附录 D

UML 概览

本附录简要介绍了本书用到的统一建模语言（Unified Modeling Language，UML）的特性。UML 提供了一套规范的表示方法，可以很好地阐释面向对象系统的设计思想。UML 并不复杂，但不能低估这些特性所包含的丰富内涵。若要快速地了解 UML 的特性，可以阅读 *UML Distilled*（由 Fowler 和 Scott 在 2003 年编写）一书。若要深入了解，则可以阅读 *The Unified Modeling Language User Guide*（由 Booch、Rumbaugh 和 Jacobsen 在 1999 年编写）一书。通过学习标准的命名方法与表示方法，我们可以更好地在设计层面上进行交流，从而提高团队的生产效率。

类

图 D.1 运用了 UML 的某些特性描述类。

如下是对该类图的说明。

- 一个矩形表示一个类，类名居中。图 D.1 中显示了三个类：`Firework`、`Rocket` 与 `simulation.RocketSim`。
- UML 并不要求在图中描述所有与该元素相关的内容，例如类的所有方法或包中包含的完整内容。

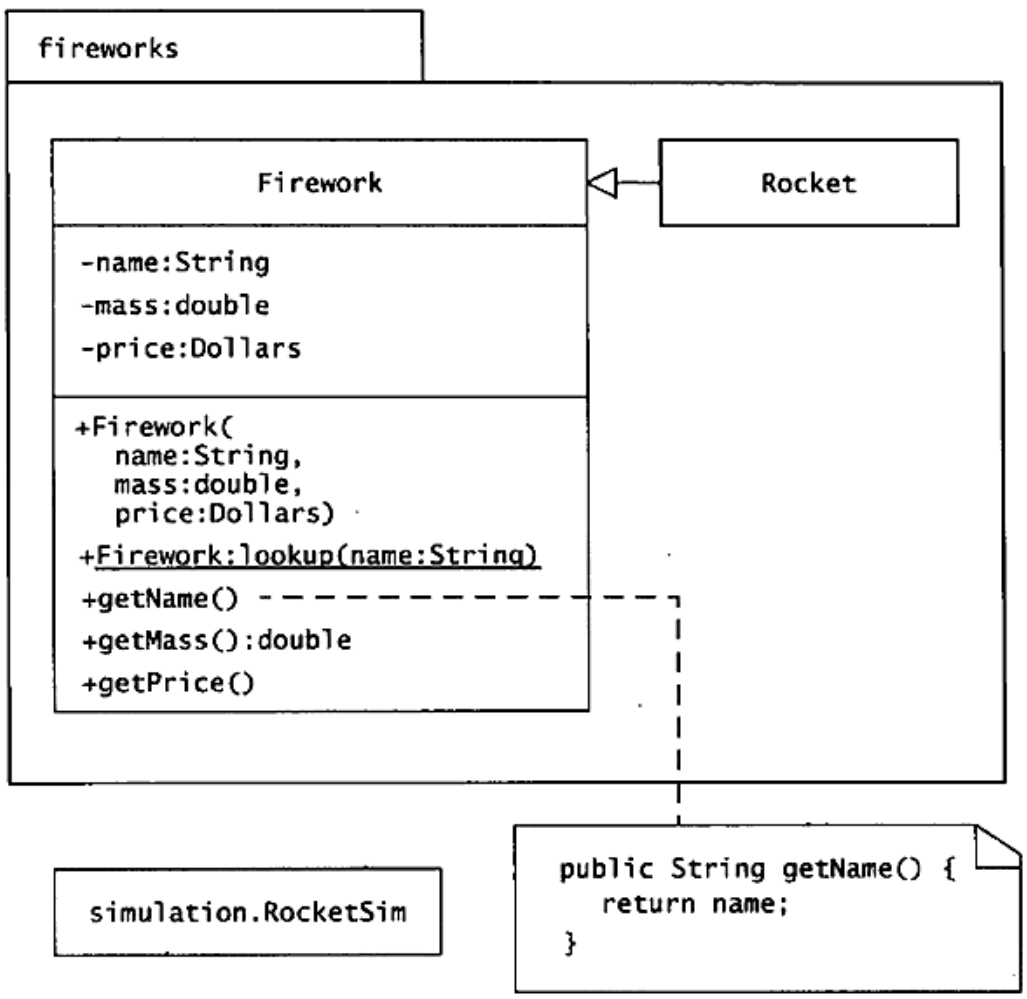

图 D.1　包含了 Firework 与 Rocket 类的 Fireworks 包

- 包也使用矩形表示，但包名是左对齐的，其下有更大的框，显示了这个包的类与其他类型。例如图 D.1 中所示的 `Firework` 包中的类。
