

第十七章 MyEclipse UML 建模

第十七章 MyEclipse UML 建模	1
17.1 介绍	1
17.1.1 UML概念及常见建模工具	1
17.1.2 MyEclipse的UML工具	2
17.2 系统需求	4
17.3 创建UML模型仓库	4
17.4 创建及修改UML 图	5
17.4.1 创建UML 图形	5
17.4.2 类图和常见修改功能	6
17.4.3 正向工程 - UML 类图生成Java代码	12
17.4.4 反向工程 - Java 代码生成UML 类图	13
17.4.5 绘图工具	17
17.4.6 用例图(Use Case Diagram)	18
17.4.7 活动图 (Activity Diagram)	18
17.4.8 序列图 (Sequence Diagram)	20
17.5 常见问题	21
17.6 小结	22
17.7 参考资料	22

17.1 介绍

17.1.1 UML 概念及常见建模工具

UML 是什么？在回答这个问题之前，请您思考这样一种现象。路边有一棵很大很粗的银杏树，好了，现在让几个从事不同职业的人，或者是操不同方言的人，来给别人介绍这棵银杏树，你猜会出现什么样的情况？考古者会说：哦，这是一棵古树，很有科研价值，对于研究这一带的地志迁移很有帮助，假设用无缝取样法看看年轮，甚至能了解几百年来的气候变迁。农民朋友会说：这是一棵长势喜人的银杏树，不过银杏是雌雄异株，嫁接上对应性别的枝条后，能够收成很多银杏。生物学家会说：银杏（Ginkgo Liloba L.），植物界，裸子植物门，银杏纲，银杏目，银杏科，银杏属，银杏种，俗称白果，公孙树。……当然，如果是外国人来介绍银杏，则另是一番情况。最后，请各位把见到的银杏树画出来，那估计又是各有千秋，甚至如果碰到印象派画家，画出来的图让你都认不出这还是棵银杏。因为，人们表达自己的形式是千奇百怪的，在软件行业，也是如此。在 UML 之前，各式各样的图表被用来描述软件系统，这带来了知识交流过程中的阻碍。UML 的提出，就是为了部分的解决软件系统的描述问题。好了，现在我们就用专业点的属于来介绍 UML：统一建模语言（Unified Modeling Language, UML），是一种直观化、明确化、构建和文档化软件系统产物的通用可视化建模语言。它捕捉了被构建系统的有关决策和理解，用来理解、设计、浏览、配置、维护以及控制系统的信息，可以与所有的开发方法、生命阶段、应用领域和媒介一同

使用。简言之，就是一组标准化的可扩展的可视化建模语言，结果表现为图形，注意它不是一门编程语言，只用来描述系统，不能用来开发和实现系统功能。1997 年，OMG 组织(Object Management Group 对象管理组织) 发布了统一建模语言。UML 的目标之一就是为开发团队提供标准通用的设计语言来开发和构建计算机应用。UML 提出了一套 IT 专业人员期待多年的统一的标准建模符号。通过使用 UML，这些人员能够阅读和交流系统架构和设计规划--就像建筑工人多年来所使用的建筑设计图一样。截至到目前为止，UML 已经推出了 2.0 版本，在企业中获得了广泛的应用。当然，它也有一些缺陷，例如一些新的软件结构，如 SOA 等，就无法用 UML 描述。另外，还有个很重要的地方就是 UML 并没有规定图的颜色，但是也没规定 UML 图不能带彩色，因此，UML 的图主要在于形状，颜色本质上是黑白的。

常见的 UML 建模工具比较多。用的最多的当属 Rational Rose (现在被 IBM 收购)，它的特点主要是老牌，出来的图看着比较正规，不过和工具的整合不够紧密，是商业的收费软件。另外还有 Borland Together，它的最强大的功能当属动态的代码和图形同步功能，即从代码生成图，并且代码改动或者图改动后，都会同步到对应的地方，这个几乎是无人能及，在生成顺序图等功能上也是很强大的，它是用 Java 语言开发的，有 Borland Together for Eclipse 这样的版本，便于在 Eclipse 下使用。还有一款就是 Sybase 出品的 Power Designer，它的主要特色就是数据库建模和 UML 建模合二为一，可以从数据库生成模型，也可以从模型生成数据库代码，是公司数据库管理员必备软件，可惜也是商业收费的。微软 Office 的 Visio 也能进行 UML 建模，只可惜，出来的图不甚标准，也无法从图生成原型代码。开源的 UML 建模工具中，用的比较广泛的当属 ArgoUML，实际上 MyEclipse 的也是基于它二次开发了一下而已。另外，Netbeans 6 也提供了 UML 建模功能。Eclipse 也有对应的 UML 插件，然而多数都是收费软件（这当属 Eclipse 社区被大公司控制的必然结局），读者可以自行搜索，或者使用参考资料中介绍的一款 EclipseUML，以及一款国产的 UML 建模软件。当然还有很多的免费 UML 建模工具可以使用，通过 Google 搜索即可。

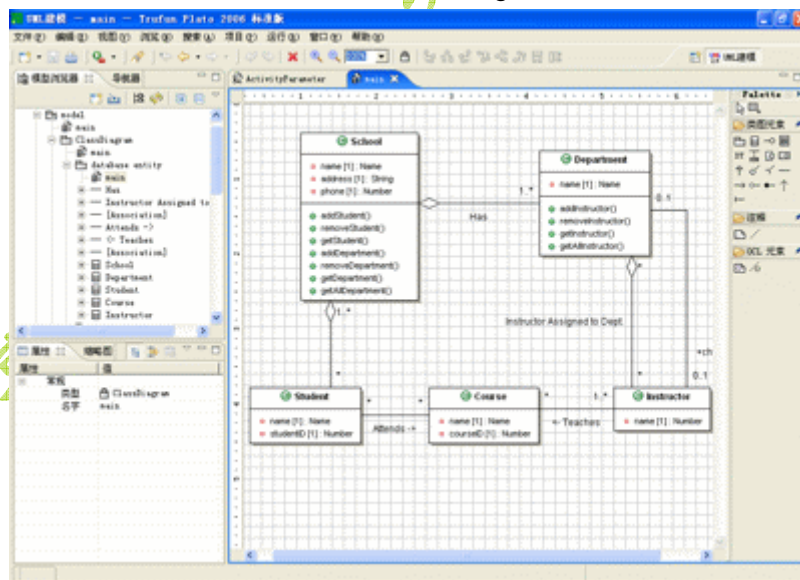


图 17.1 典型 UML 建模工具界面

17.1.2 MyEclipse 的 UML 工具

MyEclipse 的 UML 建模工具实际上是基于 ArgoUML 开发的，并加入了反向工程和正向工程的能力，可以从代码生成 UML 图或者从图生成代码。MyEclipse UML 为开发人员提供了下面的 UML 功能：

- UML 图: 用例图, 类图, 序列图, 协作图, 状态图, 活动图, 部属图
- 集成的图片编辑器, 大纲视图和属性视图
- UML 透视图
- 不限位置的绘制工具
- 通过热区直接编辑接点和连接内容
- 图片保存在 UML Model Repository (UML 模型仓库) 文件中 (例如 cardemo. umr)
- 包含 UML 模型文件的项目类型以及文件存储位置不受限制
- 从模型生成 Java 代码
- 导出图形为图片文件, 支持多种格式: GIF、PNG、PS、EPS 和 SVG
- 从任何 MyEclipse J2EE 或者 Eclipse Java 项目生成类图
 - 批量反向工程处理任何项目或者源代码目录, Java 包或者单独类的组合
 - 从 Eclipse 的 Package Explorer 透视图拖拽任何 Java 类或者接口添加进 UML 类图上
 - 自定义反向工程设置
- 从 Outline 视图将任何 UML 元素拖放到任何兼容的 UML 图中
- 对类图进行自动布局

另外, MyEclipse 还提供了专门的 UML 透视图, 来便于进行 UML 的设计工作。MyEclipse UML 透视图提供了精心调整过的的界面组织来方便进行建模活动。默认视图包括 UML Diagram Editor (UML 图形编辑器), Outline (大纲) 视图, 已经自定义的 Properties (属性) 视图。工具栏上的操作按钮进行了调整和增加, 这样依赖使创建新图的过程快速和方便的多。图 17.2 列出了对各部分进行了标示的 UML 透视图。

