOptionPane.YES_OPTION)
48             day += 2;
49
50         answer = JOptionPane.showConfirmDialog(null,
51             "Is your birthday in these numbers?\n" + set3);
52
53         if (answer == JOptionPane.YES_OPTION)
54             day += 4;
55
56         answer = JOptionPane.showConfirmDialog(null,
57             "Is your birthday in these numbers?\n" + set4);
58
59         if (answer == JOptionPane.YES_OPTION)
60             day += 8;
61
62         answer = JOptionPane.showConfirmDialog(null,
63             "Is your birthday in these numbers?\n" + set5);
```

数字图书馆
PDG


```

64
65     if (answer == JOptionPane.YES_OPTION)
66         day += 16;
67
68     JOptionPane.showMessageDialog(null, "Your birthday is " +
69         day + "!");
70 }
71 }

```

该程序显示确认对话框，提示用户回答某个数字是否在集合1中（第38行），是否在集合2中（第44行），是否在集合3中（第50行），是否在集合4中（第56行）以及是否在集合5中（第62行）。如果得到的答案是Yes，就将该集合中的第一个数加到day中（第42、48、54、60和66行）。

关键术语

Boolean expression（布尔表达式）	fall-through behavior（向下贯通行为）
Boolean value（布尔值）	operator associativity（运算符结合方向）
boolean type（boolean类型）	operator precedence（运算符优先级）
break statement（break语句）	selection statement（选择语句）
conditional operator（条件运算符）	short-circuit evaluation（短路求值）
dangling-else ambiguity（悬空else歧义）	

本章小结

- 一个boolean变量可以存储值true或false。
- 关系运算符（<、<=、==、!=、>、>=）和数值及字符一起运算，然后产生一个布尔值。
- 布尔运算符&&、||、|和^对布尔值和布尔变量进行计算。
- 当对p1&&p2求值时，Java先求p1的值，如果p1为true，再对p2求值；如果p1为false，就不在对p2求值。当对p1||p2求值时，Java先求p1的值，如果p1为false，再对p2求值；如果p1为true，就不在对p2求值。因此，&&也称为条件与运算符或短路与运算符，而||也称为条件或运算符或短路或运算符。
- 使用选择语句可以对有可选择路径的情况进行程序设计。选择语句有以下几种类型：if语句、if...else语句、嵌套if语句、switch语句和条件表达式。
- 各种if语句都是基于布尔表达式来控制决定的。根据表达式的值是true或false，选择两种可能路径中的一种。
- switch语句根据switch表达式的类型char、byte、short或int来控制决定。
- 在switch语句中，关键字break是可选的，但它通常用在每个分支的结尾，以中止执行switch语句的剩余部分。如果没有出现break语句，则执行接下来的case语句。

复习题

3.2节

3.1 列出六个比较运算符。

3.2 是否允许做以下的类型转换？如果允许，给出转换结果。

```

boolean b = true;
i = (int)b;

int i = 1;
boolean b = (boolean)i;

```


3.3~3.11节

3.3 如果number分别是30和35,那么图a中的代码和图b中的代码会打印出什么结果?

```
if (number % 2 == 0)
    System.out.println(number + " is even.");
    System.out.println(number + " is odd.");
```

a)

```
if (number % 2 == 0)
    System.out.println(number + " is even.");
else
    System.out.println(number + " is odd.");
```

b)

3.4 假设 $x=3$ 而 $y=2$,若有输出,给出下面代码的输出结果。当 $x=3$ 而 $y=4$ 时,输出结果是什么?当 $x=2$ 而 $y=2$ 时输出又是什么?画出下列代码的流程图:

```
if (x > 2) {
    if (y > 2) {
        z = x + y;
        System.out.println("z is " + z);
    }
}
else
    System.out.println("x is " + x);
```

3.5 下面哪些语句是等价的?哪些语句缩进正确?

