**<Learn UML with JUDE>**

**目 录**

[一、Overview 2](#_Toc375821772)

[二、UML and UML tools 2](#_Toc375821773)

[三、Description of JUDE 2](#_Toc375821774)

[四、 Installation of JUDE 4](#_Toc375821775)

[五、Fundamental Components and Basic Operation of JUDE 5](#_Toc375821776)

[六、 UML about Bowling 7](#_Toc375821777)

[七、 Draw a UseCase diagram 8](#_Toc375821778)

[八、Draw an Activity Diagram 11](#_Toc375821779)

[九、 Draw a Class Diagram 13](#_Toc375821780)

[十、 Draw a Sequence diagram 16](#_Toc375821781)

[十一、Other Diagrams 19](#_Toc375821782)

[十二、 Features of JUDE 20](#_Toc375821783)

JUDE是一个开源的轻量级建模工具。本文将通过一系列的实例指导大家如何使用JUDE来画UML。通过一些实例去学习使用JUDE来画UML。

# 一、Overview

    UML and UML tools  
     Description of JUDE  
     Installation of JUDE  
     Fundamental Components and Basic Operation of JUDE  
     UML about Bowling  
     Draw UseCase Diagram  
     Draw Activity Diagram  
     Draw Class Diagram  
     Draw Sequence Diagram  
     Other Diagrams  
      Features of JUDE

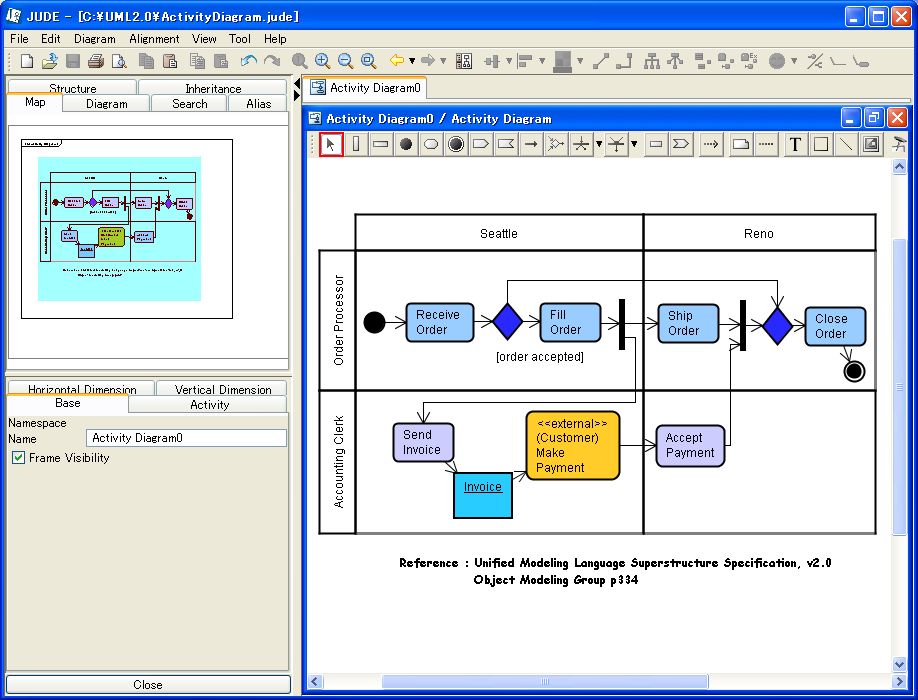
# 二、UML and UML tools

当你画UML的时候，是直接手画还是使用工具呢？使用工具的话我们可以利用工具本身的一些有利条件，比如：

   1、 可以让你画一个干净的diagram；  
2、 元素的尺寸可以画的比较合适；  
3、 方便和其它人分享、交流你的UML模型；  
4、 在需要的时候你可以重复的执行undo或redo操作；  
5、 当你的diagram不符合UML规范时，会给你一个警告提示；  
6、 数据可以通过多种方式输入（图表，树形结构图等等）；  
7、 数据可以很容易的被重用；  
8、 你可以管理所有的数据和它们之间的关系；

如果你是初学UML，你会发现第四和第五条对你非常有用。因为出现的错误可以很容易的被“undo”，所以你能画出比较完善的UML。同样，当你创建了一个抽象类时，抽象类的类名将会根据规范的要求自动以斜体的方式进行显示。使用JUDE绘画UML将会使你从中得到很多有利的条件，JUDE将会成为你学习UML的导师，去尝试一下吧。记住，如果你画错了，你总是可以使用“Undo”来重新画的。