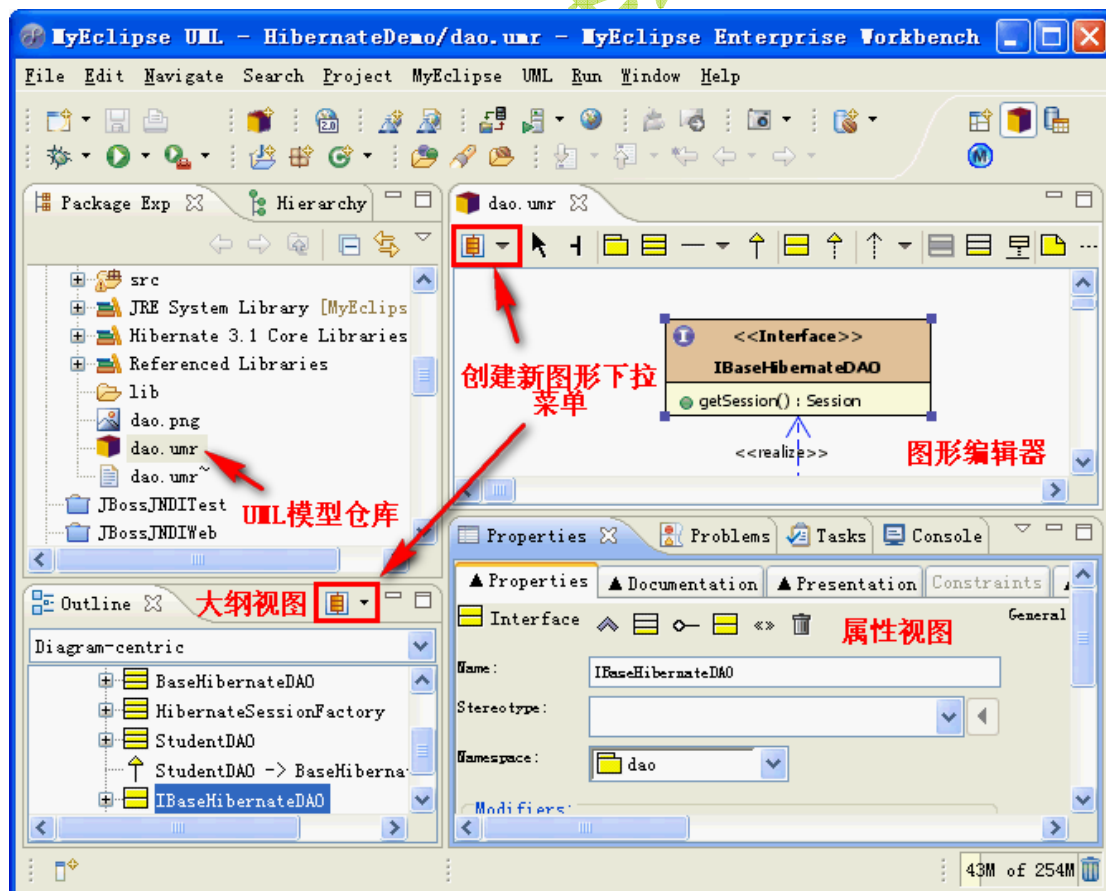



图 17.2 MyEclipse UML 透视图

17.2 系统需求

由于 MyEclipse 集成 ArgoUML 实现 UML 的开发功能，而 ArgoUML 当前版本则需要 JDK/JRE1.5 或者更高版本，因此必须安装这些版本的 JDK/JRE 之后运行 MyEclipse 才可使用。当然 MyEclipse All IN One 版本本身已经自带了 JRE 1.5，无需额外配置。对操作系统的支持上，则致辞 Windows Vista, XP, 2003 和 2000 以及 Linux，而苹果的 Mac 系统则不支持，这是因为 SWT 本身的一个向 SWT 中插入 Swing 组件的 Bug 所导致的。本书中的例子和截图均在 Windows XP 中文版 SP2 下运行和测试。

17.3 创建 UML 模型仓库

MyEclipse UML 使用一种特殊的文件格式来保存所有的 UML 图形以及模型元素到单一文件中，即 UML Model Repository (UMR)。UMR 文件的后缀是 ".umr"，并且在 Package Explorer 视图中以图标的方式进行区分显示。在 Eclipse 项目中可以创建任意多个 UMR 文件。其实 UMR 文件的真实存储格式是 ZIP，可以使用 7Zip, WinZIP, WinRAR 等软件打开，就跟 JAR 文件的存储格式差不多。

使用 MyEclipse UML 创建图形的第一步工作，就是用新建 UML 向导来创建一个 UML 模型仓库文件 (UMR)。为了方便起见，我们事先已经创建了一个名为 *UMLTest* 的 Java 项目。具体步骤如下：

1. 启动新建 UMR 向导。可以通过使用菜单 **File > New > UML Model Repository**，或者点击工具栏上的按钮，也可以来启动创建 UML 模型文件的向导。
2. 在 **Enter or select the parent folder**（输入或者选择父目录）一栏中选中要保存文件的项目，在 **UML Model Repository**（UML 模型仓库）一栏中，输入文件名，在 **UML Root Model Name**（UML 根模型名称）一栏中，输入想要的名字。此过程可以参考图 17.3 进行相关的设置。

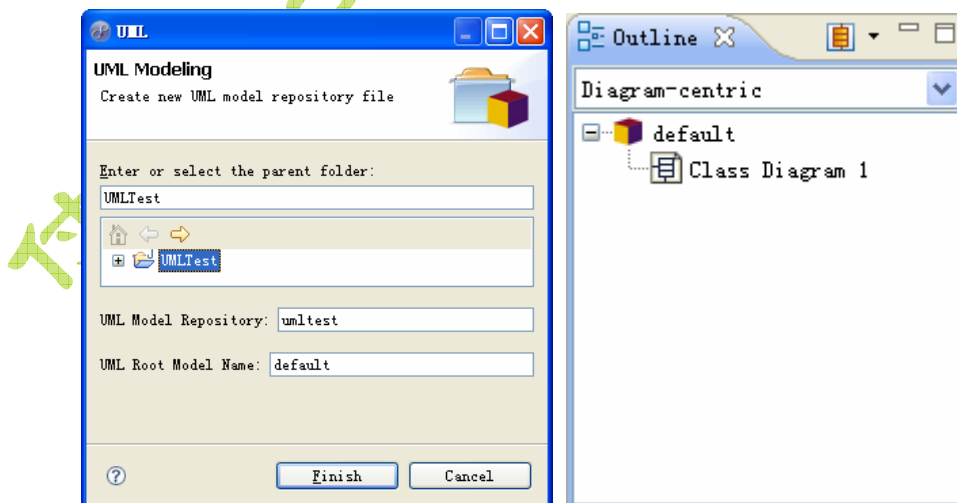


图 17.3 新建 UMR 向导及空模型文件

3. 选择 **Finish** 按钮，关闭向导对话框，并在刚指定的位置下创建 UML 文件，稍后 MyEclipse 会启动 UML Diagram Editor (UML 图形编辑器)。默认情况下新建的

UMR 文件只有一个空的类图，参考图 17.3 右侧大纲视图。

4. 切换 MyEclipse 透视图为 **MyEclipse UML**，这样便于修改和创建 UML 图。

17.4 创建及修改 UML 图

本节我们会简单介绍常见的几种 UML 图形的概念以及创建方式。由于本书重点在于 MyEclipse 的工具使用上，所以 UML 的概念我们并没有打算做 100% 的详细介绍，图的种类也选择了常用的几种，其它的读者可以自行查找资料进行学习。这样省出的篇幅，用于加入后续章节的其它内容，给读者提供更多方面的参考资料。

17.4.1 创建 UML 图形

单个 UML 模型文件可以包含任意数目的和任何类型的 UML 图形。向文件中添加图形，需要在编辑器中打开 UML 文件，然后从编辑器工具栏上左数第一个按钮，从下拉列表中选择要建的图的类型即可，参考图 17.4。另外还可以从 **Outline**（大纲）视图下点击视图中的类似按钮，也可以创建图形，参考图 17.2。支持的 UML 图形列表则参考表 17.1。由于这一特征，如果你想和别人分享做好的 UML 模型，把这个文件发给对方即可（当然前提是他安装有 MyEclipse 或者 Argo UML 来查看它）。

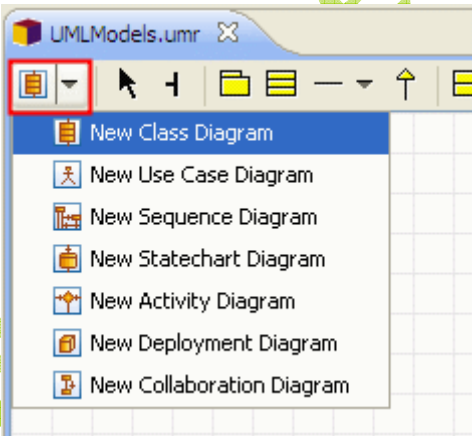


图 17.4 使用编辑器器工具栏来创建图形

	用例图
	类图
	序列图
	协作图
	状态图
	活动图
	部署图

表 17.1 可以创建的 UML 图类型列表

当一个模型文件中加入了多个图之后，如何进行切换到想编辑的图呢？有两个地方可以进行图之间的切换：第一个地方是在大纲视图，双击需要修改的图的名字即可；另一个地方就是在打开的图的编辑器的工具栏上，有一个下拉的列表可以切换图。此过程请参考图

17.5。

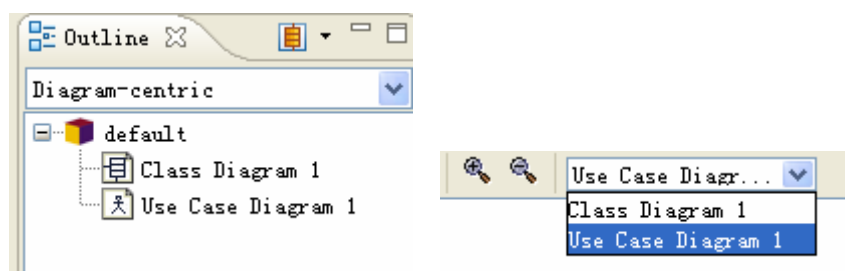


图 17.5 在大纲和编辑器中切换多个图

向编辑器的图中加入 UML 元素可以通过单击编辑器工具栏上的按钮来完成，另外双击工具栏上的组件按钮可进入 **Mass Add**（批量添加）模式，此时每点击一次画布就加入一个元素，要退出此模式可以点击编辑器工具栏上的光标状 **Select**（选择）按钮，即可。另外，带有下拉列表选择的功能，例如矩形绘制，也可以进入批量添加模式。在后面几节的内容中，笔者将会对各种类图的概念，图形特点和创建方式逐一进行介绍。

如果要将绘制的图形导出为图片文件，也比较容易，在编辑器的画布空白处右键点击，然后选择菜单 **Export as image**，接着在弹出的对话框中输入要保存的图片文件名以及存储路径即可，支持这些导出格式：PNG，GIF，PostScript，Encapsulated PostScript（封装过的 PostScript）和 SVG（Scalable Vector Graphics，可伸缩矢量图形）格式。这样导出的图片可以用作图软件来处理，或者插入到文档或者网页中进行显示。

17.4.2 类图和常见修改功能

实际建模时候，大概类图是最经常用到的一种图形了。类图表示不同的实体（人、事物和数据）如何彼此相关；换句话说，它显示了系统的静态结构。类图可用于表示逻辑类，逻辑类通常就是业务人员所谈及的事物种类——订单，商品，顾客，经理，管理员，账户等等。类图也可以表现面向对象中提到的接口。在 Java 语言中，几乎所有的类，都可以用类图来表示，之所以用了几乎这个不确定词，是因为 UML 本身有其局限性，例如下面的例子就无法准确的用 UML 表示出来（模拟人和眼睛的关系）：

```
public class Person {
    class Eye {}
}
```

，这样的结构在 Java 中随处可见，是内部类，但是却无法用 UML 很好的表示出来。类似的其它内容还有 Web 服务，JSP 等严格说也可以作为类描述的 Java 模块，但是这些都无法很好的用 UML 表示，不过，按照 UML 的规定，开发工具可以自行扩充模型后来对这些内容进行建模。