```
if (i > 0) if
(j > 0)
x = 0; else
if (k > 0) y = 0;
else z = 0;
```

a)

```
if (i > 0) {
    if (j > 0)
        x = 0;
    else if (k > 0)
        y = 0;
}
else
    z = 0;
```

b)

```
if (i > 0)
    if (j > 0)
        x = 0;
    else if (k > 0)
        y = 0;
    else
        z = 0;
```

c)

```
if (i > 0)
    if (j > 0)
        x = 0;
    else if (k > 0)
        y = 0;
else
    z = 0;
```

d)

3.6 假设 $x=2$ 而 $y=3$,若有输出,给出下列代码的输出结果。如果 $x=3$ 而 $y=2$,输出是什么?当 $x=3$ 而 $y=3$ 时,输出又是什么?

提示 首先正确缩进这些语句。

```
if (x > 2)
    if (y > 2) {
        int z = x + y;
        System.out.println("z is " + z);
    }
else
    System.out.println("x is " + x);
```

3.7 下面这两个语句等价吗?

```
if (income <= 10000)
    tax = income * 0.1;
else if (income <= 20000)
    tax = 1000 +
        (income - 10000) * 0.15;
```

```
if (income <= 10000)
    tax = income * 0.1;
else if (income > 10000 &&
        income <= 20000)
    tax = 1000 +
        (income - 10000) * 0.15;
```

3.8 下面哪些是调用Math.random()后可能出现的输出?

323.4, 0.5, 34, 1.0, 0.0, 0.234

3.9 如何产生一个满足条件 $0 \leq i < 20$ 的随机整数 i ?如何产生一个满足条件 $10 \leq i < 20$ 的随机整数 i ?如何产生一个满足条件 $10 \leq i \leq 50$ 的随机整数 i ?

3.10 编写一个if语句实现:如果 y 大于0就给 x 赋值为1。

3.11 (1) 编写一个if语句实现:如果score大于90就给pay增加3%。(2) 编写一个if语句实现:如果score大于90就给pay增加3%,若不满足条件就给pay增加1%。

3.12 下面代码中的错误是什么?

82 • 第3章 选 择

```

if (score >= 60.0)
    grade = 'D';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 90.0)
    grade = 'A';
else
    grade = 'F';

```

3.13 使用布尔表达式改写下面的语句:

```

if (count % 10 == 0)
    newLine = true;
else
    newLine = false;

```

3.12~3.14节

3.14 假设x为1, 给出下列布尔表达式的结果:

```

(true) && (3 > 4)
!(x > 0) && (x > 0)
(x > 0) || (x < 0)
(x != 0) || (x == 0)
(x >= 0) || (x < 0)
(x != 1) == !(x == 1)

```

3.15 编写一个布尔表达式满足: 若变量num中存储的数值在1到100之间时, 表达式的值为true。

3.16 编写一个布尔表达式满足: 若变量num中存储的数值在1到100之间或值为负数时, 表达式的值为true。

3.17 假设x和y都是int类型, 下面哪些是合法的Java表达式?

```

x > y > 0
x = y && y
x /= y
x or y
x and y
(x != 0) || (x = 0)

```

3.18 假定x是1, 经过下面的表达式运算后, x的值是多少?

```

(x >= 1) && (x++ > 1)
(x > 1) && (x++ > 1)

```

3.19 如果ch是'A'、'p'、'E'或'5', 那么表达式ch>='A'&&ch<='Z'的值是什么?

3.20 假设运行下面的程序时, 从控制台输入的是2 3 6, 那么输出是什么?

```

public class Test {
    public static void main(String[] args) {
        java.util.Scanner input = new java.util.Scanner(System.in);
        double x = input.nextDouble();
        double y = input.nextDouble();
        double z = input.nextDouble();

        System.out.println("(x < y && y < z) is " + (x < y && y < z));
        System.out.println("(x < y || y < z) is " + (x < y || y < z));
        System.out.println("!(x < y) is " + !(x < y));
        System.out.println("(x + y < z) is " + (x + y < z));
        System.out.println("(x + y < z) is " + (x + y < z));
    }
}

```

3.21 编写当年龄age大于13且小于18时结果为true的布尔表达式。

3.22 编写当体重weight大于50或身高大于160时结果为true的布尔表达式。

3.23 编写当体重weight大于50且身高height大于160时结果为true的布尔表达式。

3.24 编写当体重weight大于50或身高height大于160, 但不能同时满足这两个条件时, 结果为true的布尔表达式。

3.15节

3.25 switch变量需要什么类型的数据? 如果在执行完case语句之后没有使用关键字break, 那么下一条要执行的语句是什么? 可以把switch语句转换成等价的if语句吗? 反过来可以将if语句转换成等价的switch语句吗? 使用switch语句的优点有哪些?

3.26 执行下列switch语句之后, y是多少?