# 三、Description of JUDE

现在有很多UML工具可以供你使用，我比较喜欢的是“JUDE/Community”（JUDE代表Java and UML Developer's Environment）。我是有点偏心的，因为我是JUDE开发团队的成员，开发和完善JUDE已经多年了。我想让更多的人使用JUDE，并像我一样去喜欢上它。  
        
    
    JUDE有两个版本：JUDE/Community（免费版）和JUDE/Professional（商业版）。JUDE具有以下的特性：

         JUDE/Community支持所有基于UML1.4的diagram；

         JUDE/Professional支持所有基于UML1.4的diagram，一些UML2.0的规范和E-R图（E-R图从JUDE/Professional version3.2开始支持）；

         无限的UNDO/REDO操作；

         拖拽操作，自由切换使用“Map View”以方便编辑较大的diagram；

         自动生成类图和model的信息；

 更多的关于JUDE的特性，请参考：[JUDE/Community page](http://jude.change-vision.com/jude-web/product/community.html" \t "_blank)。

# 四、 Installation of JUDE

现在让我们来安装JUDE。要运行JUDE首先需要安装Java环境。

   1、系统要求

|  |  |
| --- | --- |
| **OS:** | Windows 2000, XP, Vista |
| **CPU:** | Pentium III 600MHz or higher |
| **Memory:** | 256MB or more |
| **Java Environment (Required):** | Sun Microsystems J2SE 1.4.1\_07 or JDK5.0 (JDK 5.0 is supported since JUDE/Community 3.1) |

JUDE可以运行在windows，Linux和Mac的J2SE环境中，但是我们没有测试过所有的发行版环境，因此，我们推荐你在windows中安装JUDE。如果你需要安装在windows之外的系统的相关信息，请参考[FAQ](http://jude.change-vision.com/jude-web/support/faq.html" \l "t8" \t "_blank)。

   2、安装Java环境

如果你的机器上已经安装了Java环境，那么你可以跳过这部分内容；如果没有，那请到SUN的网站上下载并安装Java环境。注意以下事项：

         确保安装的java版本满足要求。JUDE可能不能在未支持的Java版本中工作；

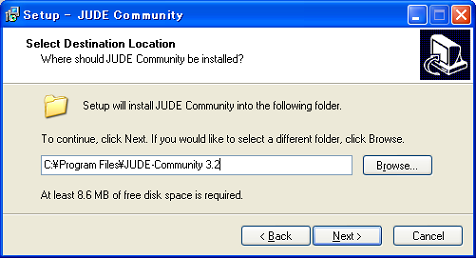
         确保安装的J2SE中包含了JRE（JUDE默认使用JRE）。

   3、下载JUDE

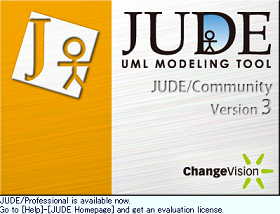
从[Download page](http://jude.change-vision.com/jude-web/download/index.html" \t "_blank)上选择最新版本的jude-community-x\_x-setup.exe（大约5M）进行下载。（顺便说一下，在这篇文章中我使用的是JUDE/Community 3.2）。

   4、安装JUDE

下载完成后，双击安装JUDE，安装向导会让你指定需要安装在的位置。

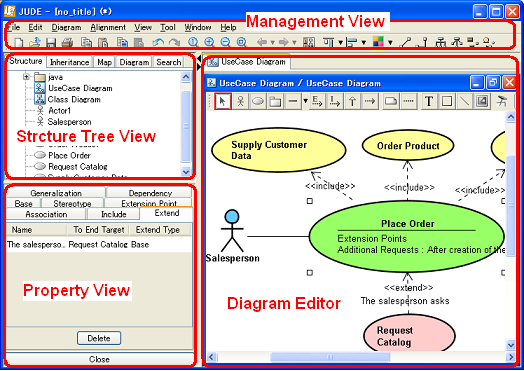
  
  
   如果你使用的是Win98或ME，你需要使用文本编辑器打开jude.bat文件，设置JUDE/HOME变量的绝对路径。jude.bat在JUDE的安装目录中。

   5、运行JUDE

从windows的开始菜单中运行JUDE。当启动JUDE的时候会显示下面这个图标。  
           
     

