```
public class SomeThing {
  private int t1;
  private static int t2;

  public SomeThing(int t1, int t2) {
    ...
  }
}
```

a)

```
public class SomeThing {
  private int t1;
  private static int t2;

  public SomeThing(int t1) {
    ...
  }

  public static void setT2(int t2) {
    SomeThing.t2 = t2;
  }
}
```

b)

实例和静态是面向对象程序设计不可或缺的部分。数据域或方法要么是实例的，要么是静态的。不要错误地忽视了静态数据域或方法。常见的设计错误是将本应该声明为静态方法的方法声明为实例方法。例如：用于计算 n 的阶乘的 factorial(int n) 方法应该定义为静态的，因为它不依赖于任何具体实例。

构造方法永远都是实例方法，因为它是用来创建具体实例的。一个静态变量或方法可以从实例方法中调用，但是不能从静态方法中调用实例变量或方法。

13.10.7 继承与聚合

继承和聚合之间的差异，就是 is-a（是一种）和 has-a（具有）之间的关系。例如，苹果是一种水果；因此，可以使用继承来对 Apple 类和 Fruit 类之间的关系进行建模。人具有名字；因此，可以使用聚合来对 Person 类和 Name 类之间的关系建模。

13.10.8 接口和抽象类

接口和抽象类都可以用于为对象指定共同的行为。如何决定是采用接口还是类呢？通常，比较强的 is-a（是一种）关系清晰地描述了父子关系，应该采用类来建模。例如，因为桔子是一种水果，它们的关系就应该采用类的继承关系来建模。弱的 is-a 关系，也称为 is-kind-of（是一类）关系，表明一个对象拥有某种属性。弱的 is-a 关系可以使用接口建模。例如，所有的字符串都是可以比较的，因此 String 类实现了 Comparable 接口。圆或者矩形是一个几何对象，因此 Circle 可以设计为 GeometricObject 的子类。圆有不同的半径，并且可以基于半径进行比较，因此 Circle 可以实现 Comparable 接口。

接口比抽象类更加灵活，因为一个子类只能继承一个父类，但是却可以实现任意个数的接口。然而，接口不能具有具体的方法。可以结合接口和抽象类的优点，创建一个接口，使用一个抽象类来实现它。可以视其方便使用接口或者抽象类。我们将在第 20 章给出这类设计的实例。

复习题

13.35 描述类的设计原则。

关键术语

abstract class（抽象类）
abstract method（抽象方法）
deep copy（深复制）
interface（接口）
marker interface（标记接口）
shallow copy（浅复制）
subinterface（子接口）

本章小结

1. 抽象类和常规类一样，都有数据和方法，但是不能用 new 操作符创建抽象类的实例。
2. 非抽象类中不能包含抽象方法。如果抽象类的子类没有实现所有被继承的父类抽象方法，就必须将该子类也定义为抽象类。
3. 包含抽象方法的类必须是抽象类。但是，抽象类可以不包含抽象的方法。
4. 即使父类是具体的，子类也可以是抽象的。
5. 接口是一种与类相似的结构，只包含常量和抽象方法。接口在许多方面与抽象类很相近，但抽象类除了包含常量和抽象方法外，还可以包含变量和具体方法。
6. 在 Java 中，接口被认为是一种特殊的类。就像常规类一样，每个接口都被编译为独立的字节码文件。
7. 接口 java.lang.Comparable 定义了 compareTo 方法。Java 类库中的许多类都实现了 Comparable。
8. 接口 java.lang.Cloneable 是一个标记接口。实现 Cloneable 接口的类的对象是可克隆的。
9. 个类仅能继承一个父类，但一个类却可以实现一个或多个接口。
10. 一个接口可以继承一个或多个接口。

测试题

回答本章位于 www.cs.armstrong.edu/liang/intro10e/quiz.html 中的测试题。

编程练习题

13.2 ～ 13.3 节

**13.1 （三角形类）设计一个扩展自抽象类 GeometricObject 的新的 Triangle 类。绘制 Triangle 类和 GeometricObject 类的 UML 图并实现 Triangle 类。编写一个测试程序，提示用户输入三角形的三条边、一种颜色以及一个表明该三角形是否填充的布尔值。程序应该根据用户的输入，使用这些边以及颜色和是否填充的信息，创建一个 Triangle 对象。程序应该显示面积、周长、颜色以及真或者假来表明是否被填充。

*13.2 （打乱 ArrayList）编写以下方法，打乱 ArrayList 里面保存的数字。

```
public static void shuffle(ArrayList<Number> list)
```

*13.3 （排序 ArrayList）编写以下方法，对 ArrayList 里面保存的数字进行排序。

```
public static void sort(ArrayList<Number> list)
```

**13.4 （显示日历）重写程序清单 6-12 中的 PrintCalendar 类，使用 Calendar 和 GregorianCalendar 类显示一个给定月份的日历。你的程序从命令行得到月份和年份的输入，例如：

```
java Exercise13_04 5 2016
```