```
x = 3; y = 3;
switch (x + 3) {
    case 6: y = 1;
    default: y += 1;
}
```

3.27 使用switch语句改写下面的if语句, 并画出switch语句的流程图:

```
if (a == 1)
    x += 5;
else if (a == 2)
    x += 10;
else if (a == 3)
    x += 16;
else if (a == 4)
    x += 34;
```

3.28 编写switch语句, 如果day是0、1、2、3、4、5、6, 那么String变量dayName被依次赋值为Sunday、Monday、Tuesday、Wednesday、Thursday、Friday、Saturday。

3.16节

3.29 使用条件运算符改写下面的语句:

```
if (count % 10 == 0)
    System.out.print(count + "\n");
else
    System.out.print(count + " ");
```

3.30 使用条件表达式改写下面的语句:

```
if (temperature > 90)
    pay = pay * 1.5;
else
    pay = pay * 1.1;
```

3.17节

3.31 输出布尔值、字符、十进制整数、浮点数和字符串的格式描述符分别是什么?

3.32 下面的语句错在哪里?

- (a) System.out.printf("%5d %d", 1, 2, 3);
- (b) System.out.printf("%5d %f", 1);
- (c) System.out.printf("%5d %f", 1, 2);

3.33 给出下面语句的输出。

- (a) System.out.printf("amount is %f %e\n", 32.32, 32.32);
- (b) System.out.printf("amount is %5.4f %5.4e\n", 32.32, 32.32);
- (c) System.out.printf("%6b\n", (1 > 2));
- (d) System.out.printf("%6s\n", "Java");
- (e) System.out.printf("%-6b%s\n", (1 > 2), "Java");
- (f) System.out.printf("%6b%-8s\n", (1 > 2), "Java");

3.34 如何创建一个格式化字符串?

3.18节

3.35 列出布尔运算符的优先级顺序。计算下面的表达式:

```
true || true && false
true && true || false
```

3.36 除=之外的所有二元运算符都是左结合的说法是真还是假?

3.37 计算下面的表达式:

```
2 * 2 - 3 > 2 && 4 - 2 > 5
2 * 2 - 3 > 2 || 4 - 2 > 5
```

3.38 $(x>0 \ \&\& \ x<10)$ 和 $((x>0)\&\&(x<10))$ 是否一样? $(x>0||x<10)$ 和 $((x>0)||x<10))$ 是否一样? $(x>0 \ || \ x<10 \ \&\& \ y<0)$ 和 $(x>0 \ || \ (x<10 \ \&\& \ y<0))$ 是否一样?

3.19节

3.39 如何显示一个确认对话框? 当调用 `JOptionPane.showConfirmDialog` 时会返回什么值?

编程练习题

教学注意 对于每一个练习题, 学生都应在编码之前仔细地分析解决该问题的需求及其策略设计。

教学注意 教师可以要求学生针对布置的练习题进行文档分析和设计。学生应该用自己的话来分析问题, 包括输入、输出以及需要计算什么, 并用伪代码描述如何解决该问题。

调试提示 在寻求帮助之前, 自己先阅读和解释一下程序, 然后手动使用几个具有代表性的输入跟踪程序, 或者使用某个IDE调试器跟踪程序。可以通过调试自己的错误来学习如何编程。

3.2节


*3.1 (代数方面: 解一元二次方程) 可以使用下面的公式求一元二次方程 $ax^2+bx+c=0$ 的两个根:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ 和 } r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$


b^2-4ac 称作一元二次方程的判别式。如果它是正值, 那么一元二次方程就有两个正根。如果它为0, 方程式就只有一个根。如果它是负值, 方程式无实根。

编写程序, 提示用户输入 a 、 b 和 c 的值, 并且显示基于判别式的结果。如果这个判别式为正, 显示两个根。如果判别式为0, 显示一个根。否则, 显示 “The equation has no real roots” (该方程式无实根)。


注意 可以使用 `Math.pow(x,0.5)` 来计算 \sqrt{x} 。下面是一些运行示例。

Enter a, b, c: 1.0 3.1 
The roots are -0.381966 and -2.61803




Enter a, b, c: 1 2.0 1 
The root is -1



Enter a, b, c: 1 2 3 
The equation has no real roots



3.2 (检查一个数字是否是偶数) 编写程序, 读入一个整数并检查它是不是偶数。这个程序的运行示例如下所示:

Enter an integer: 25 
Is 25 an even number? false



Enter an integer: 2000 
Is 2000 an even number? true



3.3~3.8节

*3.3 (代数方面: 求解 2×2 线性方程) 可以使用Cramer规则解下面的 2×2 线性方程组:

$$\begin{aligned} ax + by &= e \\ cx + dy &= f \end{aligned} \quad x = \frac{ed - bf}{ad - bc} \quad y = \frac{af - ec}{ad - bc}$$

编写程序，提示用户输入a、b、c、d、e和f，然后显示结果。如果 $ad - bc$ 为0，报告消息“The equation has no solution”（方程式无解）。

Enter a, b, c, d, e, f: 9.0 4.0 3.0 -5.0 -6.0 -21.0 Enter
x is -2.0 and y is 3.0



Enter a, b, c, d, e, f: 1.0 2.0 2.0 4.0 4.0 5.0 Enter
The equation has no solution



****3.4（游戏：学习加法）** 编写程序，产生两个100以下的整数，然后提示用户输入这两个整数的和。如果答案正确，程序报告结果true；否则，报告false。该程序类似于程序清单3-1。

****3.5（游戏：三个数的加法）** 程序清单3-1中的程序产生两个整数，并提示用户输入这两个整数的和。修改该程序使之能产生三个一位整数，然后提示用户输入这三个整数的和。

***3.6（医疗应用程序：BMI）** 修改程序清单3-5，让用户输入重量、英尺和英寸。例如：一个人身高是5英尺10英寸，输入的英尺值就是5而英寸值为10。

3.7（财务应用程序：整钱兑零） 修改程序清单2-10，使之只显示非零的币值单位，用单词的单数形式显示一个单位，例如1 dollar and 1 penny（1美元和1美分）；用单词的复数形式显示多于一个单位的值，例如2 dollars and 3 pennies（2美元和3美分）。（使用输入值23.67来测试该程序。）

***3.8（对三个整数排序）** 编写程序对三个整数排序。这些整数都由输入对话框输入，并分别存储在变量num1、num2和num3中。程序对这些数进行排序，使之满足 $num1 \leq num2 \leq num3$ 。

3.9（商业方面：检查ISBN） ISBN（国际标准书号）以前是一个10位整数 $d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}$ ，最后的一位 d_{10} 是校验和，它是使用下面的公式用另外9个数计算出来的：

$$(d_1 \times 1 + d_2 \times 2 + d_3 \times 3 + d_4 \times 4 + d_5 \times 5 + d_6 \times 6 + d_7 \times 7 + d_8 \times 8 + d_9 \times 9) \% 11$$

如果校验和为10，那么按照ISBN的习惯，最后一位应该表示为X。编写程序，提示用户输入前9个数，然后显示10位ISBN（包括前面起始位置的0）。程序应该读取一个整数输入。例如：输入的是013601267，那么程序就该显示0136012671。

***3.10（游戏：加法测验）** 程序清单3-4随机产生一个减法问题。修改这个程序，随机产生一个计算两个小于100的整数的加法问题。

3.9~3.19节

***3.11（给出一个月的总天数）** 编写程序，提示用户输入月份和年份，然后显示这个月的天数。例如：如果用户输入的月份是2而年份是2000，那么程序应该显示“February 2000 has 29 days”（2000年2月有29天）。如果用户输入的月份为3而年份为2005，那么程序就应该显示“March 2005 has 31 days”（2005年3月有31天）。

3.12（检测数字） 编写程序，提示用户输入一个整数，然后判断这个整数是否能被5和6都整除，或者不能被它们中的任何一个整除，或者只能被其中一个整除。下面是分别输入10、30和23时的一些运行示例。