# 五、Fundamental Components and Basic Operation of JUDE

下面我会通过截屏来指导你进行一些基本的操作。

  
  
   1、基础组件

         管理视图  
    管理视图包括整个project相关的功能按钮，比如：文件操作、编辑和一些经常使用的工具条。

         Diagram编辑器  
    Diagram编辑器用于编辑diagram和model。你可以同时打开多个diagram。

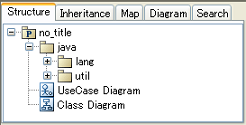
         结构树视图  
    以树型结构的方式显示model。你能够通过model元素的弹出菜单来执行各种不同的操作。

         继承树视图  
    以树型结构的方式显示model中类之间的继承关系。

         Map视图  
    Map视图对在Diagram编辑器中已经打开的diagram提供了概览。可以在这个区域进行拖拽以在Diagram编辑器进行显示。这个功能对较大的diagram是非常有用的。

         Property视图  
    Property视图用于显示和编辑model元素的properties。

   2、创建一个新的project

在JUDE中，UML数据以.jude的后缀进行保存。这些文件被称作project文件。第一次运行JUDE时是没有project打开的。现在就让我们创建一个新的project。从主菜单中选择[File] - [New]，当你创建了一个新的文件时，结构树视图和Property视图就会显示出来了。  
          

新的project在名为“no\_title”的package中创建了用例图和类图。“no\_title”的package对应java中的默认package。在这个指南中我们将不使用输出java的特性。名称为java的package关联java的API，并且持有一些基本的类，比如：String和Vector等。这些类s

   3、创建package和Diagram

通常在结构树视图中通过弹出菜单来创建一个新的package。要创建一个新的diagram的话，从结构树视图的弹出菜单中选择[Create Diagram]或者从主菜单中选择[Diagram]。

   在JUDE/Community(3.2)中你可以创建以下这些diagram：

         Class diagram

         UseCase diagram

         Statechart diagram

         Activity diagram

         Sequence diagram

         Collaboration diagram

         Component diagram

         Deployment diagram

 也支持Object diagram，Package diagram和Robustness diagram。

    4、如何创建和编辑

 使用JUDE画diagram的操作和其它的画图工具基本一样，如果你没有使用过画图工具，那我们现在就开始学习使用吧。

# 六、 UML about Bowling

现在都准备好了，我们可以开始画diagram了。我们画什么diagram呢？你最近有去打过保龄球吗？我非常享受我们公司前几天举办的保龄球赛，我猜很多人都打过保龄球，所以我们就用保龄球来作为这个UML的主题吧。让我们来画一个保龄球的计分系统的UML。  
      http://jude.change-vision.com/jude-web/download/try_uml_img/06.png

   这里有一些要求：

         一旦比赛开始就显示保龄球分数的跟踪记录；

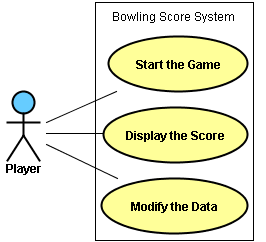
         选手每次扔出保龄球后就更新分数；

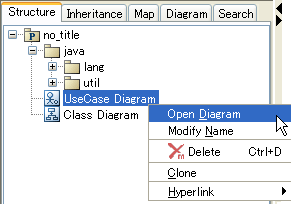
         选手在比赛时能改变分数；

         当有人发起攻击时显示一个动画；

我们的目的是体验使用UML建模工具去画UML，diagram画的是否完美并不重要。那么就让我们开始吧。

# 七、 Draw a UseCase diagram

现在让我们开始画图吧。我们以一个用例图作为开始，我们画一个下图这样的diagram。  
           

打开一个用例图将会伴随打开一些默认的project文件。在结构树视图的Usecase Diagram上点右键，选择[Open Diagram]，或者双击在Diagram编辑器中打开。  
           

在Diagram编辑器中，这里有一个工具按钮面板可以编辑diagram。  
      http://jude.change-vision.com/jude-web/download/try_uml_img/09_en.png

