



看懂UML类图和时序图【转载】

转载

life_爱水的鱼

2015-09-09 08:50:51



8805



已收藏 7

分类专栏：

软件工程感想与总结

文章标签：

uml

类图

时序图

前言：

本文是在博客上浏览了几篇博文的基础上进行的相应整理总结，感谢原创博主的无私分享。本文最后有附上原创博文地址，UML作为程序员的一个交流与规划工具，我们还是要好好学习其中的约定和表示的：

看懂UML类图和时序图

这里不会将UML的各种元素都提到，我只想讲讲类图中各个类之间的关系；能看懂类图中各个类之间的线条、箭头代表什么意思后，也就足够应对日常的工作和交流；同时，我们应该能将类图所表达的含义和最终的代码对应起来；有了这些知识，看后面章节的设计模式结构图就没有什么问题了；

本章所有图形使用Enterprise Architect 9.2来画,所有示例详见根目录下的design_patterns.EAP

从一个示例开始

点赞5

评论1

分享

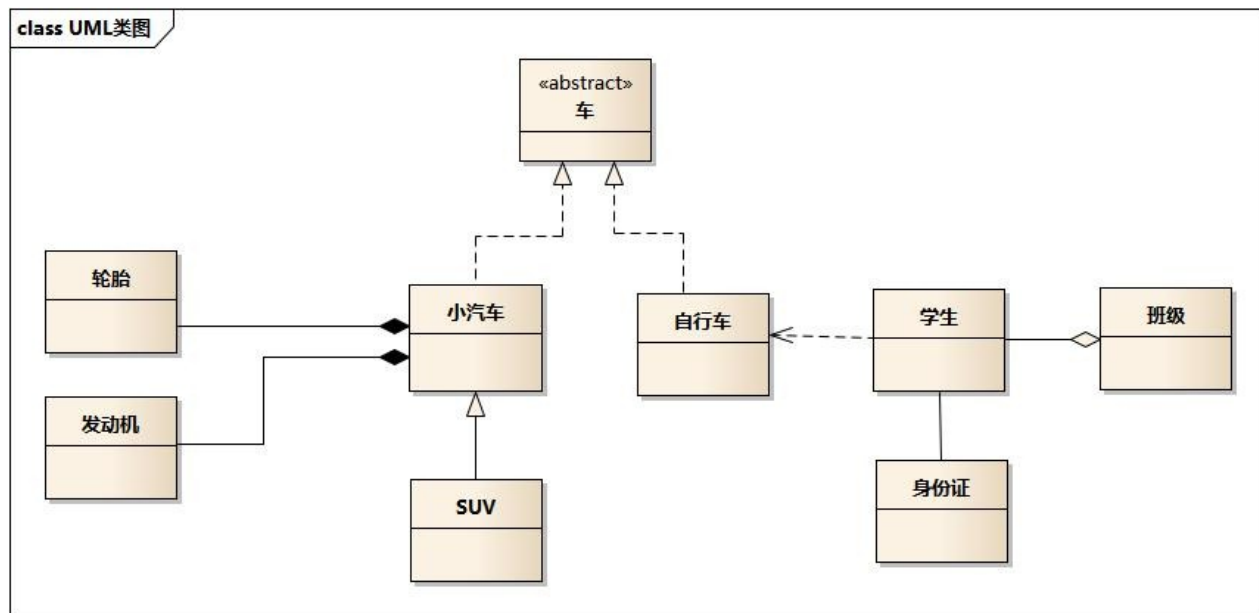
已收藏7

举报

关注

一键三连

请看以下这个类图，类之间的关系是我们需要关注的：



- 1 车的类图结构为<>，表示车是一个抽象类；
- 2 它有两个继承类：小汽车和自行车；它们之间的关系为实现关系，使用带空心箭头的虚线表示；
- 3 小汽车与SUV之间也是继承关系，它们之间的关系为泛化关系，使用带空心箭头的实线表示；
- 4 小汽车与发动机之间是组合关系，使用带实心箭头的实线表示；
- 5 学生与班级之间是聚合关系，使用带空心箭头的实线表示；
- 6 学生与身份证之间为关联关系，使用一根实线表示；
- 7 学生上学需要用到自行车，与自行车是一种依赖关系，使用带箭头的虚线表示；

下面详细介绍这六种关系；

类之间的关系

1：泛化关系(Generalization)

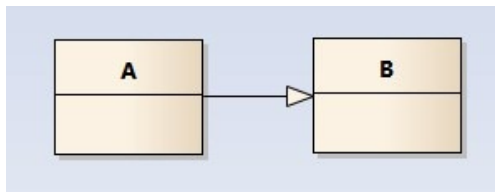
【泛化关系】：是一种继承关系，表示一般与特殊的关系，它指定了子类如何特化父类的所有特征和行为。