```
10 is divisible by 5 or 6, but not both
30 is divisible by both 5 and 6
23 is not divisible by either 5 or 6
```

3.13（财务应用程序：计算税款） 程序清单3-6给出计算单身登记人税款的源代码。整个程序清单3-6给出的就是完整的源代码。

3.14（游戏：猜硬币的正反面） 编写程序，让用户猜一猜是硬币的正面还是反面。这个程序随机产生

一个整数0或者1，它们分别表示硬币的正面和反面。程序提示用户输入一个猜测值，然后报告这个猜测值是正确的还是错误的。

*3.15 (游戏:彩票) 修改程序清单3-9, 产生三位整数的彩票。程序提示用户输入一个三位整数, 然后依照下面的规则判定用户是否赢得奖金:

- (1) 如果用户输入的所有数匹配彩票的确切顺序, 奖金是10 000美金。
- (2) 如果用户输入的所有数匹配彩票的所有数字, 奖金是3000美金。
- (3) 如果用户输入的其中一个数匹配彩票号码中的一个数, 奖金是1000美金。

3.16 (任意字符) 使用Math.random()编写程序, 显示任意的一个大写字母。

*3.17 (游戏:剪刀、石头、布) 编写可以玩最流行的剪刀-石头-布游戏的程序。(剪刀可以剪布, 石头可以砸剪刀, 而布可以包石头。) 程序提示用户随机产生一个数, 这个数为0、1或者2, 分别表示石头、剪刀和布。程序提示用户输入值0、1或者2, 然后显示一个消息, 表明用户和计算机谁赢了游戏, 谁输了游戏, 或是打成平手。下面是运行示例:

```
scissor (0), rock (1), paper (2): 1
The computer is scissor. You are rock. You won
```



```
scissor (0), rock (1), paper (2): 2
The computer is paper. You are paper too. It is a draw
```



*3.18 (使用输入对话框) 使用输入对话框改写程序清单3-8。

3.19 (验证三角形的有效性) 编写程序, 读取三角形的三条边, 并确定输入是否有效。如果任意两条边的和大于第三条边则输入有效。以下是该程序的运行示例:

```
Enter three edges: 1 2.5 1
Can edges 1, 2.5, and 1 form a triangle? false
```



```
Enter three edges: 2.5 2 1
Can edges 2.5, 2, and 1 form a triangle? true
```



*3.20 (科学方面: 风寒温度) 练习题2.17给出计算风寒温度的公式。这个公式适用于温度在华氏-58°到41°之间, 并且风速大于或等于2的情况。编写一个程序, 提示用户输入一个温度值和一个风速值。如果输入值是合法的, 那么显示风寒温度, 否则显示一条消息, 表明温度或风速是不合法数值。

综合题

**3.21 (科学方面: 某天是星期几) 泽勒一致性是由克里斯汀·泽勒开发的用于计算某天是星期几的算法。这个公式是:

$$h = \left(q + \left\lfloor \frac{26(m+1)}{10} \right\rfloor + k + \left\lfloor \frac{k}{4} \right\rfloor + \left\lfloor \frac{j}{4} \right\rfloor + 5j \right) \% 7$$

其中:

- h是一个星期中的每一天 (0为星期六; 1为星期天; 2为星期一; 3为星期二; 4为星期三; 5为星期四; 6为星期五)。
- q是某月的天数。
- m是月份 (3为三月, 4为四月, ..., 12为十二月)。一月和二月分别记为上一年度的13和14月。
- j是世纪数 (即 $\left\lfloor \frac{\text{year}}{100} \right\rfloor$)

• k 是世纪的年数（即 $year \% 100$ ）。

编写程序，提示用户输入年、月和该月的哪一天，然后显示它是一周中的星期几。下面是一些运行示例：