如果你想知道每个按钮的作用，你可以把鼠标放在这个按钮上，悬浮的提示信息将会显示对这个按钮的描述。在Diagram编辑器中，你可以用这些按钮切换画图的模式，比如：当你想在Diagram编辑器中移动一个元素，打开[select mode]你就可以选择你想移动的元素执行移动操作了；如果你想连续的创建某个确定的元素，你可以切换模式到[Lock Selected Mode]等等。

|  |  |
| --- | --- |
| **Button** | **Description** |
| http://jude.change-vision.com/jude-web/download/try_uml_img/10.png | Select Mode. 当这个模式打开时，你就可以对某个元素执行选择、编辑、移动和改变大小的操作了。 |
| http://jude.change-vision.com/jude-web/download/try_uml_img/11.png | Lock Selected Mode. 使你能够连续的创建所选择的元素。比如：你在打开Lock Selected Mode时创建一个Actor，一旦你在Diagram编辑器中添加了一个Actor，你就可以连续的创建更多的Actor。 |
| http://jude.change-vision.com/jude-web/download/try_uml_img/12.png | Set Relation End to the center of the item. It places the ends of lines. (e.g. Associations, Generalizations, or Dependencies) at the center of Model Elements. |

在工具面板上选择Actor，然后在Diagram编辑器中的任何地方点击后将创建一个Actor，输入“Player”作为这个Actor的name。现在让我们像创建Actor一样再创建三个用例。你可以拖拽他们移动元素所在的位置，当你想重命名他们的时候，双击这个元素的名称即可。

   现在我们来画Actor和三个用例之间的连线。每条线都有它们自己的意义，我们在这里需要一条称作Association的线。在工具面板上选择Association按钮去创建一个Association，你有两次点击操作，分别是点击开始目标和结束目标。首先，点击“Player”这个Actor，然后在“Start the Game”这个用例上点击，现在一条association线就画在了“Player”和“Start the Game”之间了，然后再按这种方式创建两条association线。现在再添加一个长方形，并写上“Bowling Score System”就完成了这个diagram了！

**小提示：取消association操作**

当你在画association线时点击在了错误的目标上，按下[ESC]或者右键就可以重做，该方法适用于两个元素之间的任何连线，比如：Generalizations、Generalizations和Dependencies。

你可能注意到当你在Diagram编辑器创建用例和Actor时也添加在了结构树视图中，同样在Diagram编辑器或者结构树视图中选择元素时，它们的property也显示在了左下角的Property视图中，你可以在这几个地方修改数据：Diagram编辑器，结构树视图，Property视图，选择使用对你来说最方便的一个。

让我们使用Property视图来更改用例图的名称，在结构树视图中选择“UseCase Diagram”，它的property将会显示在Property视图中，将它的名称改为“Bowling UseCase Diagram”再按下[Enter]就完成了。

**小提示：删除元素**

要删除Actor或者用例，从它们的弹出菜单中选择[Delete from Model]或者[Delete from Diagram]，提供两种方式删除元素是因为像Actor和用例这样的元素可以和其它的diagram共享，也可被用于其它的diagram。

|  |  |
| --- | --- |
| **[Delete from Diagram] Ctrl+D** | 仅从当前的diagram中删除元素，在model中依然保留。 |
| **[Delete from Model] Delete** | 从所有的diagram中完全删除。 |

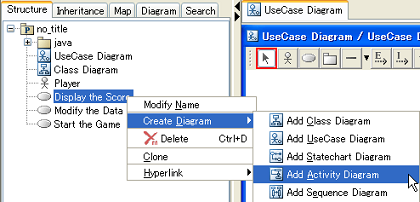
**小提示：缩放和滚动**

我将会给你介绍JUDE中用于查看操作的特性。如果我们只是画较小比例的用例图，那么在视图中不需要用这个特性。当在处理较大比例的diagram时，特别是你需要在diagram的某些部分进行修改，JUDE的这个特性可以非常容易的进行该操作。希望显示的区域可以在JUDE的Diagram编辑器中用右键拖拽，使你可以对角的移动它们，在主菜单中选择[View]或者使用工具条可以进行放大或缩小，也可以使用Map视图指定显示你想看到的区域。