类图上使用包含三个部分的矩形来描述，如图 17.6 所示。最上面的部分显示类的名称，中间部分包含类的属性（成员变量）列表，最下面的部分包含类的操作（或者说“方法”、函数）列表。当在一个类图上画一个类元素时，你必须要有顶端的区域，下面的二个区域是可选择的（当图描述仅仅用于显示系统的高层架构时，下面的两个区域是不必要的）。

位于类图最顶部的，是类名。类名上可以显示图标，也可以不显示，这是可选的功能，同理所有的 UML 图都可以拥有定制的图标，这是可以扩展到内容。

类的属性栏（中部区域）在分隔线上列出类的属性列表。属性节是可选择的。每个属性的定义使用如下格式：

name : attribute type

即 名称 : 属性 类型。例如图中的变量定义 `username:String`，是类型在后，属性名在前，等价于 `String username` 这样的 Java 代码。如果此模型和实际的编程语言无关，作为业务

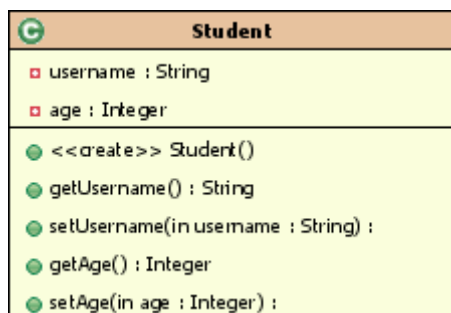


图 17.6 单个类图

系统的描述图，是可以出现中文类型的描述，例如：单价:美元这样的变量定义；如果要用此图生成代码，则只能使用标准的变量定义和类型。个别的建模工具，例如 Together，支持给属性名加入别名方便阅读和理解。

类操作列表位于类图长方形的第三个（最底部）区域中，它也是可选的。和属性一样，类的操作以列表显示。操作使用下列格式定义：

name(parameter list) : type of value returned

即：操作名 (参数列表)：返回值类型。构造器也包含在操作列表中。如图 17.6 中所示的，`<<create>>Student()` 表明这是一个构造器（构造器没有返回值）；而 `getUsername():String` 则定义了一个返回值为 `String` 类型的方法 `getUsername()`；另外 `setUsername(in username : String)` 定义了一个设置用户名的方法，输入参数是 `String` 类型的，参数名为 `username`，`in` 这个标记用来指示参数是输入值（当然不是必须的，不写 `in` 也可以）。

除了类之外，常见的类图还包括接口。接口如图 17.7 中左上的 `IMoveable` 所示，定义了一个行走和停止的行为。实际上接口中也可以规定属性，但是 MyEclipse UML 在这里不支持这样的格式。如果一个类需要实现一个接口，则可以通过带有闭合的单向箭头的虚线进行连接，即表示实现定义。例如图中的 `Animal` 实现了 `IMoveable` 接口，用 Java 到代码来表示就是：`class Animal implements IMoveable`。

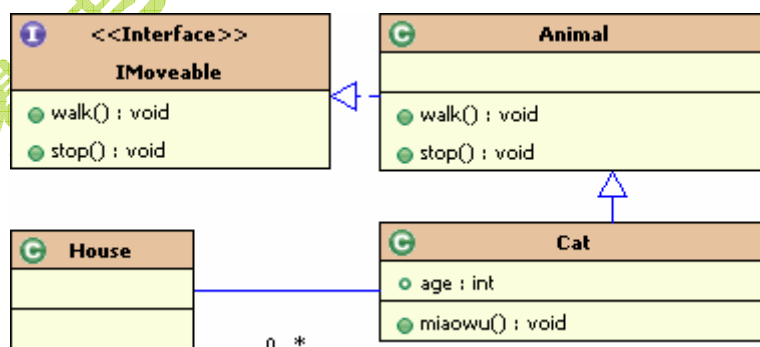


图 17.7 接口和类继承，实现，关联

类之间可以有继承关系，通过带有闭合单向箭头的实线进行连接。例如图 17.7 中，类 `Cat` 继承自 `Animal`。另外，类和类之间还可以有关联关系等，例如一个家庭对应多只猫。双向关联关系已实现的方式连接，并在线上显示关联类型。例如图中的 `House` 关联着多只 `Cat`。表 17.2 列出了双向关联中的可能只所表示的含义的描述信息。如果是单向关联，则用

不封闭的单向箭头来表示：→，从一个类连接到另一个类。此外还可以有聚合等关系，我们就不多做介绍了，有兴趣的读者可以阅读参考资料中的 IBM 网站中的文章获得更多信息。

表示	含义
0..1	0 个或 1 个
1	只能 1 个
0..*	0 个或多个
*	0 个或多个
1..*	1 个或我个
3	只能 3 个
0..5	0 到 5 个
5..15	5 到 15 个

表 17.2 双向关联中可能的多重值描述

另外一个很重要的概念就是**可见性**。在面向对象的设计中，存在属性及操作可见性的记号。UML 识别四种类型的可见性：**public**，**protected**，**private** 及 **package**（和 Java 中的内容对应的非常好）。UML 规范并不要求属性及操作可见性必须显示在类图上（其实 MyEclipse 生成的图使用不同的和 Eclipse 中一致的图标来表示可见性了，但是别的工具一般是按照 UML 的规范来使用记号来标识的），但是它要求为每个属性及操作定义可见性。为了在类图上的显示可见性，放置可见性标志于属性或操作的名字之前。虽然 UML 指定四种可见性类型，但是实际的编程语言可能增加额外的可见性，或者不支持 UML 定义的可见性。表 17.3 显示了 UML 支持的可见性类型的不同标志。可见性的例子如：**+username:String** 这样的属性定义。

标志	可见性类型
+	Public
#	Protected
-	Private
~	Package

表 17.3 可见性列表

在 Java 的开发中，经常会遇到包（Package）这个概念。UML 中也支持类似的概念：**软件包**，用来将多个相关的类或者接口组织在一块。软件包使建模者能够组织模型分类器到名字空间中，这有些象文件系统中的文件夹。把一个系统分为多个软件包使系统变成容易理解，尤其是在每个软件包都表现系统的一个特定部分时。软件包在 UML 中以文件夹的方式进行显示，并在文件夹头部显示包名，文件夹中包含类或者接口。图 17.8 中列出了一个包含两个类的软件包 *lang*。

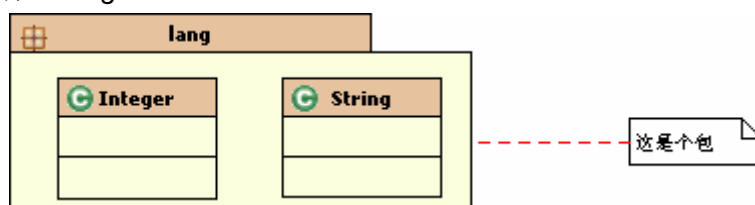


图 17.8 软件包及注释

另外，和常见的编程语言支持注释一样，UML 也支持注释（Comment），如图 17.8 右侧即是一个注释元素以及注释关联线。注释可以加到任何对象上。

现在关于 UML 的概念已经做了这么多的介绍，接下来就来看看在 MyEclipse 中如何绘制类图。首先自然是打开新建的 Class 图，然后可以看到 UML 编辑器的工具栏按钮列表：

图标	英文提示	说明
	Select	恢复为准备选中状态
	Broom	扫帚(点击后向各个方向拖动可以使图形对齐)
	New Package	新建包
	New Class	新建类
	New Association	新建关联等关系(下拉选择)
	New Generalization	新建泛化类(继承)
	New Interface	新建接口
	New Realization	新建实现类
	New Dependency	新建依赖类(下拉选择)
	New Attribute	新建属性
	New Operation	新建操作
	New Association Class	新建关联类
	New Comment	新建注释
	New Comment Link	新建注释连接
	New Rectangle	新建方形(下拉选择)
	Zoom In	放大
	Zoom Out	缩小

表 17.4 类图编辑器工具栏

在这些工具栏中，最常用的当属类图编辑功能了。点击 UML 编辑器工具栏上的 按钮即可创建一个新的类。双击类图的三个方框的空白处：类名，属性列表和操作列表处，即可快速修改类名或者加入新的属性和操作定义。如果双击已经存在的属性或者操作定义，则可以变为文本输入框，进入修改模式来修改。基本上所有的 UML 元素都支持此种修改方式，MyEclipse 的术语叫 **Text hot-zone (文字热区)**。文字热区是节点或者连接点的中间的可编辑区域，例如关联类的连线中间，通过双击可以激活。激活后，热区内显示方形的可编辑输入框。输入框内可以直接输入文本，也支持删除、复制、粘贴、剪切带操作。要提交文字热区内地修改使之生效，可以点击文字热区之外的任何地方即可，随后编辑区消失并代之以显示原来被编辑的 UML 内容。关于类图编辑和热区，可以参考图 17.9。

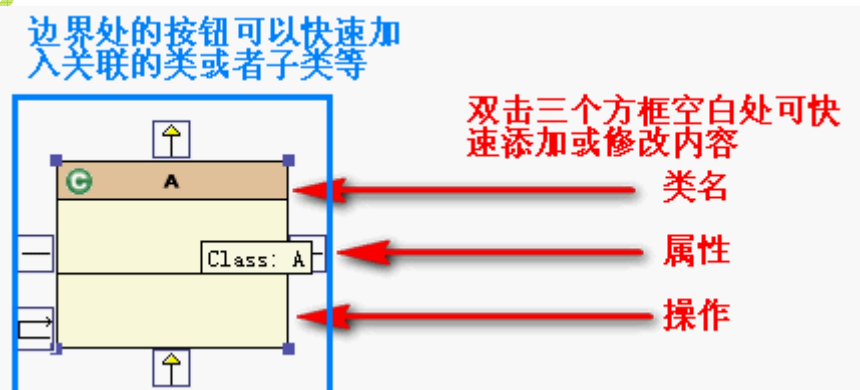




图 17.9 快速编辑类，连接区和编辑区和 Text hot-zone

另外一种热区， **Connection hot-zone**，即连接热区，它位于哪里呢？它位于类图的四周，鼠标移过时可以看到多出的几个按钮，一般提供连接到节点的一些关联类，自关联，子类，依赖等等，可以点击按钮到另一个节点来完成连接，也可以只单击按钮自动生成一个对应的关联类。

要修改图形的大小，需要把鼠标放到图形的四个角上的某一个，出现黑色的小方块后，拖动即可调节其大小（不过图有个最小值，不能调节的比最小值还小）。要移动元素，可以用鼠标画框的方式选中需要移动的元素，然后再其中的任意一个元素上点击并拖放即可移动图形的位置。

