

西工大计算机学院高级语言程序设计

实 验 报 告

学号： 2020303245 姓名： 夏卓 班级： 10012006

课程名称	高级语言程序设计	实验课次	UML 第 3 次
实验项目	实验七 包图	实验时间	2020/5/13
指导老师评议	<p>成绩评定：（满分为 100 分）</p> <p>指导教师评语：</p> <p>指导教师签名：</p>		
实验内容（要求、算法、步骤和方法）	请参见 实验说明书		
实 验 结 果	<p>具体代码请参见附件中电子版的 java 源程序。源程序的简要说明如下：</p> <hr/> <p>UML 实验： <u>包图.mdl</u> 说明： B 中 B、C、D 之间存在循环依赖关系，故需要从 B 包中提取出与 D 有依赖关系的类组成新类，然后让 D 依赖与此新类以消除循环依赖。</p> <hr/> <p>实验结果：(包括运行结果，UML 图，以及实验要求中需要书面回答的内容)</p> <p>1. 包属于 UML 中的何种事物？有何作用？ 包是将多个元素组织为语义相关的组的通用机制。对于复杂系统进行建模时，经常需要处理大量的类、接口、组件、节点和图，这时就有必要将这些元素进行分类，即把语义相近并倾向于同一变化的元素组织起来加入同一个包中，以方便理解和处理整个模型。</p> <p>2. UML 中包的表示方法有哪几种？ 包图中包的基本模型元素包括包的名称、包的可见性、包的构造型等，它们在 UML 中都有不同的表示方式。</p> <p>(1) 包的命名。在 UML 中，包的标准形式是用两个矩形进行表示的，</p>		

即一个小矩形(标签)和一个大矩形,包的名称位于大矩形的中间,它有两种形式:简单名和路径名。其中简单名仅包含一个名称字符串;路径名是以包处于的外围包的名字作为前缀并加上名称的字符串。

(2) 包的可见性。包有 private, protected 和 public 三种可见性。private 定义的私有元素对包外部元素完全不可见;protected 定义的被保护的元素只对那些与包含这些元素的包有泛化关系的包可见;public 定义的公共元素对所有引入的包以及它们的后代都可见。

(3) 包的构造型。在 Rose 中,支持四种包的构造型:业务分析模型包、业务设计包、业务用例模型包和 CORBAModule 包。

3. 包与包之间存在哪几种关系?

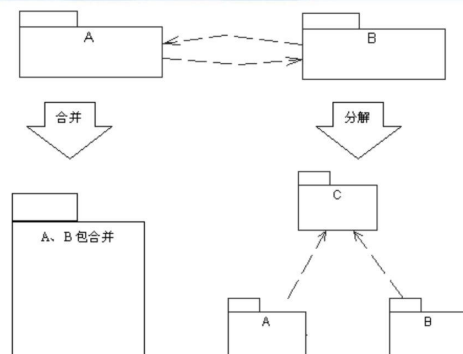
包与包之间的关系总体可以概括为依赖关系和泛化关系。包间的依赖关系可以分为很多种,如实现依赖、继承依赖,访问和引入依赖等。两个包之间的依赖关系通常是指这两个包所含的模型元素之间存在着一个或多个依赖。包之间的泛化关系与对象图之间的泛化关系十分类似,对象类之间的泛化的概念和表示在此大都可以使用。如果两个包之间存在由泛化关系,就是指其中的特殊性包必须遵循一般性包的接口。

4. 包与包之间的依赖关系有何特点?