类的继承结构表现在UML中为：泛化(generalize)与实现(realize)：

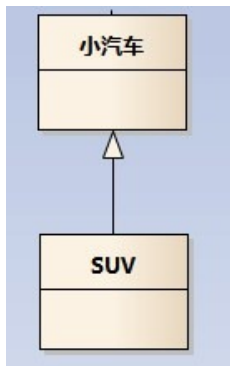
继承关系为 is-a 的关系；两个对象之间如果可以用 is-a 来表示，就是继承关系：（..是..）

eg：自行车是车、猫是动物

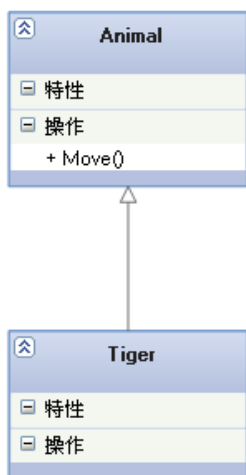
泛化关系用一条带空心箭头的直接表示；如下图表示（A继承自B）；



eg：汽车在现实中有实现，可用汽车定义具体的对象；汽车与SUV之间为泛化关系；



eg：例如：老虎是动物的一种，即有老虎的特性也有动物的共性；



1 | 注：最终代码中，泛化关系表现为继承非抽象类；

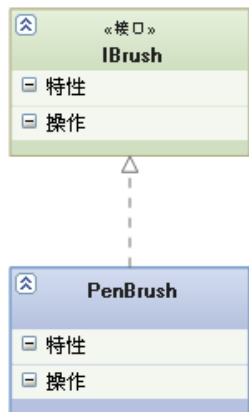
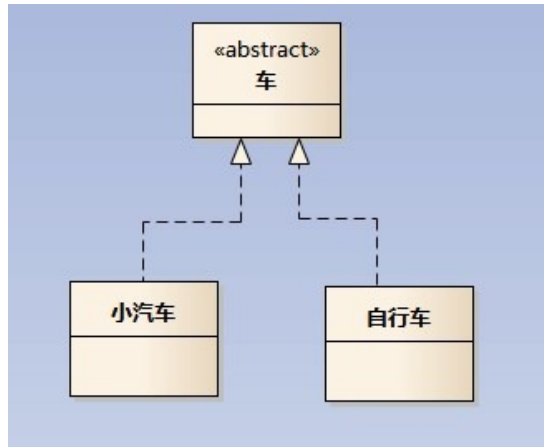
2：实现关系(Realization)

【实现关系】：是一种类与接口的关系，表示类是接口所有特征和行为的实现。

1 | 【箭头指向】：带三角箭头的虚线，箭头指向接口

实现关系用一条带空心箭头的虚线表示；

eg：“车”为一个抽象概念，在现实中并无法直接用来定义对象；只有指明具体的子类(汽车还是自行车)，才可以用来定义对象（“车”这个类在C++中用抽象类表示，在JAVA中有接口这个概念，更容易理解）



1 | 注：最终代码中，实现关系表现为继承抽象类；

3：聚合关系(Aggregation)

【聚合关系】：是整体与部分的关系，且部分可以离开整体而单独存在。如左和轮胎是整体和部分的关系，轮胎离开车仍然存在。

点赞5

评论1

分享

已收藏7

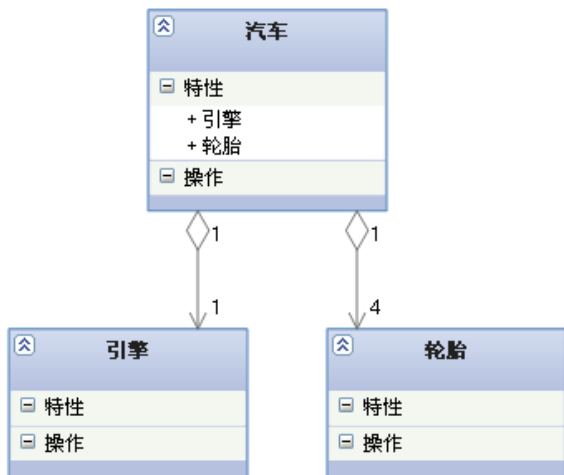
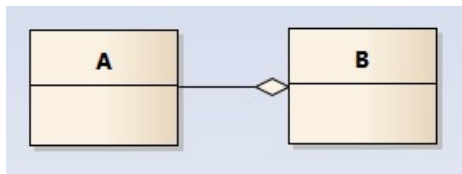
举报

关注

一键三连

- 1 | 聚合关系是关联关系的一种，是强的关联关系；关联和聚合在语法上无法区分，必须考察具体的逻辑关系。
- 2 |
- 3 | 【箭头及指向】：带空心菱形的实心线，菱形指向整体

聚合关系用一条带空心菱形箭头的直线表示，如下图表示A聚合到B上，或者说B由A组成；



聚合关系用于表示实体对象之间的关系，表示整体由部分构成的语义；例如一个部门由多个员工组成；与组合关系不同的是，整体和部分不是强依赖的，即使整体不存在了，部分仍然存在；例如，部门撤销了，人员不会消失，他们依然存在；

- 1 | 【代码体现】： 成员变量

👍 点赞5

💬 评论1

🔄 分享

★ 已收藏7

🚩 举报

关注

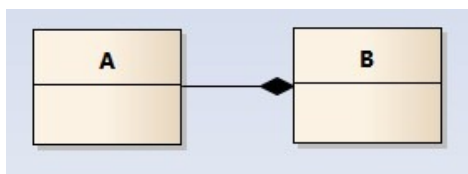
一键三连

4：组合关系(Composition)

【组合关系】：是整体与部分的关系，但部分不能离开整体而单独存在。如公司和部门是整体和部分的关系，没有公司就不存在部门。

- 1 组合关系是关联关系的一种，是比聚合关系还要强的关系，
- 2 它要求普通的聚合关系中代表整体的对象负责代表部分的对象的生命周期。
- 3
- 4 【箭头及指向】：带实心菱形的实线，菱形指向整体

组合关系用一条带实心菱形箭头直线表示，如下图表示A组成B，或者B由A组成；



与聚合关系一样，组合关系同样表示整体由部分构成的语义；比如公司由多个部门组成；
但组合关系是一种强依赖的特殊聚合关系，如果整体不存在了，则部分也不存在了；例如，公司不存在了，部门也将不存在了；