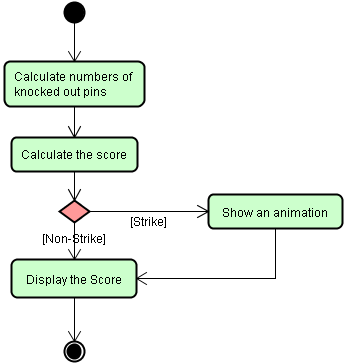
小提示：撤销和重做

JUDE的撤销和重做的特性可以让你重复执行100次！如果你到现在为止都没有用过撤销，去尝试一下看看它是怎么工作的！在你画图的时候不用担心画错了，这是件多么舒心的事！

# 八、Draw an Activity Diagram

现在让我们着重在“Display the Score”用例，在活动图中画出它的流程。首先，你需要在你的project文件中添加一个活动图，选择“Display the Score”用例，然后在弹出菜单中选择[Create Diagram] - [Add Activity Diagram]，一个新的活动图将会被添加，并且在Diagram编辑器中打开。  
      

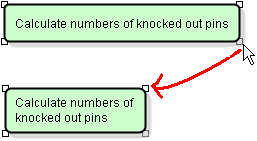
把活动图命名为“Behavior of displaying Scores”。活动图在工具条中有下面这些按钮：  
      http://jude.change-vision.com/jude-web/download/try_uml_img/14_en.png

现在让我们来画一个下面这样的活动图。  
          

活动图和流程图非常相似，它描述了事件如何流动和连接到下个动作。在这个活动图中，描述了在选手扔出保龄球后执行的事件，以及计算将要显示的分数。从这个活动图中，你将会得知在选手全中后会显示一个动画。

让我们开始创建除箭头外的所有元素，你可以像之前创建Actor和用例一样去创建它们，从工具面板选择元素，然后在Diagram编辑器中点击创建它们。你可以在Diagram编辑器或者Property视图中将EntryAction设置为ActionState。

**调整大小**

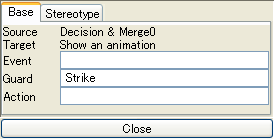
“Calculate numbers of knocked out pins”在ActionState中以一行显示的话过长，所以让我们调整一下这个ActionState的大小，让它的名称换行显示。调整元素的大小只需要选择元素的一个角进行拖拽。  
       

现在让我们添加元素之间的箭头以显示这些动作之间的流程，这个箭头称作Transition。要画一条Transition，你可以像在用例图中画Association一样点击两个目标进行连接，但是你需要小心创建时点击的顺序，我马上会告诉你如何连续的创建T ansition。

**小提示：连续的创建**

         方法1：在工具面板上打开[Selected Mode]；

         当创建元素的时候按住[Shift]；

你能连续创建它们了吗，这对你来说非常有用，也非常容易。[Strike]是Transition的guard的值，点击Transition箭头，然后在Property 视图的Guard列中输入“Strike”。  
     

下面我将会介绍如何画出清晰的diagram的三个技巧。

**小提示：多选**

这里有三种方式在Diagram编辑器中进行多选。

         方法1：在多个元素上拖拽出一个长方形的选择域，长方形中的所有元素都将被选中；

         方法2：在diagram的弹出菜单中选择[Select All]；

         方法3：在选择多个元素时按住[Shif]；

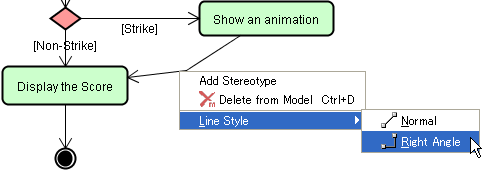
**小提示：线性对齐**

在JUDE中元素可以纵向和横向对齐，使用[Align vertical center]让左边的元素垂直对齐（除了“Show an animation”），让diagram看起来整齐一点，可以用鼠标拖出一个长方形包围所有目标元素以选中它们，然后从主菜单选择[Alignment] - [Align Vertically] - [Align Vertical Center]或者选择主菜单下的工具条中的按钮。

**小提示：线条样式**

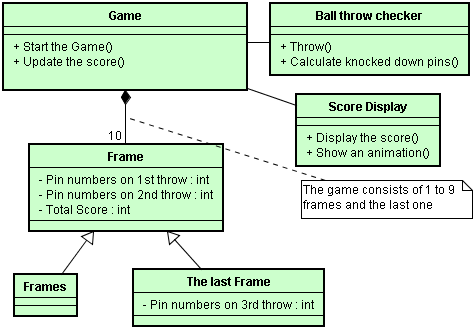
在JUDE中有两种类型的线条。

|  |  |
| --- | --- |
| **Normal Line Style (Default):** | 用直线或多角折线连接diagram 中的元素 |
| **Right Angle Line Style:** | 使用直角线连接diagram中的元素 |