```
Enter year: (e.g., 2008): 2002
Enter month: 1-12: 3
Enter the day of the month: 1-31: 26
Day of the week is Tuesday
```



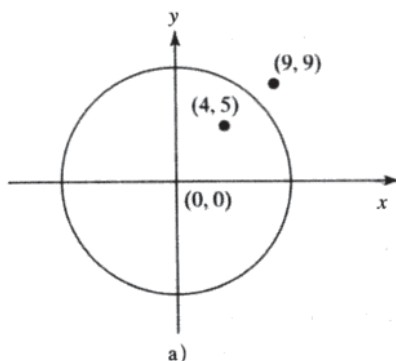
```
Enter year: (e.g., 2008): 2011
Enter month: 1-12: 5
Enter the day of the month: 1-31: 2
Day of the week is monday
```



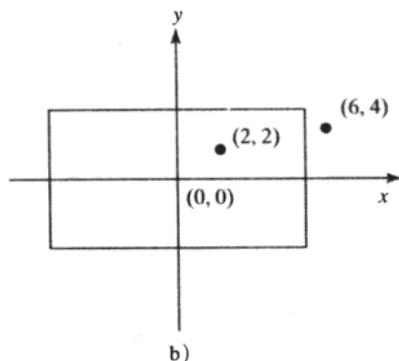
提示 对正数 n 而言 $[n] = (\text{int})n$ 。一月和二月在这个公式里是用13和14表示的。所以需要将由用户输入的月份1转换为13，将用户输入的月份2转换为14，同时将年份改为前一年。

****3.22**（几何方面：点是否在圆内？）编写程序，提示用户输入一个点 (x, y) ，然后检查这个点是否在以原点 $(0, 0)$ 为圆心、半径为10的圆内。例如： $(4, 5)$ 是圆内的一点，而 $(9, 9)$ 是圆外的一点，如图3-7a所示。

提示 如果一个点到 $(0, 0)$ 的距离小于或等于10，那么该点就在圆内，计算距离的公式是 $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ 。以下是两个运行示例。



a)



b)

图3-7 a) 圆内和圆外的点；b) 矩形外和矩形内的点

```
Enter a point with two coordinates: 4 5
Point (4.0, 5.0) is in the circle
```



```
Enter a point with two coordinates: 9 9
Point (9.0, 9.0) is not in the circle
```



****3.23**（几何方面：点是否在矩形内？）编写程序，提示用户输入点 (x, y) ，然后检测该点是否在以原点 $(0, 0)$ 为中心、宽为10、高为5的矩形中。例如： $(2, 2)$ 在矩形内而 $(6, 4)$ 在矩形外，如图3-7b所示。

提示 如果到点 $(0, 0)$ 的水平距离小于等于10/2且到点 $(0, 0)$ 的垂直距离小于等于5/2，该点就在矩形内。这里有两个运行示例。这两个运行示例如下所示。

Enter a point with two coordinates: 2 2
Point (2.0, 2.0) is in the rectangle



Enter a point with two coordinates: 6 4
Point (6.0, 4.0) is not in the rectangle



- **3.24** (游戏: 挑一张牌) 编写程序, 模拟从一副52张的牌中选择一张牌。程序应该显示牌的大小 (Ace、2、3、4、5、6、7、8、9、10、Jack、Queen、King) 以及牌的花色 (Clubs (黑梅花)、Diamond (红方块)、Heart (红心)、Spades (黑桃))。下面是这个程序的运行示例:

The card you picked is Jack of Hearts

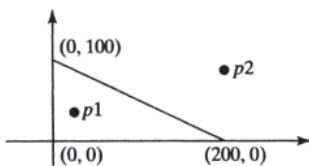


- **3.25** (计算一个三角形的周长) 编写程序, 读取三角形的三条边, 如果输入值合法就计算这个三角形的周长; 否则, 显示这些输入值不合法。如果任意两条边的和大于第三边, 那么输入值都是合法的。
- 3.26** (使用运算符&&、||和^) 编写一个程序, 提示用户输入一个整数值, 然后判定它是否能被5和6整除, 是否能被5或6整除, 以及是否能被5或6整除但是不能同时被它们整除。下面是这个程序的运行示例:

Enter an integer: 10
Is 10 divisible by 5 and 6? false
Is 10 divisible by 5 or 6? true
Is 10 divisible by 5 or 6, but not both? true



- **3.27** (几何方面: 点是否在三角形内?) 假设一个直角三角形放在一个平面上, 如下图所示。直角点在 (0, 0) 处, 其他两个点分别在 (200, 0) 和 (0, 100) 处。编写程序, 提示用户输入一个点的x坐标和y坐标, 然后判定这个点是否在该三角形内。下面是运行示例:



Enter a point's x- and y-coordinates: 100.5 25.5
The point is in the triangle



Enter a point's x- and y-coordinates: 100.5 50.5
The point is not in the triangle



- **3.28** (几何方面: 两个三角形) 编写一个程序, 提示用户输入两个三角形中点的x坐标和y坐标以及它们的宽度和高度, 然后判定第二个三角形是在第一个三角形内, 还是和第一个三角形重叠, 如图3-8所示。

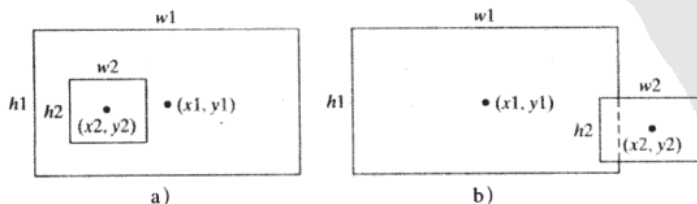


图3-8 a) 一个三角形在另一个三角形里; b) 一个三角形和另一个三角形重叠

下面是运行示例：