1 | 【代码体现】： 成员变量

5：关联关系(Association)

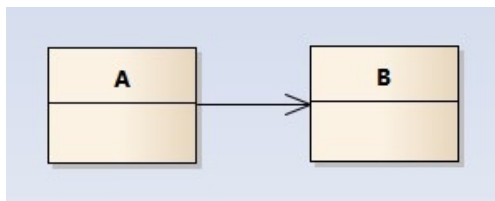
【关联关系】：是一种拥有的关系，它使一个类知道另一个类的属性和方法；如：老师与学生，丈夫与妻子
关联可以是双向的，也可以是单向的。双向的关联可以有两个箭头或者没有箭头，单向的关联有一个箭头。

1 | 【箭头及指向】：带普通箭头的实心线，指向被拥有者

关联关系是用一条直线表示的；它描述不同类的对象之间的结构关系；它是一种静态关系，通常与运行状态无关，一般由常识等因素决定的；它一般用来定义对象之间静态的、天然的结构；所以，关联关系是一种“强关联”的关系；

比如，乘车人和车票之间就是一种关联关系；学生和学校就是一种关联关系；

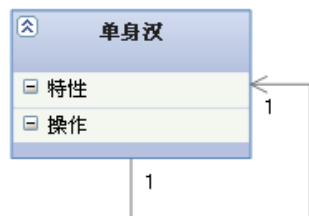
关联关系默认不强调方向，表示对象间相互知道；如果特别强调方向，如下图，表示A知道B，但 B不知道A；





上图中，老师与学生是双向关联，老师有多名学生，学生也可能有多名老师。但学生与某课程间的关系为单向关联，一名学生可能要上多门课程，课程是个抽象的东西他不拥有学生。

下图为自身关联：



1 | 注：在最终代码中，关联对象通常是以成员变量的形式实现的；

6：依赖关系(Dependency)