包间依赖关系的存在表示存在一个自底向上的方法(一个存在声明),或存在一个自顶向下的方法(限制其他任何关系的约束),对应的包中至少有一个给定种类的依赖关系的关系元素。这是一个“存在声明”,并不意味着包中的所有元素都有依赖关系。对于由对象类组成的包,如果两个包的任何对象类之间存在着任何一种依赖,则这两个包之间就存在着依赖关系。包的依赖关系同样也是使用一根虚箭线表示,虚箭线从依赖源指向独立目的包。

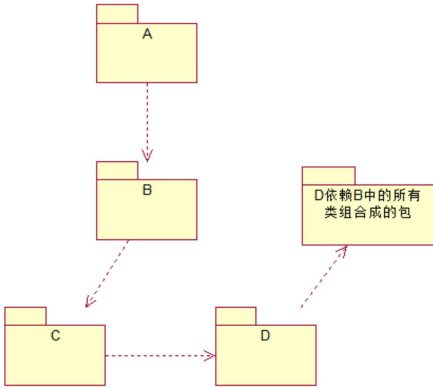
5. 如何避免包与包之间出现循环依赖关系? 各举一例。

- (1) 合并两个包;
- (2) 从两个包中提取公共元素,把他们封装为第三个包,原来的两个包依赖于新包。



6. 建立包图应注意哪些问题?

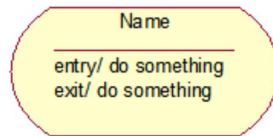
- (1) 包的命名要简单、具有描述性

	<p>(2) 避免包间的循环依赖</p> <p>(3) 包依赖应该反映内部关系</p> <p>(4) 建立模型时包的嵌套不宜过深，嵌套的层数一般以 2 层到 3 层为宜</p> <p>7. 选 C</p> <p>包之间的依赖关系是引入依赖，而引入依赖是没有传递性的，一个包的输出不能通过中间包被其他的包引入。因此 package1 与 package3 之间没有依赖关系，根据依赖的单向性可知，对输出包的修改，只需检查引入包中的元素是否需要做相应修改即可，故 C 正确。</p> <p>8. A 正确，B 错误</p> <p>因为 B 中 B、C、D 之间存在循环依赖关系，故需要从 B 包中提取出与 D 有依赖关系的类组成新类，然后让 D 依赖与此新类</p>  <pre> graph TD A[A] -.-> B[B] B[B] -.-> C[C] C[C] -.-> D[D] D[D] -.-> B[B] D[D] -.-> New["D依赖B中的所有类组成的包"] New --> B[B] </pre>
<p>实 验 心 得 体 会</p>	<p>1. 实验中遇到的问题以及解决方法</p> <p>最开始对包之间的依赖关系掌握不熟，不知道如何消除循环依赖，后来翻阅 ppt 并结合书上相关内容，明白了可以通过合并包或者提取公共元素，把他们封装为第三个包的方法消除循环依赖，同时直到了包之间的依赖关系没有传递性。</p> <p>2. 实验体会</p> <p>通过本次实验，我了解了包与包之间的各种依赖关系的区别，掌握了建立包与包之间依赖关系的方法，能够通过提取公共元素，把他们封装为第三个包的方法消除循环依赖。</p>

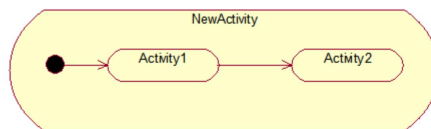
课程名称	高级语言程序设计	实验课次	UML 第 3 次
实验项目	实验八 活动图	实验时间	2020/5/13
指导老师评议	成绩评定：（满分为 100 分） 指导教师评语： 指导教师签名：		
实验内容（要求、算法、步骤和方法）	请参见 实验说明书		
实验结果	<p>具体代码请参见附件中电子版的 mdl 源程序。源程序的简要说明如下：</p> <hr/> <p>UML 实验： 网上购物.mdl 说明： 用户使用支付宝进行网上购物的活动图包括： 用户浏览购物网页-->挑选商品加入购物车-->勾选待购买商品-->选择支付宝支付-->若用户取消订单或订单超时则订单取消，否则用户输入密码，当密码正确时购物完成。 网上书店.mdl 参与交互的泳道有用户、系统和商家。用户下订单后系统会生成送货单，同时当用户付款完毕后，商家收到订单，根据订单发货，当用户收到书籍后订单完成。</p> <hr/> <p>实验结果：(包括运行结果，UML 图，以及实验要求中需要书面回答的内容)</p> <p>1. 什么是活动？UML 中如何表示活动？ 活动是某件事情正在进行的状态，它既可以是现实生活中正在进行的某一项工作，也可以是软件系统某个类对象的一个操作。在 UML 中可以利用活动图表示活动，活动用圆角椭圆表示。</p> <p>2. 活动图中包括哪些元素？分别如何表示？ UML 活动图中包含的图形元素有 动作状态：使用平滑的圆角矩形表示，动作状态表示的动作写在矩形内部。</p>		