这个会显示如图 13-9 中的日历。

也可以不输入年份来运行程序。这种情况下，年份就是当前年份。如果不指定月份和年份来运行程序，那么就是指当前月份。

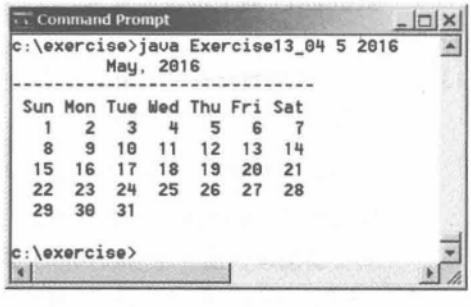


图 13-9 程序显示 2016 年五月的日历

13.4 ～ 13.8 节

*13.5 （将 GeometricObject 类变成可比较的）修改 GeometricObject 类以实现 Comparable 接口，并且在 GeometricObject 类中定义一个静态的求两个

GeometricObject 对象中较大者的 max 方法。画出 UML 图并实现这个新的 GeometricObject 类。编写一个测试程序，使用 max 方法求两个圆中的较大者和两个矩形中的较大者。

*13.6 （ComparableCircle 类）创建名为 ComparableCircle 的类，它继承自 Circle 类，并实现 Comparable 接口。画出 UML 图并实现 compareTo 方法，使其根据面积比较两个圆。编写一个测试程序求出 ComparableCircle 对象的两个实例中的较大者。

*13.7 （可着色接口 Colorable）设计一个名为 Colorable 的接口，其中有名为 howToColor() 的 void 方法。可着色对象的每个类必须实现 Colorable 接口。设计一个名为 Square 的类，继承自 GeometricObject 类并实现 Colorable 接口。实现 howToColor 方法，显示一个消息 Color all four sides（给所有的四条边着色）。

画出包含 Colorable、Square 和 GeometricObject 的 UML 图。编写一个测试程序，创建有五个 GeometricObject 对象的数组。对于数组中的每个对象而言，如果对象是可着色的，那就调用 howToColor 方法。

*13.8 （修改 MyStack 类）重写程序清单 11-10 中的 MyStack 类，执行 list 域的深度复制。

*13.9 （将 Circle 类改成可比较的）改写程序清单 13-2 中的 Circle 类，它继承自 GeometricObject 类并实现 Comparable 接口。覆盖 Object 类中的 equals 方法。当两个 Circle 对象半径相等时，则这两个 Circle 对象是相同的。画出包括 Circle、GeometricObject 和 Comparable 的 UML 图。

*13.10 （将 Rectangle 类变成可比较的）改写程序清单 13-3 的 Rectangle 类，它继承自 GeometricObject 类并实现 Comparable 接口。覆盖 Object 类中的 equals 方法。当两个 Rectangle 对象面积相同时，则这两个对象是相同的。画出包括 Rectangle、GeometricObject 和 Comparable 的 UML 图。

*13.11 （八边形类 Octagon）编写一个名为 Octagon 的类，它继承自 GeometricObject 类并实现 Comparable 和 Cloneable 接口。假设八边形八条边的边长都相等。它的面积可以使用下面的公式计算：

$$面积 = (2+4/\sqrt{2}) \times 边长 \times 边长$$

画出包括 Octagon、GeometricObject、Comparable 和 Cloneable 的 UML 图。编写一个测试程序，创建一个边长值为 5 的 Octagon 对象，然后显示它的面积和周长。使用 clone 方法创建一个新对象，并使用 compareTo 方法比较这两个对象。

*13.12 （求几何对象的面积之和）编写一个方法，求数组中所有几何对象的面积之和。方法签名如下：

```
public static double sumArea(GeometricObject[] a)
```

编写测试程序，创建四个对象（两个圆和两个矩形）的数组，然后使用 sumArea 方法求它们的总面积。

*13.13 （使得 Course 类可复制）重写程序清单 10-6 中的 Course 类，增加一个 clone 方法，执行 students 域上的深度复制。

13.9 节

*13.14 （演示封装的好处）使用新的分子分母的内部表达改写 13.13 节中的 Rational 类。创建有两个整数的数组，如下所示：

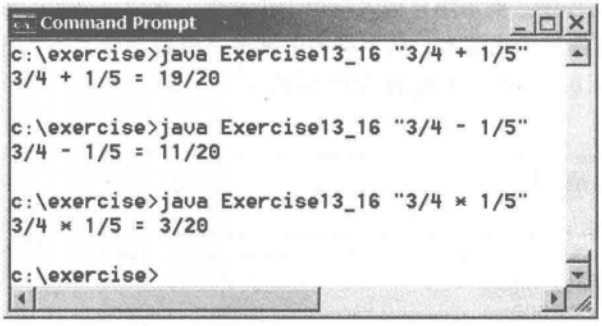
```
private long[] r = new long[2];
```

使用 r[0] 表示分子，使用 r[1] 表示分母。在 Rational 类中的方法签名没有改变，因此，无须重新编译，前一个 Rational 类的客户端应用程序可以继续使用这个新的 Rational 类。