【依赖关系】：是一种使用的关系，即一个类的实现需要另一个类的协助，所以要尽量不使用双向的互相依赖。

1 | 【箭头及指向】：带箭头的虚线，指向被使用者

依赖关系是用一套带箭头的虚线表示的；如下图表示A依赖于B；他描述一个对象在运行期间会用到另一个对象的关系；

👍 点赞5

💬 评论1

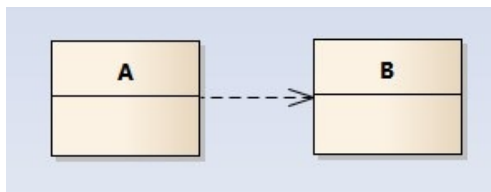
🔗 分享

★ 已收藏7

🚩 举报

关注

一键三连



与关联关系不同的是，它是一种临时性的关系，通常在运行期间产生，并且随着运行时的变化；依赖关系也可能发生变化；

显然，依赖也有方向，双向依赖是一种非常糟糕的结构，我们总是应该保持单向依赖，杜绝双向依赖的产生；

注：在最终代码中，依赖关系体现为类构造方法及类方法的传入参数，箭头的指向为调用关系；依赖关系处理临时知道对方外，还是“使用”对方的方法和属性；

1 | 【代码表现】： 局部变量、方法的参数或者对静态方法的调用

UML类图小结：

各种关系的强弱顺序：

👍 点赞5

💬 评论1

🔄 分享

★ 已收藏7

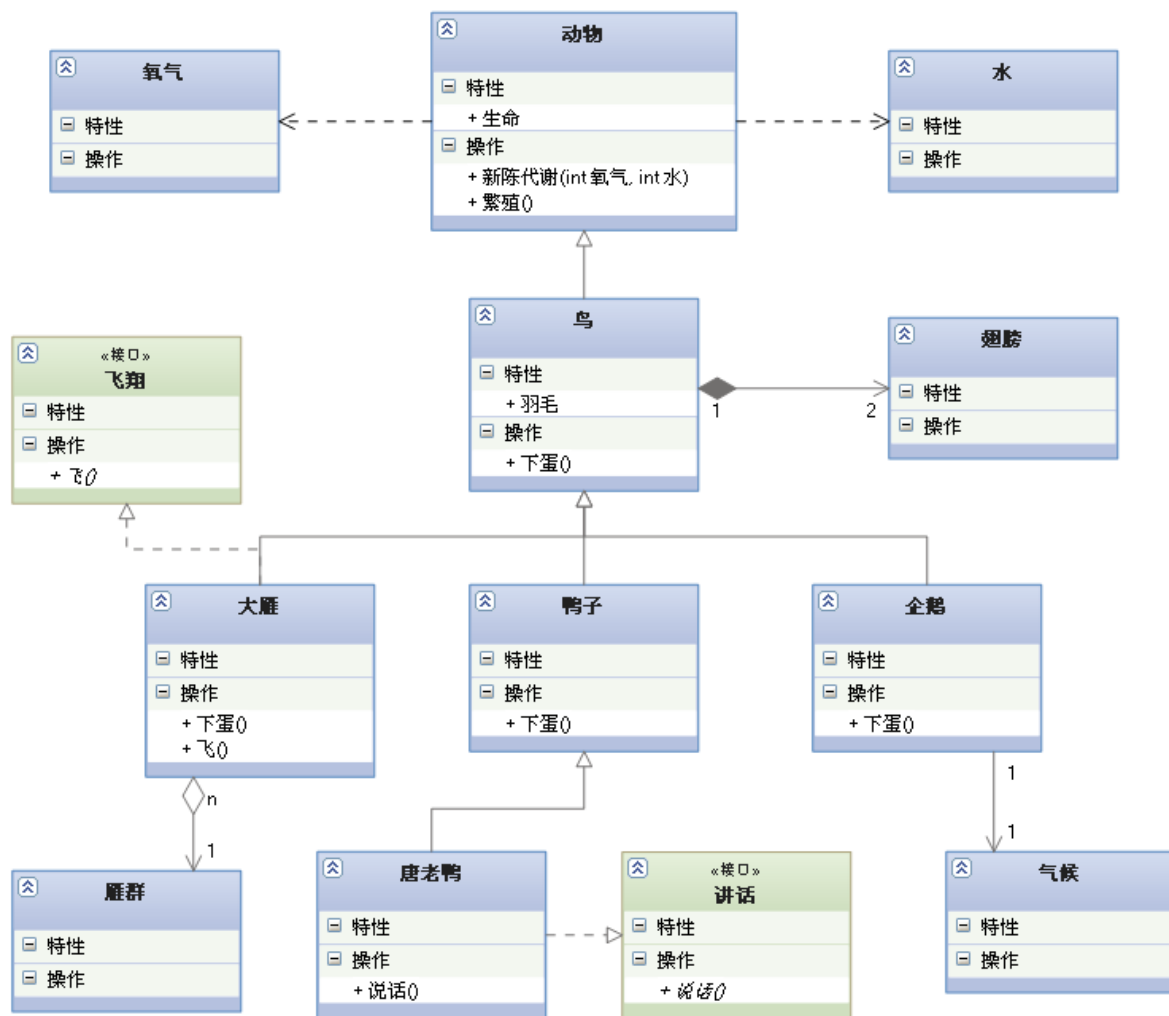
🚩 举报

关注

一键三连

1 | 泛化 = 实现 > 组合 > 聚合 > 关联 > 依赖

下面这张UML图，比较形象地展示了各种类图关系：



时序图

为了展示对象之间的交互细节，后续章节对设计模式中每个

点赞5

评论1

分享

已收藏7

举报

关注

一键三连

时序图 (Sequence Diagram) 是显示对象之间交互的图，这些对象是按时间顺序排列的。时序图中显示的是参与交互的对象及其对象之间消息交互的顺序。

时序图包括的建模元素主要有：对象 (Actor)、生命线 (Lifeline)、控制焦点 (Focus of control)、消息 (Message) 等等。

时序图

时序图是一种强调时间顺序的交互图，在时序图中，首先把参与交互的对象放在图的上方，沿X轴方向排列。通常把发起交互的对象放在左边，较下级对象依次放在 右边，然后把这些对象发送和接受的消息沿Y轴方向按时间顺序从上到下放置。这样就提供了控制流随着时间推移的清晰的可视化轨迹。

时序图元素

1、角色

系统角色，可以是人、及其甚至其他的系统或者子系统

2、对象

对象包括三种命名方式：

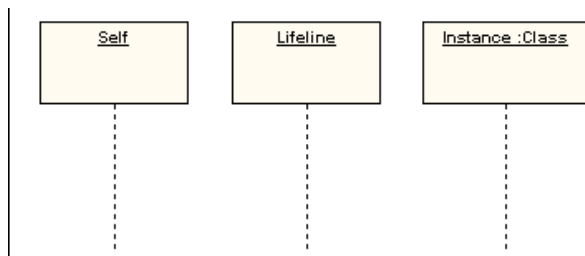
第一种方式包括对象名和类名；

第二中方式只显示类名不显示对象名，即表示他是一个匿名对象；

第三种方式只显示对象名不显示类明。

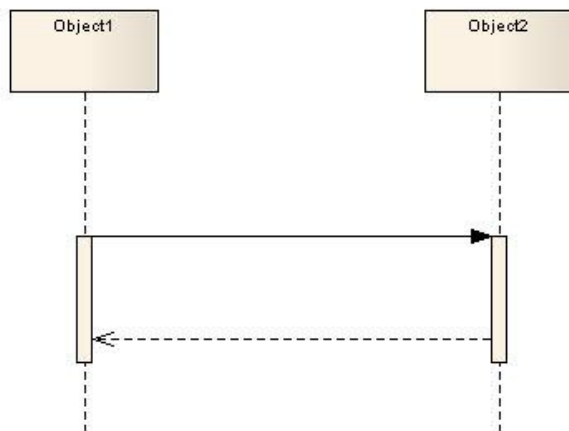
3、生命线

生命线在顺序图中表示为从对象图标向下延伸的一条虚线，表示对象存在的时间，如下图



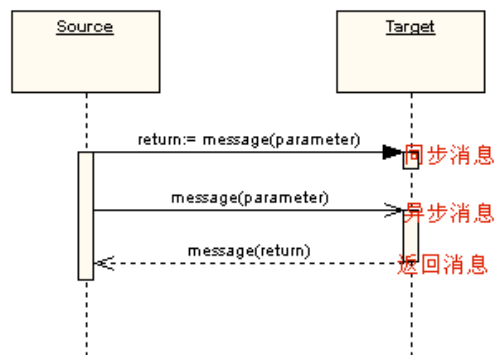
4、控制焦点

控制焦点是顺序图中表示时间段的符号，在这个时间段内对象将执行相应的操作。用小矩形表示



5、消息

消息一般分为同步消息（ Synchronous Message ），异步消息（ Asynchronous Message ）和返回消息（ Return Message ）。如下图所示：