线条样式默认设置为[Normal Line Stile]，然而[Right Angle Line Style]有时看起来比较好，那么我们为什么在这个活动图中不用[Right Angle Line Style]呢，选择所有的线条，然后在它的弹出菜单中选择[Line Style] - [Normal]或者在工具条中选择按钮。  
     

# 九、 Draw a Class Diagram

    现在我们来画一个类图。类图和时序图是UML中最常用的两种diagram，所以我们要比之前的学习多花点时间。



    这个类图着重在分析保龄球比赛和它的分数上，描述了一个保龄球比赛由10个Frame构成，除了最后一个Frame有三次掷球外，其它的每个Frame都只有两次掷球。首先，我们把类图的名字更改为“Analysis Class diagram”。

    类图的工具条中有下面这些按钮。

http://jude.change-vision.com/jude-web/download/try_uml_img/20_en.png

     现在让我们先画出轮廓，然后再添加详细的信息。首先需要创建六个类，你可以像创建Actor和用例一样去创建它们。从关联的两个类中下面那个类开始的箭头称作“Generalization”，当创建Generalizations的时候，需要先在开始目标上点击一下。

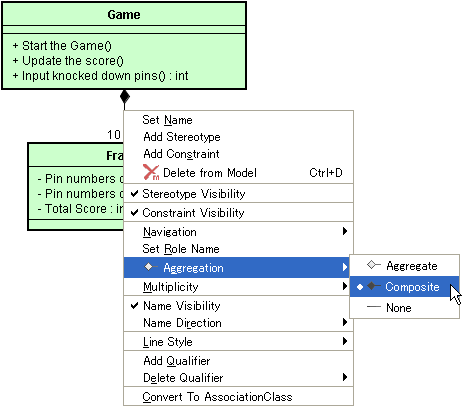
    上面这个类图中的关联除了Generations外，还有一种虚线被称作“Associations”，现在让我们开始画这些Associations。在Game类和Frame类之间这种线被称作“Composition”，表明是Game是由Frame组成的，如果没有Frame，那也就不存在Game。

**如何设置Association的Property**

     Association有很多property可以设置，比如：Aggregation、Composition、Multiplicity等等。在这个case中，你需要设置Game类的Composition。

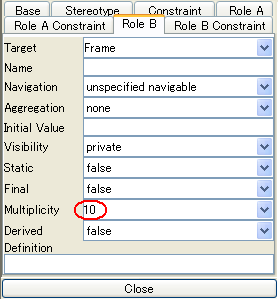
* 方法1：拖拽工具面板上的Association按钮；
* 方法2：在Association的弹出菜单中设置；
* 方法3：在Association的Property视图中设置；

    现在我们采用第二种方法，从弹出菜单中设置Association。



    选择不同的association显示的弹出菜单也是不同的，点击结束于Game类的association连线，你就可以设置Game类的property了。

    现在让我们在Property视图中设置Multiplicity，当在Diagram编辑器中选择Association时，这个Association的property就会显示在Property视图中（看下面的图例），然后你就可以修改Multiplicity了。



    当你设置multiplicity时要注意这里针对association的不同目标显示了两个role选项卡，请确保你选择的是正确的。在这个case中，选择Target为“Frame”的选项卡，然后设置multiplicity为10。

    现在让我们来给这些类添加属性和方法吧。

**添加类属性**

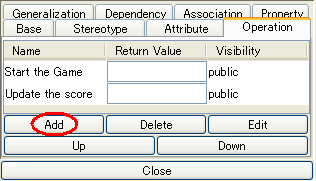
    现在让我们给Frame类添加三个属性，我们可以通过三种方法执行该操作。

* 方法1：在Diagram编辑器中使用弹出菜单；
* 方法2：在project视图中使用弹出菜单；
* 方法3：使用这个类的Property视图；

    在Diagram编辑器中选择Frame类，并且在它的弹出菜单中选择[Add Attribute]，然后输入“Pin numbers on 1st throw”作为属性的名称，你可以使用同样方式添加更多的属性，也可以尝试使用另外两种方法作为练习。如果你想删除一个属性，你可以从Property视图或者结构树视图中去删除它们。属性的可见性默认为private，类型默认为void，你可以使用Property视图去改变他们。你还可以使用快捷键“Ctrl+F”添加属性（添加操作的快捷键是“Ctrl+M”）。