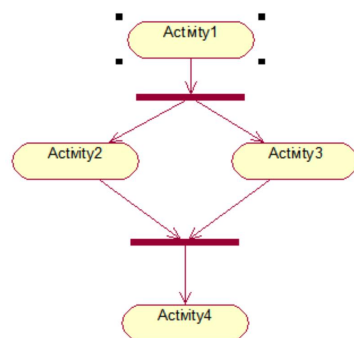
活动状态：同样使用平滑的圆角矩形表示，但是它可以在图标中给出入口动作和出口动作等信息。



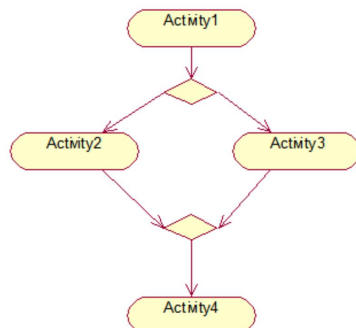
组合状态：组合活动是一种内嵌活动图的状态，其中可以包含不同的活动。



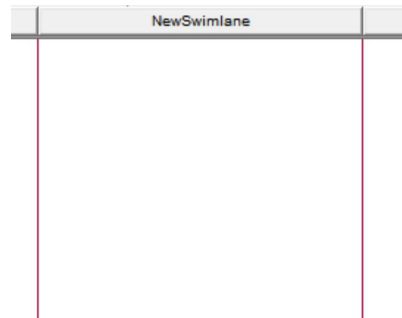
分叉与汇合：分叉和汇合在 UML 中都用粗黑线表示。分叉具有一个输入转换，两个或者多个输出转换；汇合相反，具有两个或多个输入转换，一个输出转换



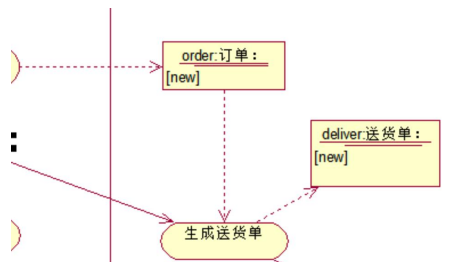
分支与合并：在 UML 中，分支与合并都是用空心的菱形表示。分支有一个输入箭头和两个输出箭头；合并有两个输入箭头和一个输出箭头。



泳道：在活动图中，每个泳道通过垂直实线与它的邻居泳道相分离。泳道上方是泳道的名称，不同泳道中的活动既可以顺序执行也可以并发执行。



对象流：活动图中对象用矩形表示，其中包含带下划线的类名，在类名下方的中括号中则是状态名，表明了对象此时的状态。用带箭头的虚线表示对象流。

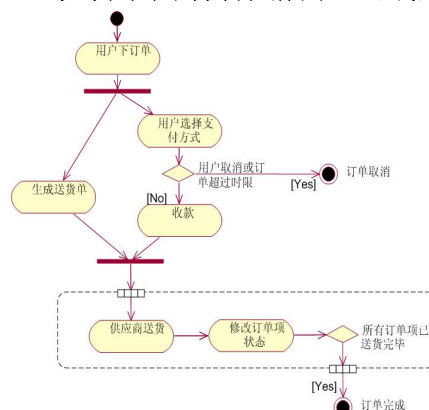


3. 分岔和分支有什么区别, 分别适用于什么样的建模目标?