最后一个和编辑有关的就是 **Properties**（属性）视图，当选中不同的元素时，会显示对应的可以修改的属性值，如图 17.10 所示。例如选中类 A 中的属性后，此时 **Properties** 视图如图 17.10 所示。**Properties** 视图中也可以编辑对应的值，并且以多页标签的方式来显示所有可以修改，添加或者删除的内容（相对比下，在编辑器的画布上能改的属性则非常有限）。**Properties** 标签下的工具栏，可以处理一些操作例如转到上级（Go Up），新建属性（New Attribute），新建数据类型（New Datatype），新建版型（New Stereotype），以及删除模型（Delete from Model）等等。在 Name 输入框中可以修改属性名，Type 可以选择数据类型，Initial Value 中可以输入初始值，Visibility 中可以选择可见度，包括 Protected（受保护的），Public（公开的）和 Private（私有的）。Changeability 中则可以选择是否为可以修改的值。Modifiers 中则可以修改修饰符。实际上每种元素都有它自己特定的属性，但大部分都是用不到的属性，所以我们就不打算在这里做完整的介绍了。

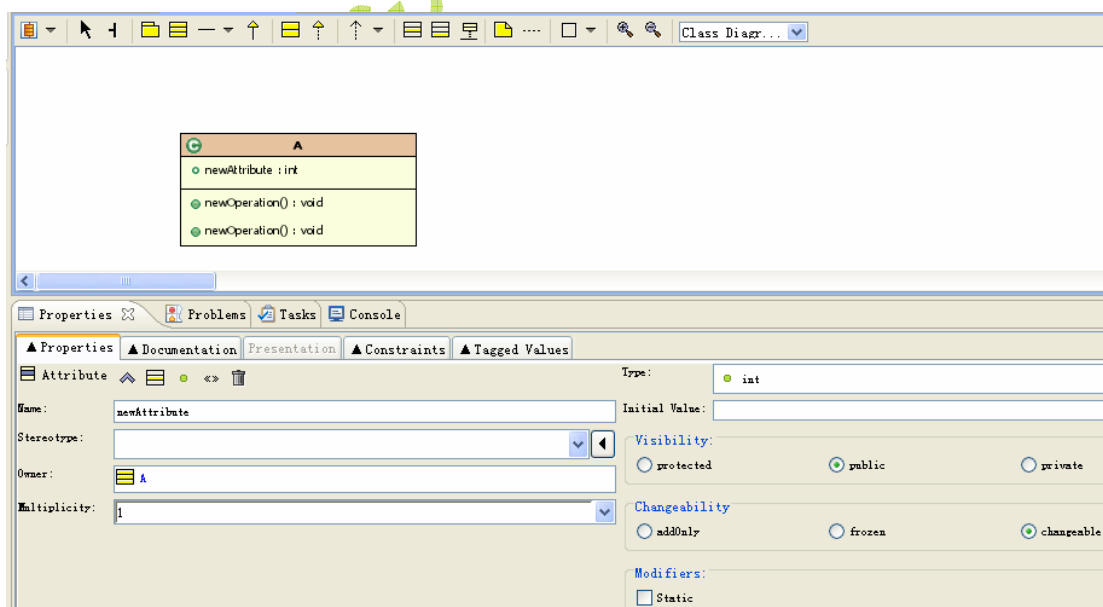


图 17.10 UML Properties 视图

Properties 视图中另外一个相对固定的标签页就是 **Documentation**（文档），可以用来

给每个 UML 的元素撰写一些说明文档（实际上这些内容是和 JavaDoc 中的标记所对应的，因为 MyEclipse 或者 ArgoUML 毕竟是 Java 语言开发的嘛）。在这一页，Author 框输入作者信息，Version 框输入版本，Since 框输入什么时候开始有，Deprecated 则指示是否为废弃的类或者属性等信息，See 则输入参考信息，Documentation 则输入详细的注释信息。

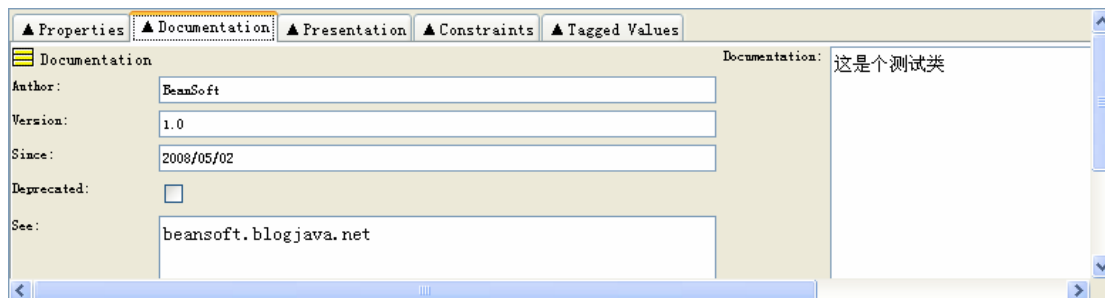


图 17.11 Documentation 标签

当选中图形的最外层时，例如直接选中类图本身，此时就可以修改 Presentation 标签下和图形显示有关的属性了。Display（显示）右侧的两个复选框分别可以指定是否显示属性（Attributes）和操作（Operations）列表。Bounds（边界）则指定图形的坐标和大小，坐标的格式是：x, y, width, height，即：横坐标，纵坐标，宽度，高度。需要说明下在计算机绘图中，坐标的起点一般都是左上角，也即（0，0）这个点。Body Fill Color 指定了图形的主体填充色彩。Header Fill Color 则指定了头部填充色彩（类图中位于最上侧，即类名显示处）。Line 则指定了线条（一般是边框等）的显示颜色。如图 17.12 所示。

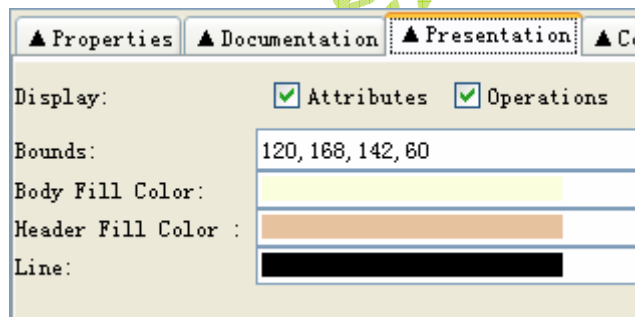


图 17.12 Presentation 标签

最后两个标签分别是 Constrains 和 Tagged Values。Constrains 主要可以给元素添加，删除和修改一些限制信息，一般情况下使用不到。Tagged Values 则列出了做了标记的值列表，例如当按照图 17.11 中设置了文档后，类的 Tagged Values 显示的内容列表如图 17.13 所示。在这个标签的工具栏上可以删除一些标签，还可以双击 Tag（标记）和 Value（取值）进行相关值的编辑。

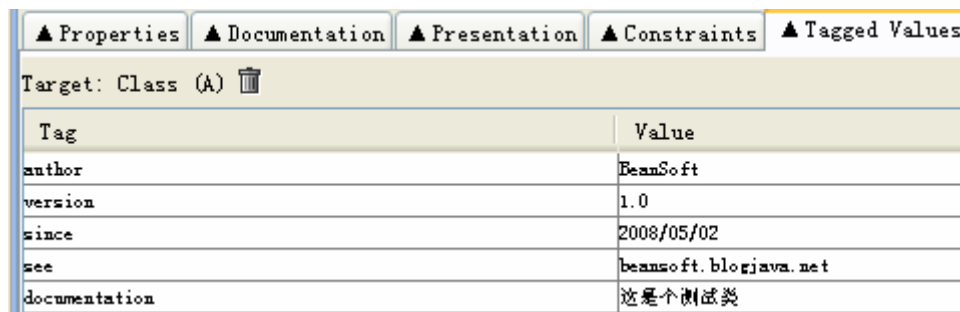


图 17.13 Tagged Values 标签

好了，关于类图的介绍基本上到此为止。现在需要大家做的是练习，开发出如图 17.14 所示的 UML 类图，这个图包括了类，接口以及包和关联关系等。可以参考配套视频，然后自己做一下练习。

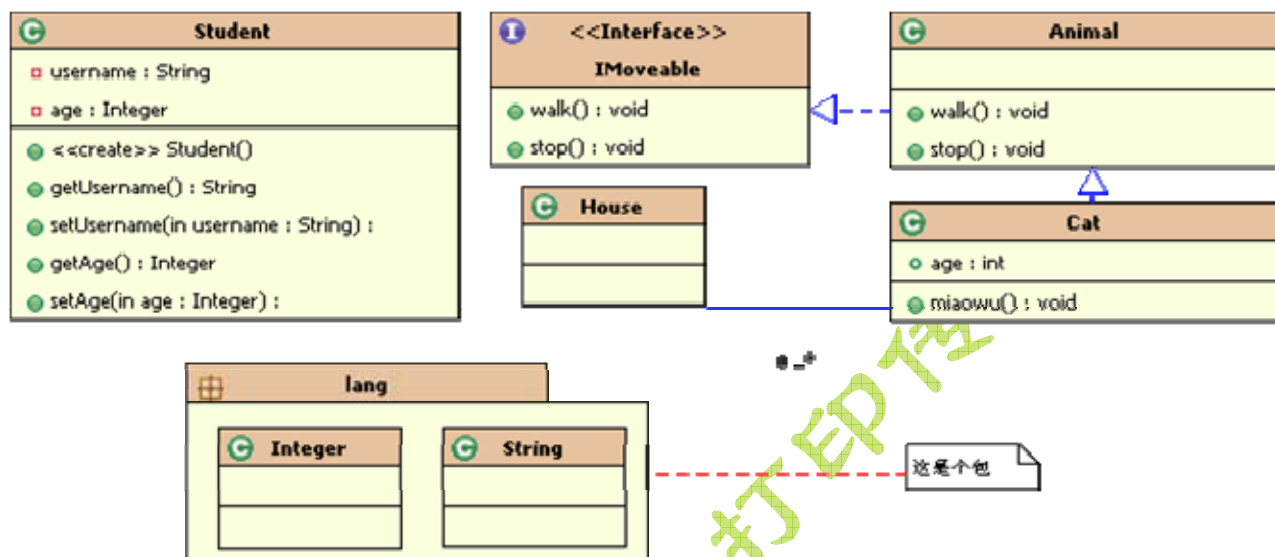


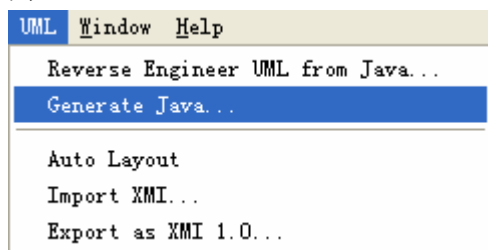
图 17.14 UML 类图练习目标

17.4.3 正向工程 - UML 类图生成 Java 代码

在实际的项目开发中，如果设计出来的图不能变成 Java 代码，那的确是让 UML 的功能大打折扣，尤其是详细到每个类的属性和操作，包名这样的详细设计，如果过后还需要一点点的把对应的代码写出来，的确让人头痛。所幸的是，MyEclipse 的 UML，或者说 ArgoUML 以及大多数的专业一点的 UML 设计器，都提供正向工程——从 UML 类图生成特定语言，例如 Java，C++ 等的源代码，说点题外话就是这当中做的最好的当属 Borland Together。相比较来说，MyEclipse 的这个生成功能，有时候会出现错误，导致无法完整的产生代码。