**添加类方法**

    你可以像操作属性一样去添加或者编辑类的方法，也可以在Property视图中添加类的方法。

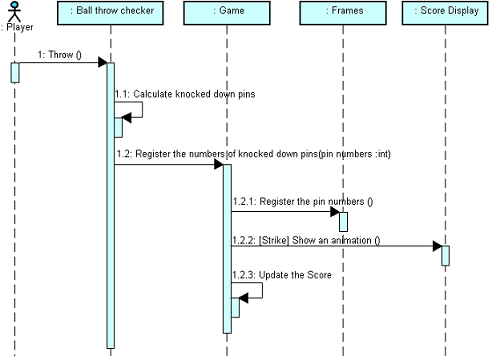


    在Property视图中按下[Add]按钮即可添加方法，要编辑所添加方法的信息，可以使用Property视图或者选中所添加的方法点击[Edit]按钮，你就会看到非常详细的Property视图，可以编辑其中所有的信息，比如这个方法的参数。当选中某个方法时，按下[Delete]按钮就可以执行删除操作，按下[Edit]按钮就会打开一个编辑该方法所有Propety的对话框。使用[Up]和[Down]按钮可以调整方法显示的顺序。

    当你添加完“Calculate knocked down pins”方法和“Pin numbers : int”属性后，该类图就完成了。到目前为止感觉如何？我们主要的目的是学习使用JUDE去画UML的diagram，所以不要担心技术细节，继续下个章节吧。

# 十、 Draw a Sequence diagram

    现在让我们画一个时序图，显示组成Game类的“Calculate knocked down pin numbers”对象的操作时序。时序图可以在结构树视图中添加在用例或者操作的下面，选择“Calculate knocked down pin numbers”方法，然后在弹出菜单中选择[Add Sequence diagram]。



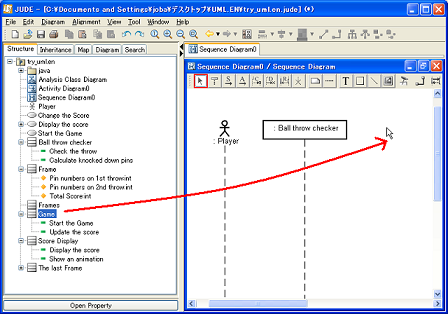
    这个diagram描述了当“Player”掷出保龄球后一系列操作的时序。“Ball Throw Checker”对象首先计算击中了多少个保龄球，然后去“Game”对象进行登记，如果全部被击中的话，“Display the Score”对象就会显示一个动画。如果是画一个类似的活动图会有一些不同，在时序图中，是按时间顺序进行显示的，以竖坐标表示时间，横坐标表示不同的对象。

    时序图的工具面板如下图：

http://jude.change-vision.com/jude-web/download/try_uml_img/25_en.png

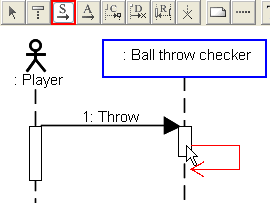
    时序图的工具面板中的按钮和我们之前看过的其它图的有一些不同，所以操作的元素也会和之前的图有所不同。

    首先，按上图创建五个对象，先不要急于完成，我还想告诉你一些事情：我们可以像之前一样选择工具面板上的按钮去创建对象，但是在时序图中创建这些对象有更简便的方法，使用结构树视图！在结构树视图中选择基类，然后拖入Diagram编辑器中的时序图。



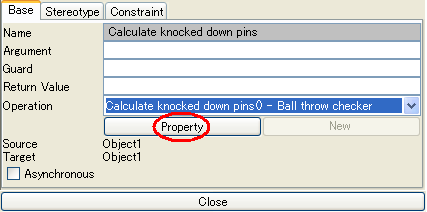
    你也可以在结构树视图中选择多个类一次全部拖入Diagram编辑器中。

    现在我们开始创建消息，从“Player”到“Ball Throw Checker”的消息是个异步消息，在工具面板上选择从左数第四个按钮，然后在“Player”的生命线上点击一下，再点击一下“Ball Checker”，消息就画出来了！双击名称“message0”将其更改为“Throw”，接下来创建消息“1.1 Count number of knocked down pins”，和创建“Throw”类似，我们选择工具面板上左数第三个按钮创建一个从“Ball Throw Checker”发往自己的同步消息（称作SelfMessage）。当你创建异步消息的时候，是点击在生命线上，但是创建同步消息时是点击在Activation上。