```
Enter r1's center x-, y-coordinates, width, and height:
2.5 4 2.5 43 --Enter
Enter r2's center x-, y-coordinates, width, and height:
1.5 5 0.5 3 --Enter
r2 is inside r1
```



```
Enter r1's center x-, y-coordinates, width, and height:
1 2 3 5.5 --Enter
Enter r2's center x-, y-coordinates, width, and height:
3 4 4.5 5 --Enter
r2 overlaps r1
```



```
Enter r1's center x-, y-coordinates, width, and height:
1 2 3 3 --Enter
Enter r2's center x-, y-coordinates, width, and height:
40 45 3 2 --Enter
r2 does not overlap r1
```



****3.29**（几何方面：两个圆）编写程序，提示用户输入两个圆的中心坐标和各自的半径值，然后决定第二个圆是否在第一个圆内，还是和第一个圆重叠，如图3-9所示。

提示 如果两个圆心的距离 $\leq |r1 - r2|$ ，就认为circle2在circle1内；如果两个圆心的距离 $\leq r1 + r2$ ，就认为circle2和circle1重叠。

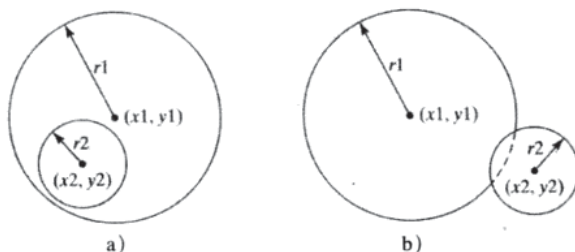


图3-9 a) 一个圆在另一个圆内；b) 一个圆和另一个圆重叠

下面是运行示例：

```
Enter circle1's center x-, y-coordinates, and radius:
0.5 5.1 13 --Enter
Enter circle2's center x-, y-coordinates, and radius:
1.1 7 4.5 --Enter
circle2 is inside circle1
```



```
Enter circle1's center x-, y-coordinates, and radius:
3.4 5.7 5.5 --Enter
Enter circle2's center x-, y-coordinates, and radius:
6.7 3.5 3 --Enter
circle2 overlaps circle1
```



```
Enter circle1's center x-, y-coordinates, and radius:
3.4 5.5 1 --Enter
Enter circle2's center x-, y-coordinates, and radius:
5.5 7.2 1 --Enter
circle2 does not overlap circle1
```