MyEclipse 支持直接从 UML 类图中的类定义生成 Java 代码。操作步骤如下：

1. 打开包含类图的 UML 仓库文件（UMR），这些类必须按照 Java 的规范进行设计才可生成 Java 代码；
2. 选择菜单 **UML > Generate Java...**，来启动正向工程（Forward-Engineering）向导，如图 17.15 所示。

图 17.15 选择菜单 **Generate Java...**

随后，将会弹出如图 17.16 所示的正向工程向导对话框。按照下列步骤进行操作即可（也可参照图中的箭头提示）：

1. 点击对话框中的 **Java Output Folder** (Java 输出目录) 右侧的 **Browse** 按钮, 在弹出的对话框中选择任何 Java 或者 J2EE 项目的源代码目录 (一般名为 `src`), 然后点击 **OK** 按钮关闭对话框。被选中的目录将存放从类图所最终生成的 Java 代码。
2. 在 **Generate Java source for selected UML Java entities** (生成选中的 UML Java 实体的 Java 源代码) 下面的类来列表中, 选中一个或者多个需要翻译为 Java 代码的类。
3. 点击 **Finish** 按钮来关闭对话框, 然后进行代码生成处理过程。

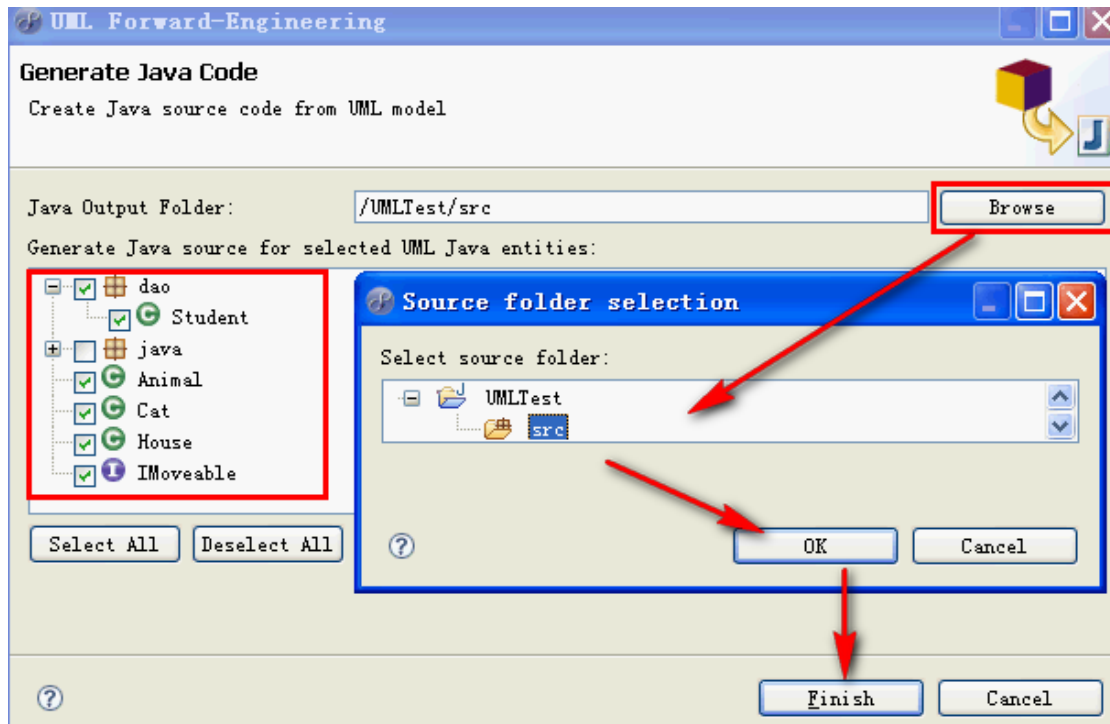


图 17.16 UML 正向工程向导

在稍后可能会出现一些警告和提示的对话框。一般包括: 无返回类型的操作定义没有写入 `void` 类型, 例如: `setAge(in age:Integer)` 应该加入 `void` 修改为: `setAge(in age:Integer):void`; 一些非法的关联选项 (虽然画图的时候可以随便来画, 但是生成的时候, 务必确保这些关联有实际的意义)。这种情况下, 对应的 Java 代码就不能生成, 只有修正了这些问题才能如期产生代码。还有的时候是 MyEclipse 本身的 Bug, 导致生成的类文件是空白的。总之此功能不是很完备。

在生成过程结束后, 就可以在指定的源代码目录中看到生成的类文件的 Java 源码了。

17.4.4 反向工程 - Java 代码生成 UML 类图

在实际的项目开发中, 就笔者所接触过的情况来说, 经常会出现项目的末期, 需要交文档的时候, 大家纷纷把自己本来没做的 UML 设计 (本来按照规则是应该在初期进行的) 补上来, 这时候, 从代码生成 UML 类图的功能就有用了。当然, 也有一种情况是, 虽然早期进行了 UML 设计, 然而毕竟计划不如变化, 到了项目后期, 大多数类的代码早就和设计的时候有了出入, 这时候, 都可以用到从代码生成 UML 类图的功能。

MyEclipse 的 Java 代码到 UML 反向工程的过程有两种不同的方式可以实现, 包括批量处理模式和拖放模式, 下面我们对这两种模式一一进行介绍。

批量处理模式：该模式下可以把任意 Java 项目中源代码目录中的任何选中的类、接口等定义转换成 UML 的类图（注意不支持从 JSP，HTML 之类的文件生成 UML 定义）。让我们按照下列步骤进行操作：

1. 打开原来的测试项目 *HibernateDemo*，然后在项目 *UMLTest* 中新建一个 UML 模型文件 *HibernateDemo.umr*，并双击并打开这个 UMR 文件。当然也可以选择以前创建的任何 Java 或者 J2EE 项目，只要项目中包含源代码即可。
2. 选择菜单 **UML > Reverse Engineer UML from Java...** 来启动 MyEclipse UML 反向工程向导。随后即弹出 UML 反向工程向导对话框，如图 17.17 所示。