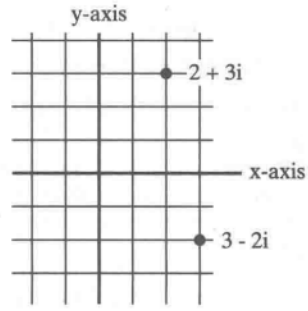
*13.15 （在 Rational 类中使用 BigInteger）使用 BigInteger 表示分子和分母，重新设计和实现 13.13 节中的 Rational 类。

*13.16 （创建一个有理数的计算器）编写一个类似于程序清单 7-9 的程序。这里不使用整数，而是使用有理数，如图 13-10a 所示。

需要使用在 10.10.3 节中介绍的 String 类中的 split 方法来获取分子字符串和分母字符串，并使用 Integer.parseInt 方法将字符串转换为整数。



a）程序从命令行得到三个参数（操作数 1、操作符、操作数 2），显示该表达式以及算数运算的结果



b）复数可以解释为一个平面上的点

图 13-10

*13.17 （数学：Complex 类）一个复数是一个形式为 $a+bi$ 的数，这里的 a 和 b 都是实数，i 是 $\sqrt{-1}$ 的平方根。数字 a 和 b 分别称为复数的实部和虚部。可以使用下面的公式完成复数的加、减、乘、除：

$$a + bi + c + di = (a + c) + (b + d)i$$

$$a + bi - (c + di) = (a - c) + (b - d)i$$

$$(a + bi)*(c + di) = (ac - bd) + (bc + ad)i$$

$$(a + bi)/(c + di) = (ac + bd)/(c^2 + d^2) + (bc - ad)i/(c^2 + d^2)$$

还可以使用下面的公式得到复数的绝对值：

$$|a+bi| = \sqrt{a^2 + b^2}$$

（复数可以解释为一个平面上的点，将（a, b）值作为该点的坐标。复数的绝对值是该点到原点的距离，如图 13-10b 所示。）

设计一个名为 Complex 的复数来表示复数以及完成复数运算的 add、substract、multiply、divide 和 abs 方法，并且覆盖 toString 方法以返回一个表示复数的字符串。方法 toString 返回字符串 a+bi。如果 b 是 0，那么它只返回 a。Complex 类应该也实现 Cloneable 接口。

提供三个构造方法 Complex(a,b)、Complex(a) 和 Complex()。Complex() 创建数字 0 的 Complex 对象，而 Complex(a) 创建一个 b 为 0 的 Complex 对象。还提供 getRealPart() 和 getImaginaryPart() 方法以分别返回复数的实部和虚部。

编写一个测试程序，提示用户输入两个复数，然后显示它们做加、减、乘、除之后的结果。下面是一个运行示例：

```
Enter the first complex number:  3.5 5.5 Enter
Enter the second complex number:  -3.5 1 Enter
(3.5 + 5.5i) + (-3.5 + 1.0i) = 0.0 + 6.5i
(3.5 + 5.5i) - (-3.5 + 1.0i) = 7.0 + 4.5i
(3.5 + 5.5i) * (-3.5 + 1.0i) = -17.75 + -13.75i
(3.5 + 5.5i) / (-3.5 + 1.0i) = -0.5094 + -1.7i
|(3.5 + 5.5i)| = 6.519202405202649
```

13.18 （使用 Rational 类）编写程序，使用 Rational 类计算下面的求和数列：

$$\frac{1}{2}+\frac{2}{3}+\frac{3}{4}+\cdots+\frac{98}{99}+\frac{99}{100}$$

你将会发现输出是不正确的，因为整数溢出（太大了）。为了解决这个问题，参见编程练习题 13.15。

13.19 （*将十进制数转化为分数*）编写一个程序，提示用户输入一个十进制数，然后以分数的形式显示该数字。提示：将十进制数以字符串的形式读入，从字符串中抽取其整数部分和小数部分，然后运用编程练习题 13.15 中使用 `BigInteger` 实现的 `Rational` 类，来获得该十进制数的有理数。这里是一些运行示例：

```
Enter a decimal number: 3.25 Enter
The fraction number is 13/4
```

```
Enter a decimal number: -0.45452 Enter
The fraction number is -11363/25000
```

13.20 （*数学：求解二元方程*）重写编程练习题 3.1，如果行列式小于 0，则使用编程练习题 13.17 中的 `Complex` 类来得到虚根。这里是一些运行示例：

```
Enter a, b, c: 1 3 1 Enter
The roots are -0.381966 and -2.61803
```

```
Enter a, b, c: 1 2 1 Enter
The root is -1
```

```
Enter a, b, c: 1 2 3 Enter
The roots are -1.0 + 1.4142i and -1.0 + -1.4142i
```

13.21 （*代数：顶点式方程*）抛物线方程可以表达为标准形式（$y = ax^2 + bx + c$）或者顶点式（$y = a(x-h)^2 + k$）。编写一个程序，提示用户输入标准形式下的整数 a、b 和 c 值，显示顶点式下面的 h 和 k 值。这里是一些运行示例：

```
Enter a, b, c: 1 3 1 Enter
h is -3/2 k is -5/4
```

```
Enter a, b, c: 2 3 4 Enter
h is -3/4 k is 23/8
```