- 当包图下显示一个类时，说明该类所在的命名空间。例如图 D.1 说明 `RocketSim` 类位于 `Simulation` 包中。
- 类的实例变量显示在类名下的矩形框中。`Firework` 类中包含 `name`、`price`、`mass` 等实例变量。实例变量名称后跟一个冒号，以及该实例变量的类型。
- 实例变量或方法前的减号（-），说明该变量或方法的可见性是 private；加号（+）说明可见性为 public；#号说明可见性为 protected。
- 类的方法显示在类名下的第二个矩形框中。
- 如果方法具有参数，通常应该将这些参数显示出来，请参见图中的 `lookup()`方法。
- 方法签名中的变量通常被组织为这样的形式：变量名称后加冒号，再加变量的类型。如果该变量的类型能说明变量的功能，也可以省略或缩写变量名。
- 在一个实例变量或方法的名称下加下画线，表示该变量或方法是静态的，如图 D.1 中的

loopup()方法，它就是静态方法。

- 可以在一个卷角的矩形框中加入一段注解。其中的文字可以是注释、约束或代码，并使用虚线将它与图中的其他元素连起来。注释可以出现在任何一个 UML 图中。

类关系

图 D.2 显示了用于对类关系进行建模的 UML 特性。

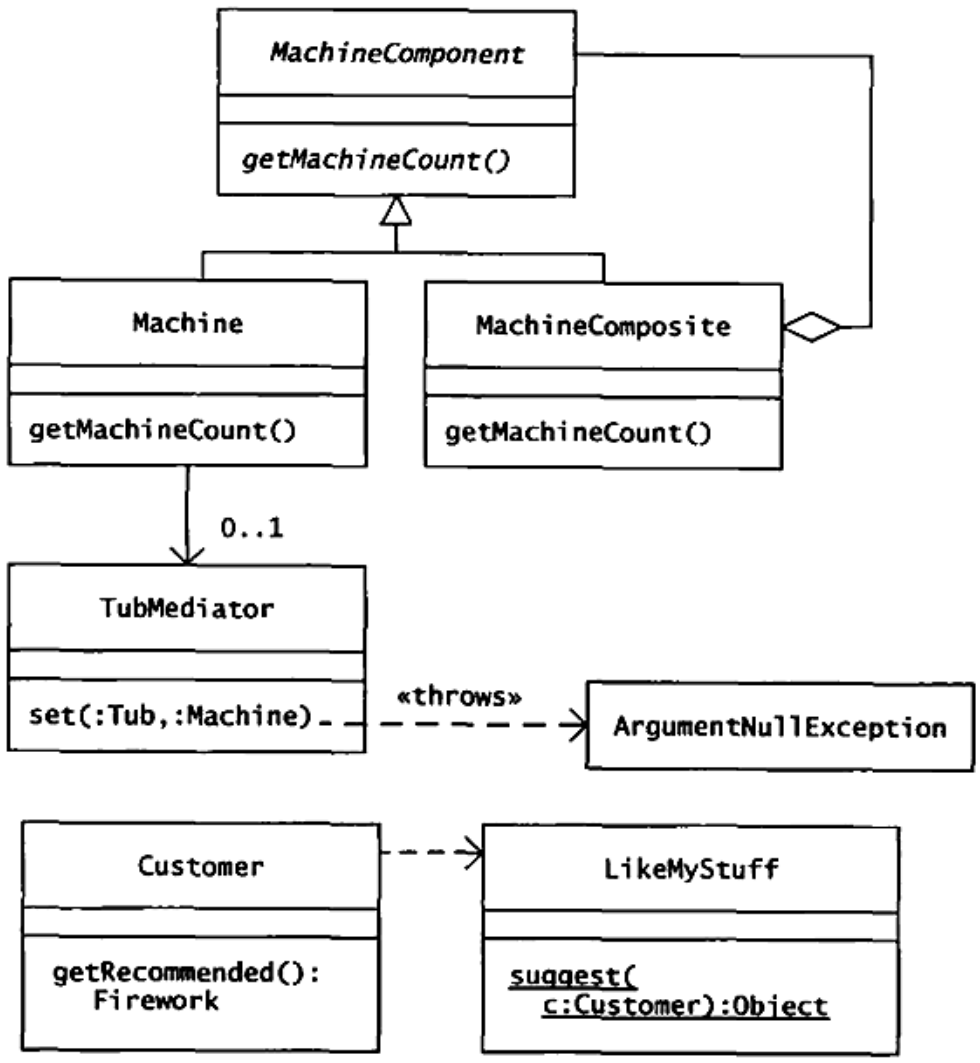

图 D.2　包含了 Machine 对象或其他组合对象的 MachineComposite 对象。Customer 类依赖于 LikeMyStuff 类，但 Customer 类不需要实例化它。