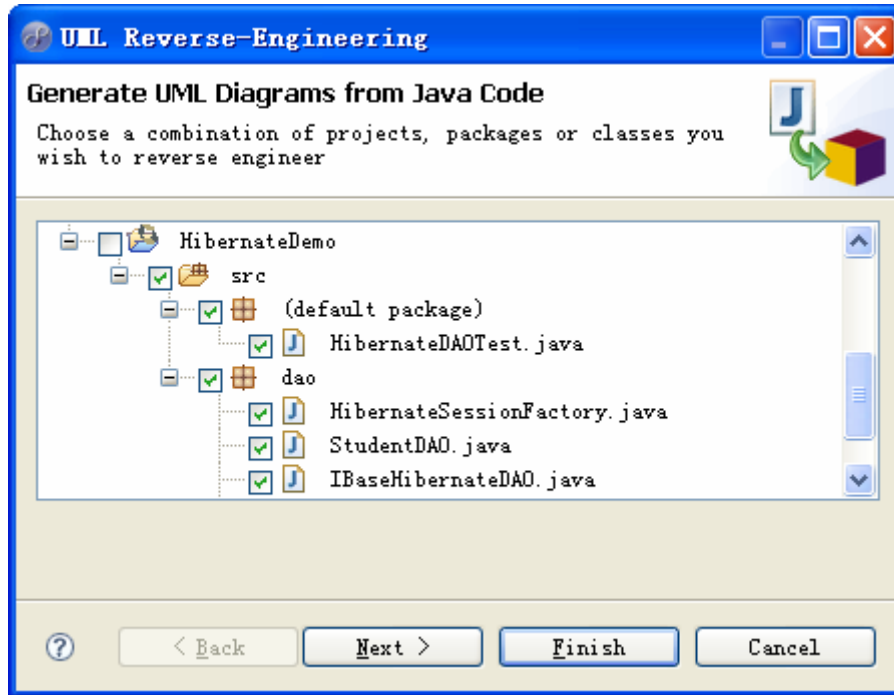


图 17.17 UML 反向工程向导第一页

3. 在向导对话框中选中需要反向工程处理的 Java 包或者类，然后点击 **Next** 按钮进入下一页，如图 17.18 所示。

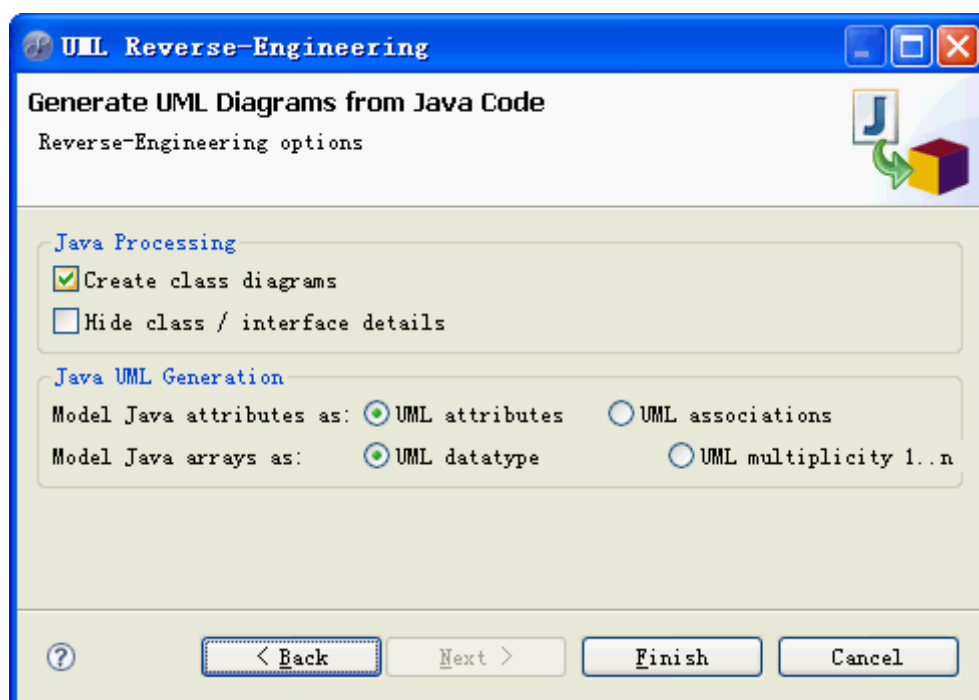


图 17.18 UML 反向工程类图生成设置

4. 在图 17.18 中进行反向工程的选项设置，一般情况下保持默认设置即可。这些设置包括 Java Process（Java 处理）下的选项：Create class diagrams（生成类图）；Hide class / interface details（隐藏类和接口的详细内容）。以及另一个选项栏目 Java UML Generation（Java UML 生成）：Model Java attribute as（将 Java 属性建模为）UML attributes（UML 属性）或者 UML associations（UML 关联）；Model Java arrays as（将 Java 数组建模为）UML datatype（UML 数据类型）或者 UML multiplicity 1..n（UML 一对多）。
5. 点击 Finish 按钮来开始反向过程处理过程。

稍等片刻后，即可看到反向过程的执行结果汇总对话框，并让我们可以选中生成的一个或者多个类图，如图 17.19 所示。检查 Outline 视图中的 UML 图列表，可以看到反向工程是基本上每个包都生成了一个类图，而不是像大家想象的那样都把类放到了一张图上。例如在这个例子中就生成了两个新类图：_classes 和 dao_classes。这样，双击这两个类文件就可以看到生成的类文件了，非常方便。基本上还没碰到过什么大问题。稍后将 UML 导出为图片，就可以在项目文档中自由的引用了。

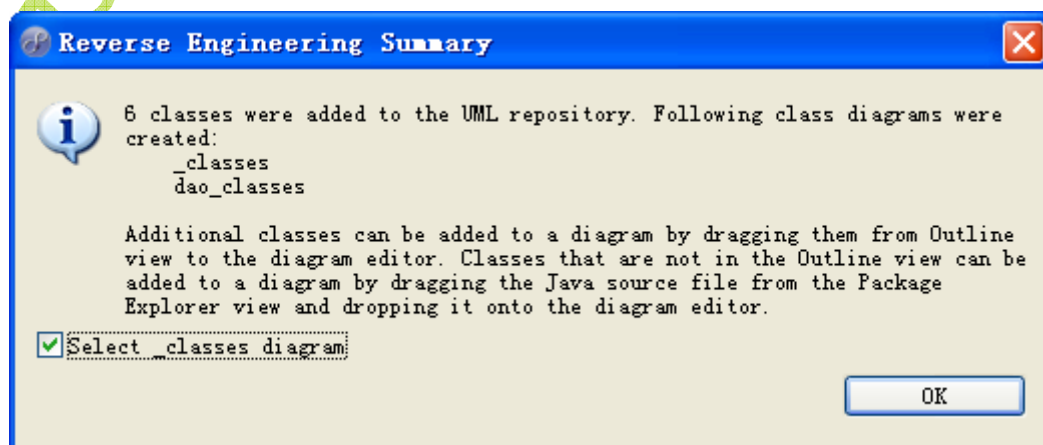


图 17.19 反向工程汇总对话框

接下来要介绍的，是反向工程的另一个模式：**拖放模式**。这个模式适合于快速小批量的从 Java 代码中所定义的和接口生成对应的 UML 类图，可以无缝的将任何 Java 类和接口添加到任一个 UML 模型的类图中。操作步骤如下：

1. 从 **Package Explorer** 视图中，选中一个或者多个类或者接口（需要提示大家的是，除了源代码包里面的类可以选中，**Library** 里面的类也可以被选中）；
2. 鼠标单击并拖动选中的类和接口（不放开鼠标左键移动光标）放到当前打开的 UML 图形编辑器中的类图，如图 17.20 所示；
3. 此时将会弹出如图 17.18 所示的 UML 反向工程类图生成设置信息，一般取默认值即可，或者按照图 17.18 后所附带的说明进行定制；
4. 点击 **Finish** 按钮即可启动生成类图向导。

当向导结束后，即可看到在类图上生成了对应的类定义，以及其相互关系。如图 17.21 所示。

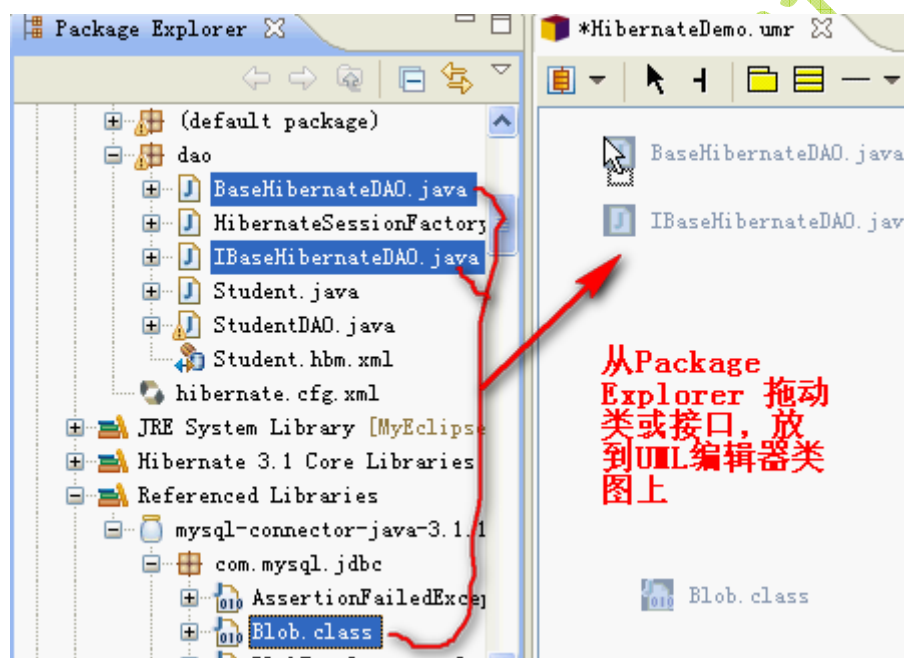


图 17.20 拖放类和接口

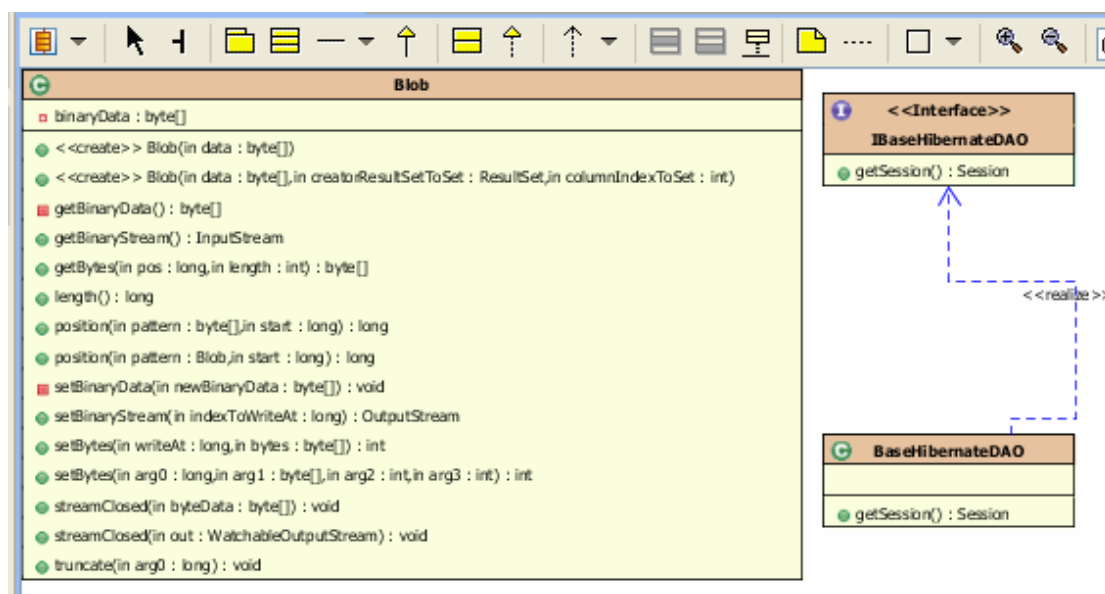


图 17.21 反向过程向导所生成的类图

17.4.5 绘图工具

除了绘制标准的 UML 类图之外，很多时候在图上加入一些自由绘制的形状标注，或者多边形，或者任意位置的提示文字，会更便于阅读和使用一些。MyEclipse UML 包含了一组画图工具，可以用来画一些简单的形状或者是加入文字，这些内容可以放入任意类型的 UML 图中。绘图工具位于 UML 编辑器的工具栏上，并且可以通过下拉的方式选择需要绘制的形状，如图 17.22 左侧图所示。当选中了一种图形后，工具栏上的按钮就可以保持此形状的选中状态，便于以后重复画图。图 17.22 右侧图展示了绘制结果。