同步消息=调用消息 (Synchronous Message)

消息的发送者把控制传递给消息的接收者，然后停止活动，等待消息的接收者放弃或者返回控制。用来表示同步的意义。

异步消息 (Asynchronous Message)

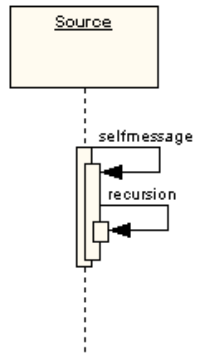
消息发送者通过消息把信号传递给消息的接收者，然后继续自己的活动，不等待接受者返回消息或者控制。异步消息的接收者和发送者是并发工作的。

返回消息 (Return Message)

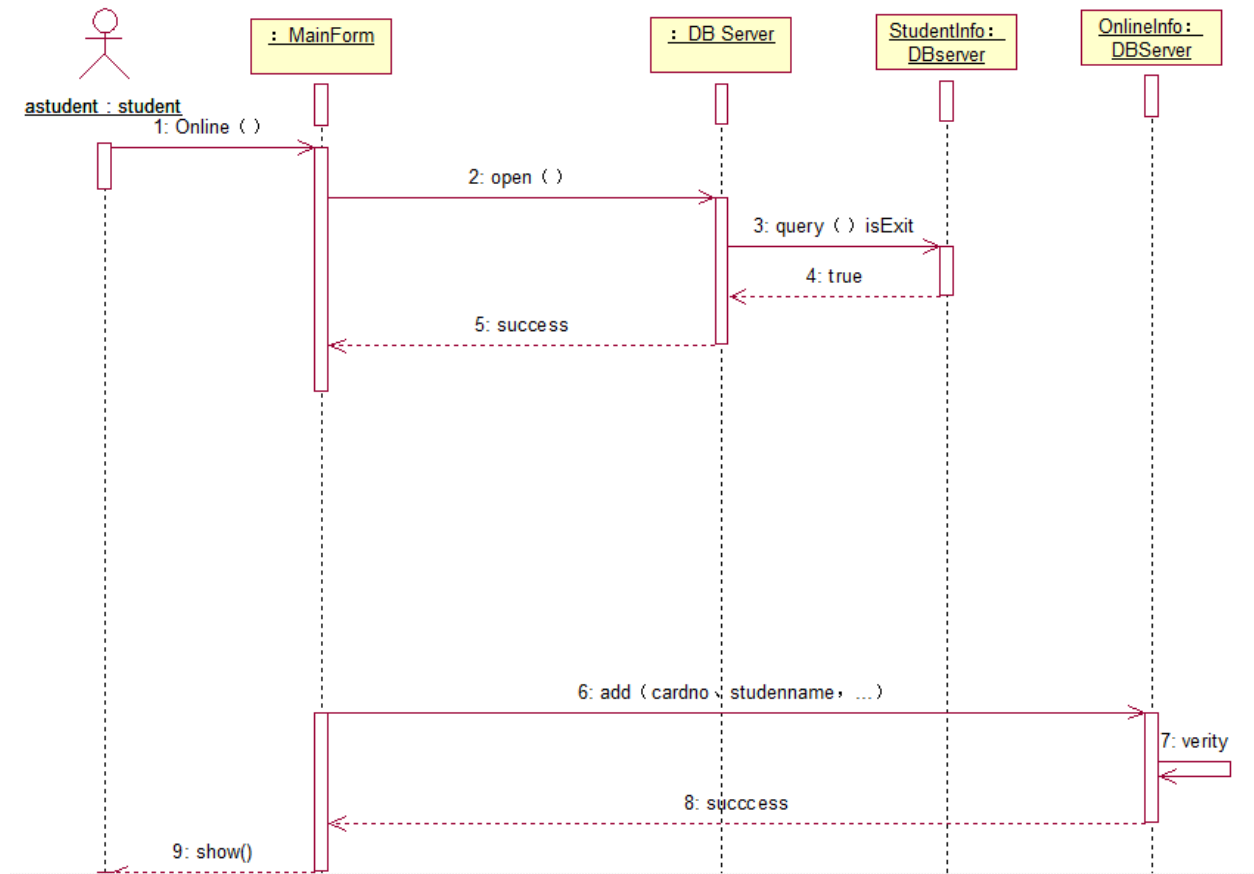
1 | 返回消息表示从过程调用返回

6、自关联消息

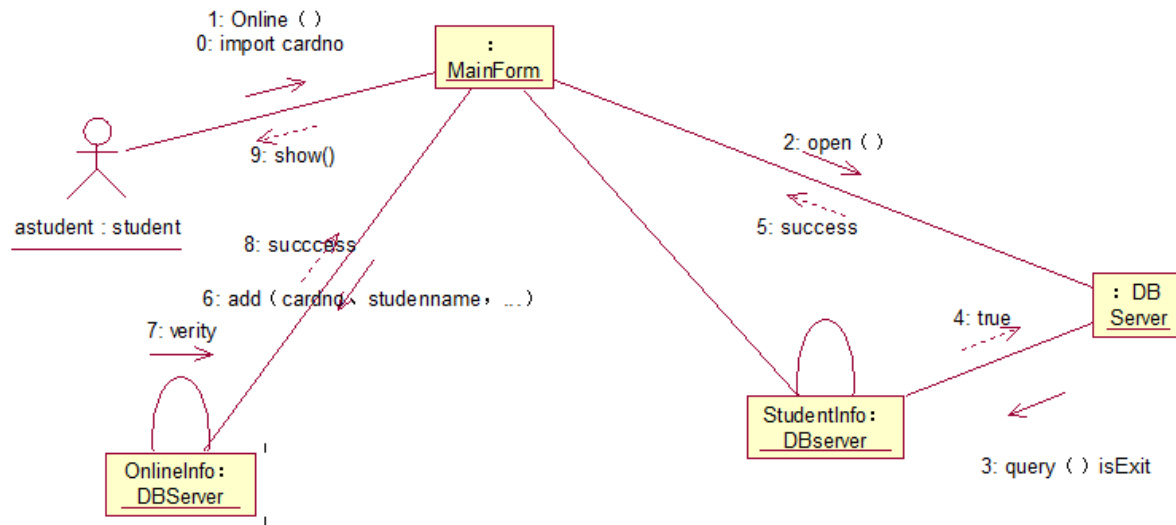
表示方法的自身调用以及一个对象内的一个方法调用另外一个方法



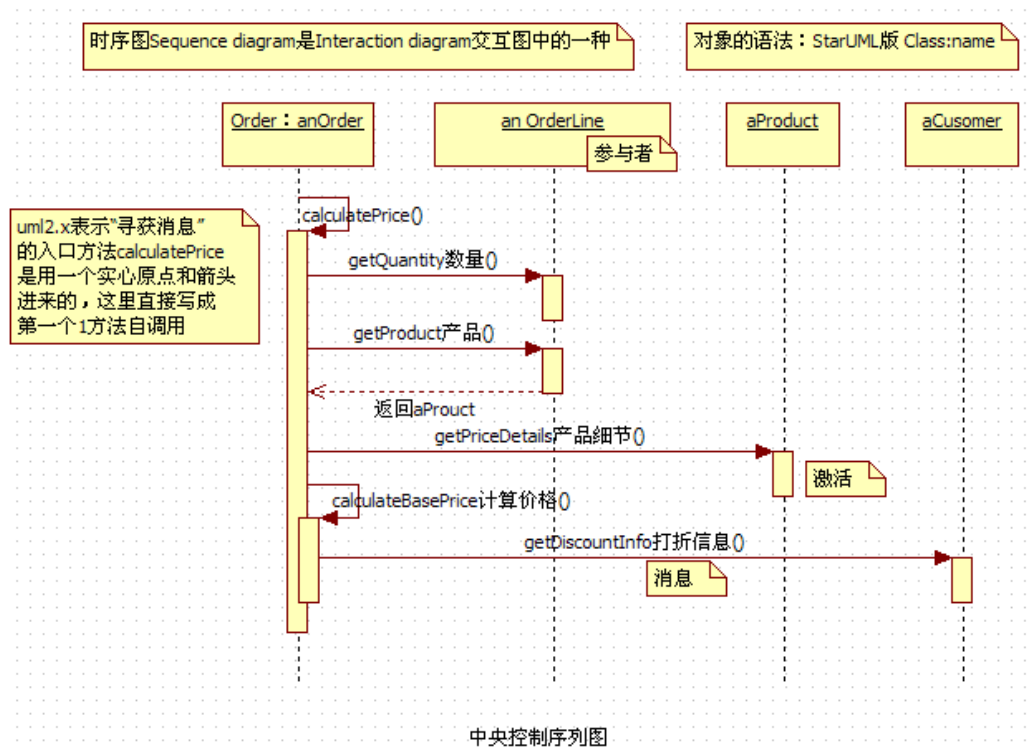
7、时序图实例：