下面是类关系表示方法的说明。

- 显示的类名或方法名若为斜体，则表示该类或方法是抽象的。方法名下有下画线则表明它是静态的。
- 使用一个闭合的空心箭头指向类的超类。
- 两个类之间的连线表示二者的实例存在关联关系。通常，类图中的连线表示一个类中实例变量引用了另一个类。例如，Machine 类使用实例变量引用了 TubMediator 对象。
- 连线一端的菱形表示一个类的实例包含另一个类的实例的集合。
- 一个开放箭头表示某种引用关系，通常指一个类引用了另一个类，且被指向的类并没有反向引用前一个类。
- 多重指示符，例如 0..1，用于指示对象之间可能出现的连接数。星号（*）表示一个类有零个到多个实例与对应类的实例存在关联关系。
- 如果一个方法会抛出异常，可以从该方法引出一个带开放箭头的虚线执行这个异常类，并使用<<throws>>构造型标记这个箭头。
- 可以用一个带箭头的虚线表示类之间不存在对象引用的依赖关系。例如，图中的 Customer 类使用了 LikeMyStuff 的静态方法 Suggest()，而不是通过对象引用。

接口

图 D.3 显示了接口的基本特性。

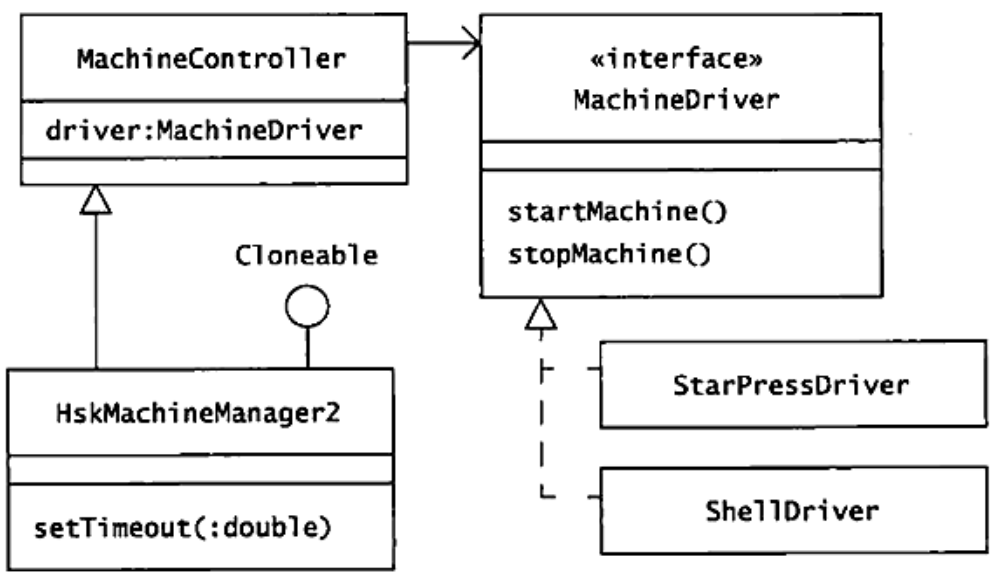

图 D.3　你可以用«interface»构造型棒棒糖图示来表示接口

如下是对接口的说明。

- 如图 D.3 所示，可以用一个包含了<<interface>>文字与接口名的矩形框来表示接口。可以使用带闭合空心箭头的虚线表示类实现了接口。
- 还可以使用一条线加一个圆形（棒棒糖形状），并在旁边标注接口名称来表示类实现了接口。
- 在 Java 语言中，接口与它的方法都是抽象的。奇怪的是，与抽象类和抽象方法不同，接口与它的方法并非斜体。