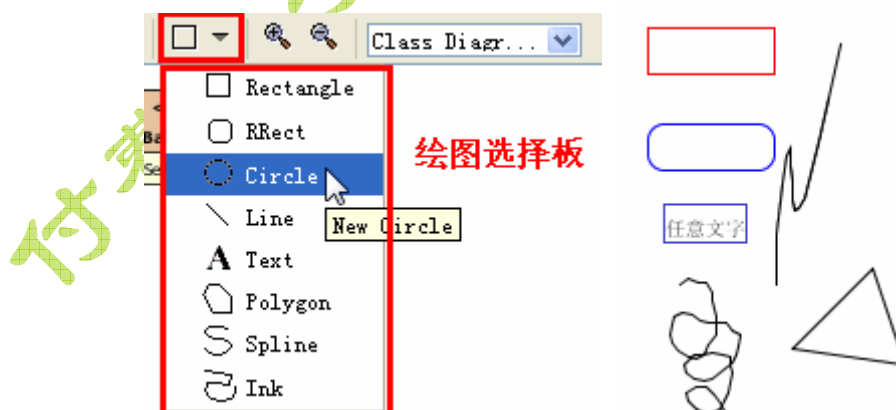


图 17.22 绘图工具栏下拉菜单及绘制的图形

绘图工具支持的类型有：Rectangle — 矩形；RRect — 圆角矩形；Circle — 环形；Line — 直线；Text — 带框的文本；Polygon — 多边形；Spline — 拟合曲线；Ink — 墨水印。线型和颜色，以及文字的颜色和字体等属性，都可以在 **Properties** 视图下的

Presentation 标签页中进行设置。

17.4.6 用例图(Use Case Diagram)

在类图之外，第二个经常用到的 UML 图，也许就是用例图了。用例图描述了系统提供的功能单元。用例图的主要目的是帮助开发团队以一种可视化的方式理解系统的功能需求，包括基于基本流程的“角色”（actors，也就是与系统交互的其他实体）关系，以及系统内用例之间的关系。用例图一般表示出用例的组织关系--要么是整个系统的全部用例，要么是完成具有功能的一组用例。要在用例图上显示某个用例，可绘制一个椭圆，然后将用例的名称放在椭圆的中心或椭圆下面的中间位置。要在用例图上绘制一个角色（表示一个系统用户），可绘制一个人形符号。角色和用例之间的关系使用简单的线段来描述，如图 17.23 所示。

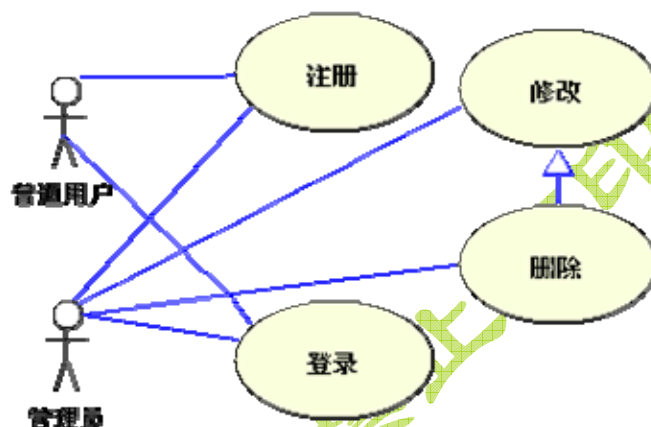


图 17.23 用例图

在这个图中，定义了两个角色：普通用户和管理员，还定义了 4 种用例：注册，登录，修改和删除。不管角色和用例，都是可以集成的，例如上图中删除这个用例继承自修改。集成的符号和类图中的一样，都是一个封闭的实线箭头。每个角色所能参与的用例，通过连接线进行关联。如图中所示的，普通用户只能注册和登录，而管理员则拥有更多的能力。

绘制用例图非常简单，点击工具栏上的 按钮可以创建一个新角色，点击 则创建一个新的用例。和类图编辑一样，双击角色或者用例图，可以快速输入它们的名称。点击 这个按钮即可创建关联关系。其它的功能例如继承，扩展，包含等等，和类图都是一样的，点击按钮后拖放连接即可形成相对关系。

一般来说，用例更多的是一种系统描述图，里面的角色和用例都是和具体类无关的，也不能从它们生成代码，它们代表的是一组相关类的抽象。

现在，就请读者完成如图 17.23 所示的用例练习。

17.4.7 活动图 (Activity Diagram)

活动图表示在处理某个活动时，两个或者更多类对象之间的过程控制流。活动图可用于在业务单元的级别上对更高级别的业务过程进行建模，或者对低级别的内部类操作进行建模。就我个人的经验来说，一般用来绘制系统的流程图。

一个活动图中只能有一个起点和一个终点。如图 17.24 所示，展示了一个用户注册+邮件确认的注册过程活动图。在图中，黑色的实心圆表示起点，而整个活动的结束点以实心圆

环表示。活动使用一个圆角矩形（活动的名称包含在内）来表示，例如：*输入用户名*，*确认注册*等等。菱形表示一个分支结构，常代表程序设计语言中的 *if/else* 这样的判断流程，例如*用户名有效性检查*这一活动即代表判断流程。活动跳转使用带打开箭头的线段来表示。另外，活动图还支持用分叉和合并来表示并发的流程处理表示，例如图中的输出提示信息 and 发送确认邮件，是两个并列进行的操作，并在*确认注册*处合并。最后流程从注册成功处转向结束。除此之外，活动图还支持对象流。活动图不但可以显示控制流，还可以显示对象取值的流，对象流的状态表达了对象是活动的输入还是输出。对于输入值，由对象流状态至活动的虚线来表示，如果活动具有一个以上的输出值或后续的控制流，则可以用分支和合并来进行。因为一般较少看到对象流的用法，所以在此我们就不多做介绍了。

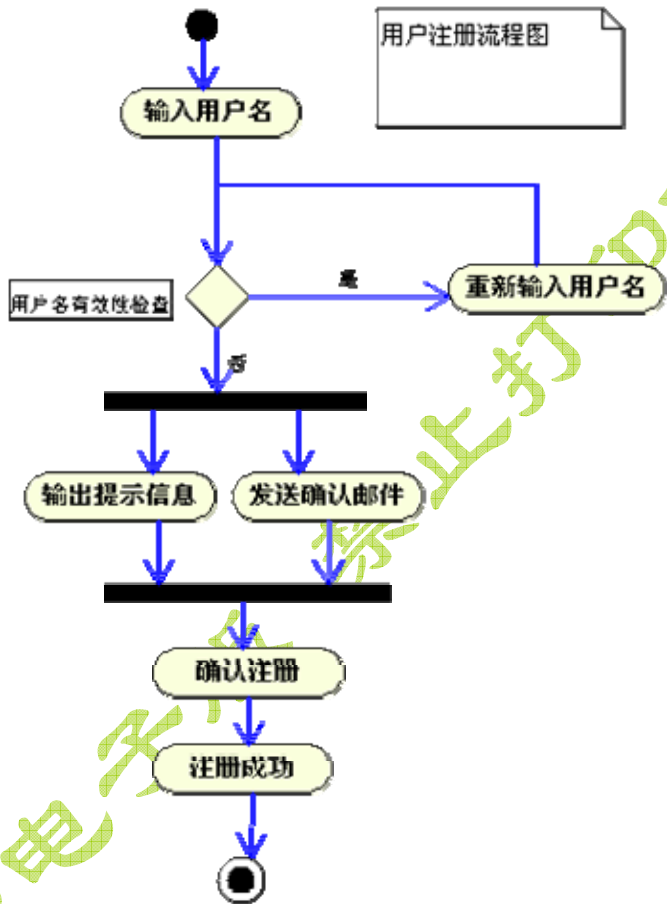


图 17.24 活动图

在 MyEclipse 中创建活动图比较容易，首先请按照 17.4.1 节的内容，创建一个活动图。接着双击活动图在编辑器中打开，可以看到编辑器的工具栏出现和绘制活动图有关的图标。点击对应的图标即可创建节点，双击状态，或者连接线，可以编辑其名称或者提示文字。这些工具如表 17.5 所示。

图标	英文提示	说明
	New Action State	新建活动状态
	New Transition	新建跳转
	New Initial	新建初始状态
	New Final State	新建结束状态
	New Junction	新建分支结构
	New Fork	新建分叉



	New Join	新建合并
	New Object Flow State	新建对象流状态

表 17.5 活动图编辑器工具栏

现在请读者按照视频进行练习，完成如图 17.24 所示的活动图。提示：Transition 连接线可以拐弯，先点击起始状态的圆角矩形，然后点击一点，最后再点击终点，如果重点在另一个跳转线上，那么两条线将会合并为一条。

17.4.8 序列图 (Sequence Diagram)

序列图显示具体用例（或者是用例的一部分）的详细流程。它几乎是自描述的，并且显示了流程中不同对象之间的调用关系，同时还可以很详细地显示对不同对象的不同调用。序列图有两个维度：垂直维度以发生的时间顺序显示消息/调用的序列；水平维度显示消息被发送到的对象实例。一个序列图的例子如图 17.25 所示。

序列图的绘制非常简单。横跨图的顶部，每个框（参见图）表示每个类的实例（对象）。在框中，类实例名称和类名称之间用空格/冒号/空格来分隔，例如，张三:Male。如果某个类实例向另一个类实例发送一条消息，则绘制一条具有指向接收类实例的开箭头的连线，并把消息/方法的名称放在连线上，例如追求(rose)。对于某些特别重要的消息，您可以绘制一条具有指向发起类实例的开箭头的虚线，将返回值标注在虚线上，例如拒绝:boolean。如果绘制出包括返回值的虚线，那么这些额外的信息可以使得序列图更易于阅读。

阅读序列图也非常简单，图 17.25 表示了一个张三追求小雯并间接和小雯妈产生联系的序列过程。从左上角启动序列的“驱动”类实例开始，然后顺着每条消息往下阅读。虽然图中所示的例子序列图显示了每条被发送消息的返回消息，但这只是可选的。另外，实例也可以自己调用自己，例如张三高兴的唱歌()方法。