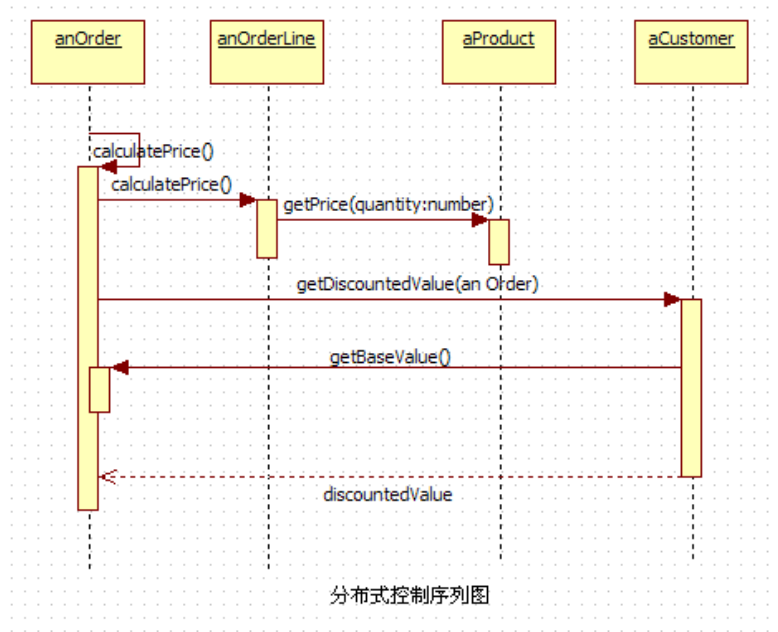
8、时序图和协作图是可以相互转化的，只要按F5就可以轻松转化。



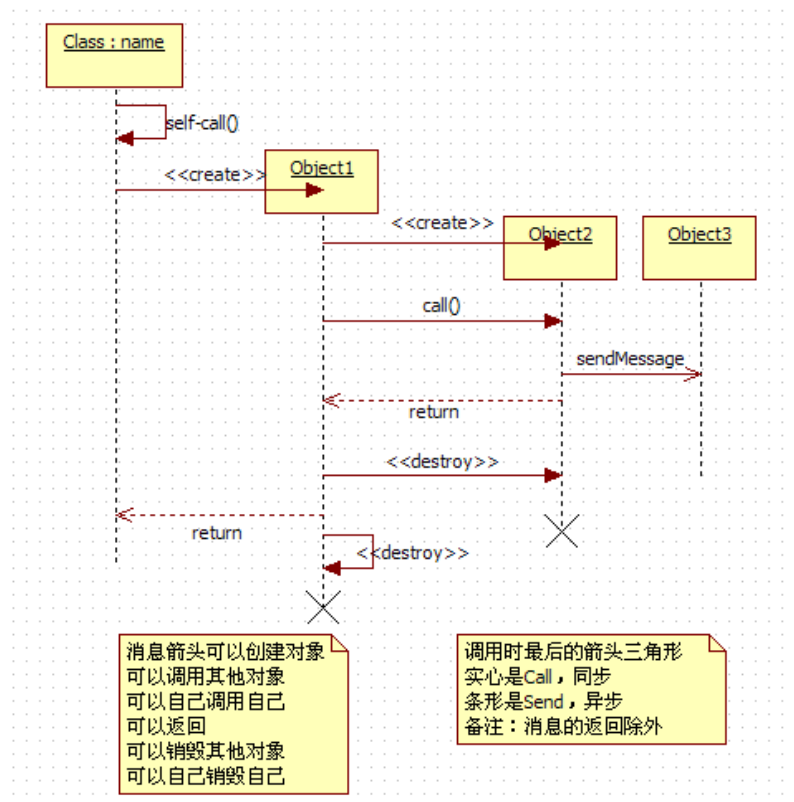
9、中央控制序列图



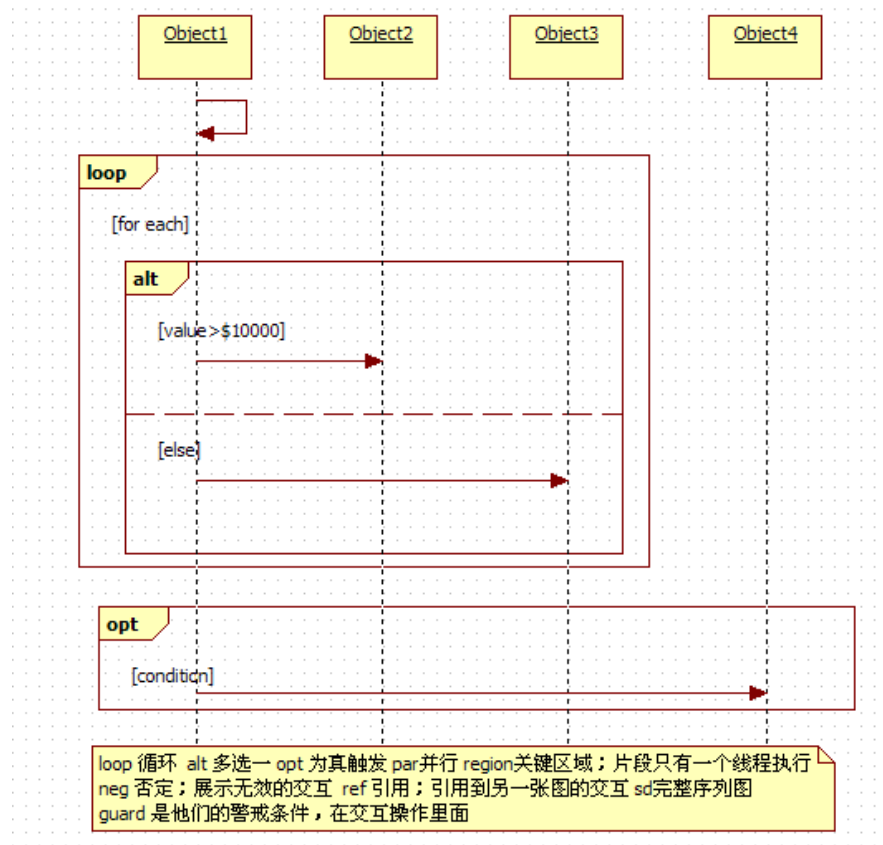
10、分布控制序列图



11、范例：[自调用、调用、创建、销毁、自销毁、同步、异步消息]



12、循环与条件：[loop、alt、opt、guard警戒]



原创地址：感谢以下博主的无私分享，谢谢：

<http://www.cnblogs.com/me115/p/4092632.html>

<http://www.open-open.com/lib/view/open1328059700311.html>

<http://www.cnblogs.com/langtianya/p/3825764.html>

<http://blog.csdn.net/ggibenben1314/article/details/8572431>

<http://blog.csdn.net/tanxiang21/article/details/8222188>

看懂UML类图和时序图

看懂UML类图和时序图 这里不会将UML的各种元素都提到，我只想

👍 点赞5

💬 评论1

🔗 分享

★ 已收藏7

🚩 举报

关注

一键三连

时序图、流程图、状态图、协作图之间的区别 rosekin的专栏 4万+
时序图 **时序图**用于描述对象之间的传递消息的时间顺序,即用例中的行为顺序. 当执行一个用例时, **时序图**中的每条消息对...



优质评论可以帮助作者获得更高权重

评论



冯浩月: 很详细的博文, 谢谢博主的分享。 3年前 回复

Excel——match index 和vlookup函数 和双条件查找匹配 Cutey_thyme的博客 4523
文章目录任务一 任务一 用vlookup函数 查找以下公司的 companyId | companyName | |----| | 上海云贝网络科技有限...