**关联消息和操作**

    操作可以关联消息，例如，当一个操作的名称在类图中被更改时，如果该操作有被关联的消息，则消息也将自动被更改。要在消息和操作之间进行关联，选择消息后，在Property视图中设置操作。我们现在创建一个“Calculate numbers of knocked out pins”消息和操作之间的关联。



    如果有些操作已经被关联到了发送消息的基类“Ball Throw Checker”，这些操作将会显示在Operation栏中，如果消息没有可关联的操作，你可以用Property视图中的[New]按钮添加一个目标操作，按下[Property]按钮将会显示出你所选择的操作的所有property，你可以修改它的名称和参数。你不需要总是对操作和消息进行关联，只要遵照其它图去确定操作和消息间的关联就可以了。现在让我们去创建其它的消息。

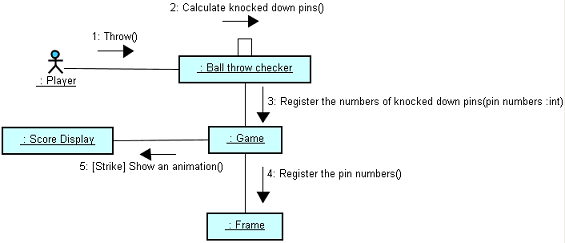
**小提示：Activation的尺寸**

    你可能注意到每次添加或移动一个新消息时，Activation都变长了，这是因为在UML规范中要求原始Activation需要比接受消息的目标Activation要长，所以你每次添加了新消息时，JUDE都会自动进行调整。JUDE建议增长Activation而不让他们变短，你可以自己将它们变得短一点，你可以像在活动图中调整ActionState那样去改变Activation的大小，选择Activation，然后拖住它的一个角进行调整，这个方法适用于除异步外所有消息的Activation。

# 十一、Other Diagrams

    这里有关于保龄球的协作图和状态图。

                                                协作图

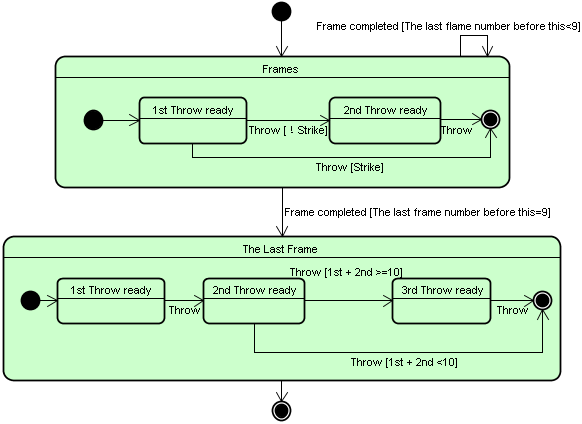


     这个协作图表达了你之前画的时序图中的相互动作。

                               协作图工具面板

http://jude.change-vision.com/jude-web/download/try_uml_img/30_en.png

                                                             状态图



    这个状态图显示了比赛是如何进行的。

                                                       状态图工具面板

http://jude.change-vision.com/jude-web/download/try_uml_img/32_en.png

# 十二、 Features of JUDE

    JUDE有很多在这里没有介绍的特性，去尝试下面的这些特性：

* 导出Java骨架代码
* 导入Java源文件
* Java源文件生成模板
* 在结构树视图中拖拽替换Model
* Boundary、Control和Entity Classes的符号标记
* 设置元素的默认颜色
* 设置模式的默认颜色，可能在JUDE的系统属性中。
* 通过拖拽Association线的角改变Association的目标。
* 向Microsoft Office中贴图
* 导入JUDE的model。

    《Learn UML with JUDE》系列文章到此就结束了，由于时间关系，有些地方翻译的还不够准确，有空的时候再来润色一下。JUDE当前最新的版本是5.5B1，但本文中所使用的版本是3.2，所以有些按钮或操作会和文中的描述有些出入，但也很容易对号入座。在翻译这篇文章的同时，也让我了解了JUDE的使用，确实是款出色的建模工具，推荐大家使用。