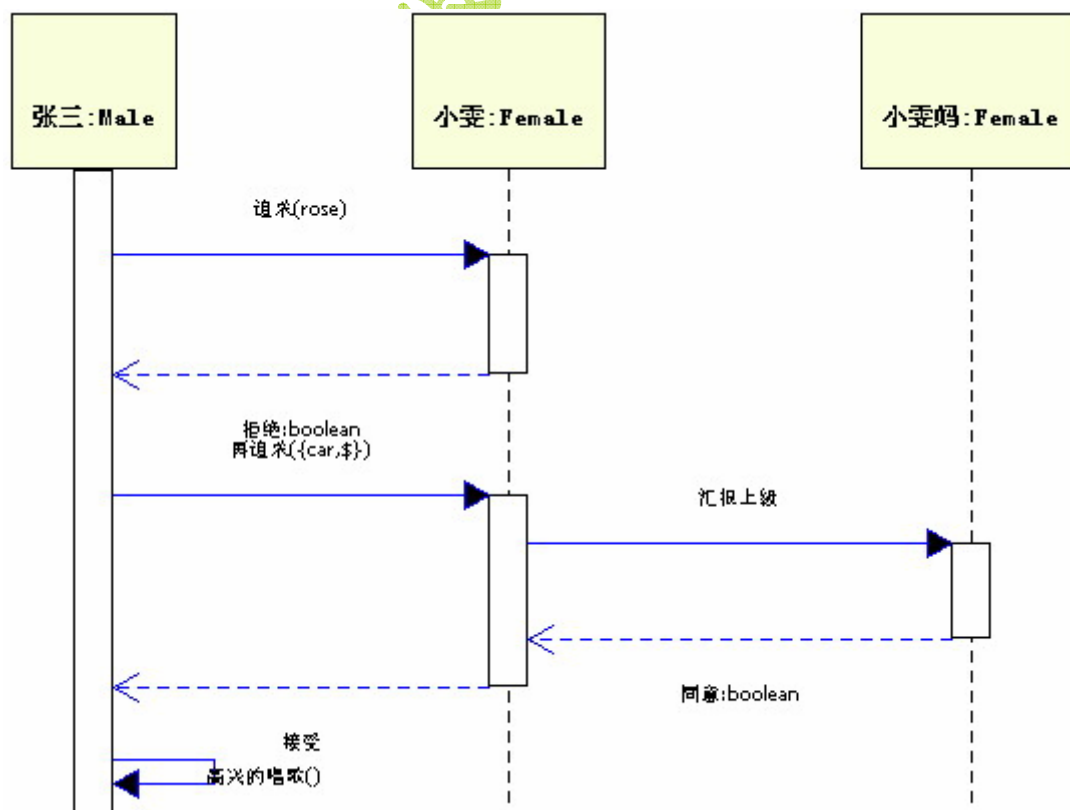


图 17.25 序列图

相比较起来,人们一般认为序列图表示的流程要比活动图精确些,但是序列图的绘制显然要多画点时间。另外,有的建模工具,例如 Together,支持从 Java 中的方法源代码直接生成对应的序列图。在 MyEclipse 中绘制序列图,首先请按照 17.4.1 节的内容,创建一个序列图。接着双击活动图在编辑器中打开,可以看到编辑器的工具栏出现和绘制活动图有关的图标。绘制的顺序是先列出顶部的参与对象列表,例如张三,小雯,小雯妈,然后再绘制方法调用过程以及返回值。至于异步调用之类的不常用的内容,我们就不做过多介绍了。这些工具如表 17.6 所示。

图标	英文提示	说明
	New Classifier Role	新建类角色
	New Call	新建调用
	New Call Self	新建自身调用
	New Return	新建返回值
	New Create	新建创建
	New Destroy	新建销毁
	New Async Call	新建异步调用

表 17.6 活动图编辑器工具栏

好了,现在就请读者练习完成图 17.25 所示的序列图。或者有兴趣可以自己选择练习一下其它的按照时间顺序进行的过程,例如 JDBC 代码编写过程,老板和员工的交互过程等等。

17.5 常见问题

问: MyEclipse UML 是否兼容 Argo UML 的图?

答: MyEclipse UML 只能兼容 Argo UML 的早期版本 (0.17 版本或者更早)的.zargo(打包)文件格式,这是因为 MyEclipse UML 虽然基于 Argo UML 开发,但是做了很多改进,而且后续版本也并不和 Argo UML 随时保持一致。如果要在 MyEclipse 中使用已经编写好的 Argo UML 模型文件,请按照下面的步骤进行:

将.zargo 文件复制到一个 Eclipse 项目中;

使用 MyEclipse UML Editor 打开.zargo 文件来浏览并修改图形。可以在文件上选择右键菜单 **Open With > Other...**,在弹出的编辑器列表中选择 **MyEclipse UML Editor**,然后点击 OK 按钮关闭对话框即可。

问: 如何对 UML 的全局颜色,字体,线宽进行设置,好让它看起来更像 ROSE 作出的图?

答: 选择菜单 **Window > Preferences**,在弹出的设置对话框左侧树上展开并选择 **MyEclipse > UML**,即可在右侧选项卡中设置多种和 UML 建模有关的属性了,参考图 17.26。另外,代码生成(Code Generation)和反向工程(Reverse Engineering)的选项也在这里进行设置。

问: UML 是否能代替 Java 进行编程?

答: UML 只能用于软件建模,虽然目标是通用建模语言,它相当于制图的标准,并不

能进行任何特定语言的开发，甚至也不能被运行，它是供人使用的图纸。

问：我如何查看 MyEclipse 自带的 UML 开发帮助文档和教程？

答：点击菜单 **Help > Help Contents**，接着在弹出的帮助文档浏览器中选择左侧的 **Contents** 一栏中的树中的节点：**MyEclipse Learning Center > UML Development**，再点击希望浏览的内容即可。注意此帮助文档是英文内容。

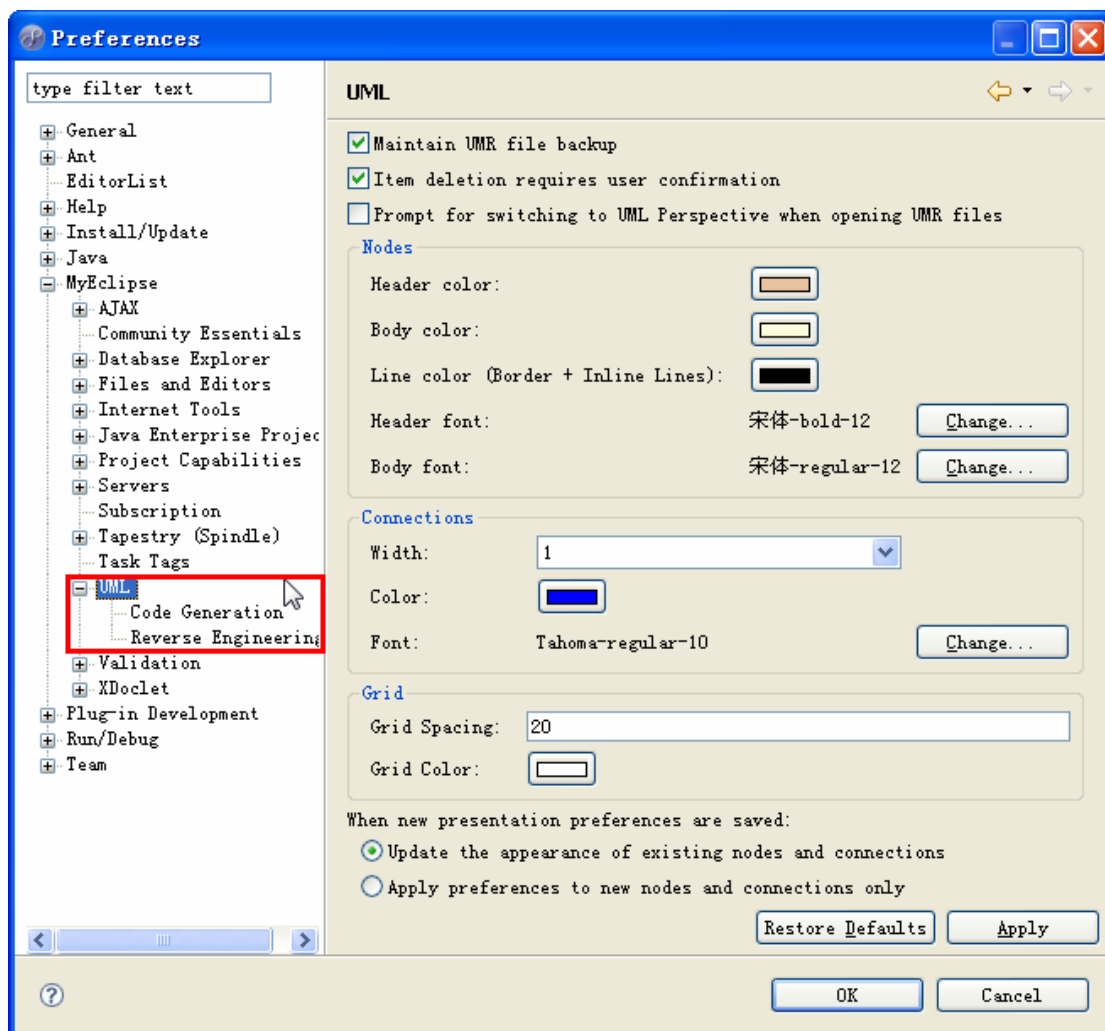


图 17.26 MyEclipse UML 的建模选项

17.6 小结

本章我们介绍了 MyEclipse 中的 UML 开发以及部分常见的 UML 图，而 UML 的概念是所有的建模工具都通用的，因此使用其它建模工具 Rose, Together, Netbeans 时，这些内容同样也可以用得着。由于重点集中在 MyEclipse 使用上，因此 UML 的内容读者可以参考下一节的内容部分或者上网搜索教程即可。

17.7 参考资料

<http://uml.sourceforge.net/index.php> 开源的 UML 建模工具 Umbrello UML Modeller，可用

于Windows, Linux, Mac等多个平台

<http://www.ibm.com/developerworks/cn/rational/products/rose/> ROSE建模软件主页

<http://www.borland.com/us/products/together/index.html> Borland Together建模软件

<http://www.sybase.com/products/modelingmetadata/powerdesigner> Power Designer数据库建模软件

<http://argouml.tigris.org/> 开源 Argo UML 建模软件

<http://www.eclipsedownload.com/download.html> 免费的 Eclipse UML 建模工具，EclipseUML 2007 3.3.0 Free is Eclipse 3.3 and Europa compliant

<http://www.trufun.net/> 楚凡科技，国产UML建模工具，支持Eclipse 3.3，UML 2.0，提供免费版本

<http://staruml.sourceforge.net/> 免费开源的StarUML，只有Windows版本

http://www.ibm.com/developerworks/cn/rational/321_uml/ 统一建模语言(UML) 版本 2.0

<http://www.ibm.com/developerworks/cn/rational/rationaledge/content/feb05/bell/> UML 基础：类图