UML常用图的几种关系的总结 拥抱变化 9万+
在**UML**的 **类图**中, 常见的有以下几种关系: 泛化 (Generalization), 实现 (Realization), 关联 (Association), 聚合 (Ag...

看懂UML类图和时序图【转载】 - BAOWHDN的博客 12-13
看懂UML类图和时序图这里不会将**UML**的各种元素都提到,我只想讲讲**类图**中各个类之间的关系; 能**看懂类图**中各个类之间...

看懂UML类图和时序图【转载】 - Jack_SivenChen的博客 - CSDN博客 5-19
转自:**看懂UML类图和时序图【转载】** 前言:本文是在博客上浏览了几篇博文的基础上进行的相应整理总结,感谢原创博主的...

UML入门 之 交互图 -- **时序图** **类图**详解 Channy_的博客 3156
[https://blog.csdn.net/doniexun/article/details/38317537] 一. **时序图** (Sequence Diagram) 1.**时序图**的概念 **时序图**定义: 描...

UML序列图——**时序图**基本使用 ffengz的博客 6万+
简介 基本元素 角色 对象 生命线 激活 消息 同步消息 异步消息 返回消息 自关联消息 组合 选择组合 替换组合 其他 一个简...

看懂UML类图和时序图_pengkai411429850的博客-CSDN博客 6-16
看懂UML类图和时序图 这里不会将**UML**的各种元素都提到,我只想讲讲**类图**中各个类之间的关系; 能**看懂类图**中各个类之间...

看懂UML类图和时序图 - CSDN博客 8-3
这里不会将**UML**的各种元素都提到,我只想讲讲**类图**中各个类之间的关系; 能**看懂类图**中各个类之间的线条、箭头代表什么...

UML类图与序列图的绘制步骤 conganguo的专栏 1万+
https://www.cnblogs.com/perfei/p/7881752.htmlRose模型 (包括所有框图、对象和其他模型元素) 都保存在一个扩展名为...

CSDN开发助手, 集成开发者常用工具, 提升开发效率
支持本地书签、tab页、历史记录搜索; 集成CSDN搜索结果; 他是一个时间转换工具; 他是一个计算器; 他是。。。 , ...

一张图,讲明白**UML类图** - CSDN博客 8-15
一张图带你**看懂UML类图** 3907 一张图带你**看懂UML类图** 用代码画

 点赞5

 评论1

 分享

 已收藏7

 举报

关注

一键三连

看懂UML类图和时序图_ZHOUBEISI的博客-CSDN博客_modelio类的泛化

8-20

http://design-patterns.readthedocs.io/zh_CN/latest/read_uml.html_modelio类的泛化

UML序列图 - 示例总汇

Warren Lynch 的博客 8106

UML Sequence Diagrams是交互图，详细说明了如何执行操作。它们捕获协作环境中对象之间的交互。序列图是时间焦点...

©2020 CSDN 皮肤主题: 大白 设计师:CSDN官方博客 [返回首页](#)

关于我们 招贤纳士 广告服务 开发助手 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 网络110报警服务 中国互联网举报中心 家长监护 Chrome商店下载 ©1999-2020北京创新乐知网络技术有限公司 版权与免责声明 版权申诉



life_爱水的鱼

码龄7年 暂无认证

43 16万+ 5万+ 13万+ 
原创 周排名 总排名 访问 等级

1765 16 38 7 59
积分 粉丝 获赞 评论 收藏

TA的主页

私信

关注

搜博主文章



热门文章

传输层TCP和UDP的区别分析与应用场景

【转载】 24706

Windows下boost安装及其在VS2013中配置

14180

从摄像头或视频文件中读取视频流并显示

9851

点赞5

评论1

分享

已收藏7

举报

关注

一键三连

Linux硬链接和软链接（符号链接）

7820

分类专栏

	配置	6篇
	STL学习	3篇
	Qt基础	2篇
	理解C++	11篇
	LeetCode and Algorithm	18篇
	C++11	



最新评论

Windows下boost安装及其在VS2013中配置
student_w: 我配置的没有成功, 显示“无法
打开源文件iostream”.....请问博主知道! ...

Windows下boost安装及其在VS2013中配置
student_w: 我配置的没有成功, 显示“无法
打开源文件iostream”.....请问博主知道! ...

传输层TCP和UDP的区别分析与应用场景...
山城忙碌人: 非常不错, 值得学习

当无符号数与有符号数一起玩耍时
xsw104205: 举个例子: a=-12,b=6; 分别用
方法一、方法二判断。方法一会走溢出 ...

看懂UML类图和时序图【转载】
冯浩月: 很详细的博文, 谢谢博主的分享 ..

 点赞5 评论1 分享 已收藏7 举报[关注](#)[一键三连](#)

从一堆（大量的）无序不重复数组成的数组
中寻找一指定数出现的位置

OpenCV中遍历图像与邻域操作

OpenCV 三种图像遍历方法

2016年 14篇

2015年 60篇

2014年 1篇

目录

[前言：](#)

看懂UML类图和时序图

从一个示例开始

类之间的关系

1：泛化关系(Generalization)

2：实现关系(Realization)

3：聚合关系(Aggregation)

4：组合关系(Composition)

5：关联关系(Association)

6：依赖关系(Dependency)

UML类图小结：

时序图

时序图

时序图元素

 点赞5

 评论1

 分享

 已收藏7

 举报

关注

一键三连