对象

对象图表示了类的特定实例，如图 D.4 所示。

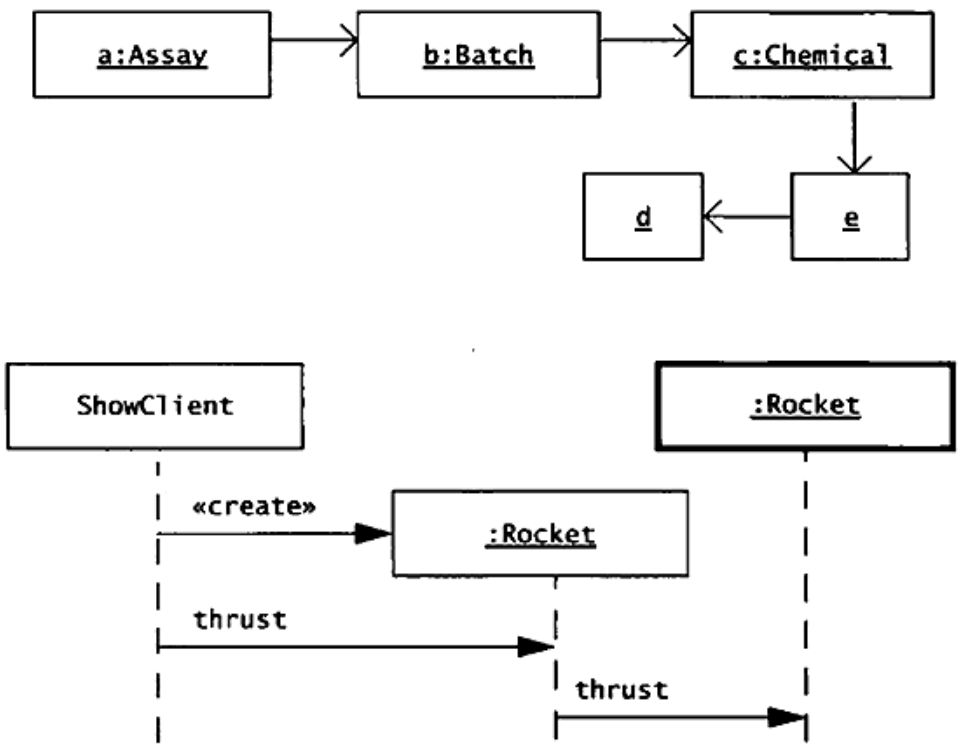

图 D.4　描述对象通常会指定对象的名称和/或对象的类型。时序图描述了连续的方法调用

如下是对对象图的说明。

- 可以通过冒号分隔的对象名与类型来描述一个对象，也可以只显示对象的名称，或者只显示冒号和类型。无论哪种情况，对象的名称与类型都要加下画线。

- 在对象之间用一条线表示一个对象引用了另一个对象。可以使用开放箭头来重点标注引用的方向。
- 可以使用图 D.4 下方的时序图表示对象发送消息给另一个对象的顺序。消息的顺序自上而下，虚线表示对象的时间区间。
- 使用<<create>>构造型表示一个对象创建了另一个对象。图 D.4 说明了 ShowClient 类创建了一个本地的 Rocket 对象。
- 可以通过为对象的矩形边框加粗表明该对象是在另一个线程、进程或另一台计算机上运行的活跃对象。图 D.4 显示了一个本地 Rocket 对象将 thrust()调用请求转发给运行在服务器上的 Rocket 对象。

状态

图 D.5 显示了 UML 的状态图。

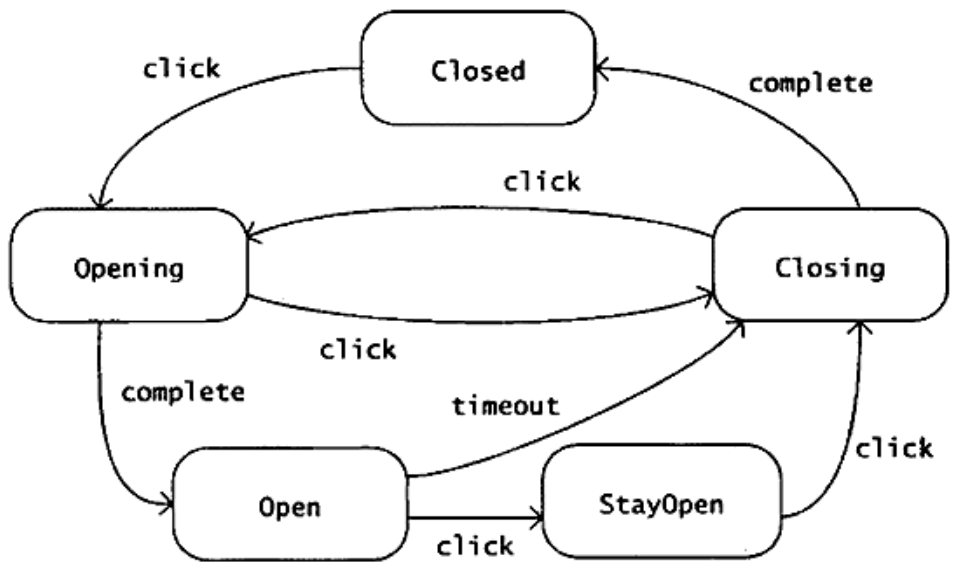

图 D.5　表示状态之间迁移的状态图

如下是对状态图的说明。

- 用带圆角的矩形框表示一种状态。
- 使用开放箭头表示状态的迁移。
- 状态图无须直接映射到类图或对象图，当然，也可以对状态图进行这样的转换，如第 22 章的图 22.3 所示。