分支的每个路径的监护条件应该是互斥的，只有一条路径的转换会被激发，常用于根据不同的监护条件选择相应的分支执行；而分叉用来表示将一个控制流分成两个或者多个并发运行的分支，常用于处理不同对象同步执行的并发控制流建模。

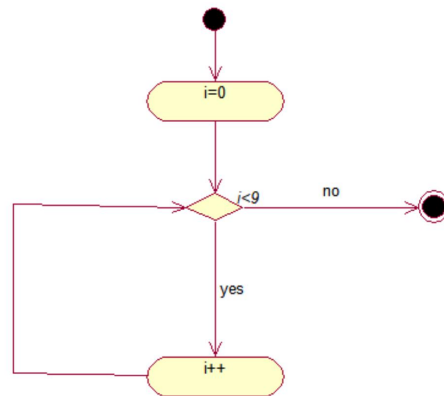
4. 活动图中监护条件和改变时间分别有何作用？两者间的区别是什么
监护条件与改变事件都是用来控制活动的处理过程，控制系统的行为。监护条件是决定执行与否，改变时间是决定什么时候执行。

5. 说明下面活动图所表达的信息

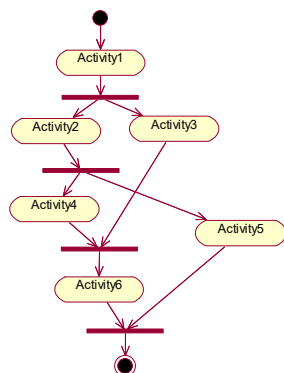


用户下订单后，系统生成送货单，同时用户选择支付方式，若用户取消或订单超时时限，则订单取消，否则进行收款，当送货单生成完毕且收款结束后控制流汇合，进入循环处理过程：供应商送货后修改订单项状态，直到所有订单项全部送货完毕，退出循环，订单完成。

6. 用活动图表示 for (i=0; i<9;i++) 循环



7. 对于如下图所示的活动图，最大可能的并发线程数是几个？

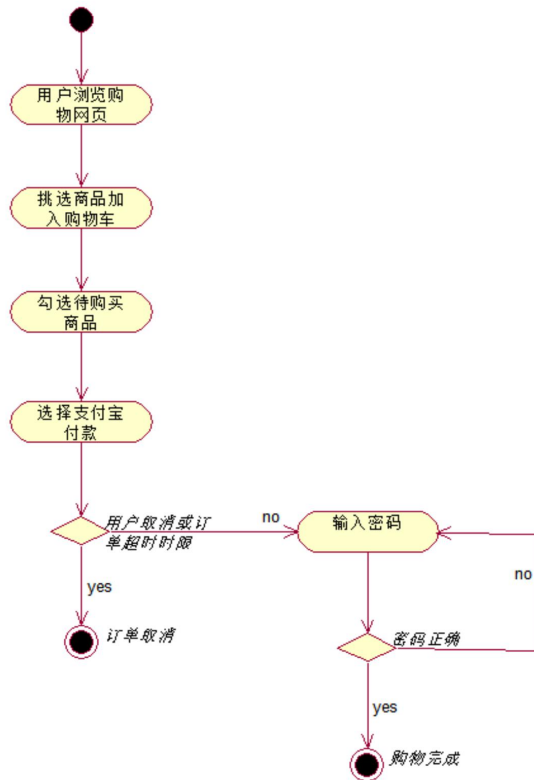


最大可能的并发线程数是 5 个，线程 1 在分叉处创建了线程 2 和 3，之后线程 2 又创建了线程 4 和 5，接着线程 3 和 4 汇合到达活动六，最后线程 5 和 3、4 汇合后的线程再次汇合。故最大并发线程数为 5 个。

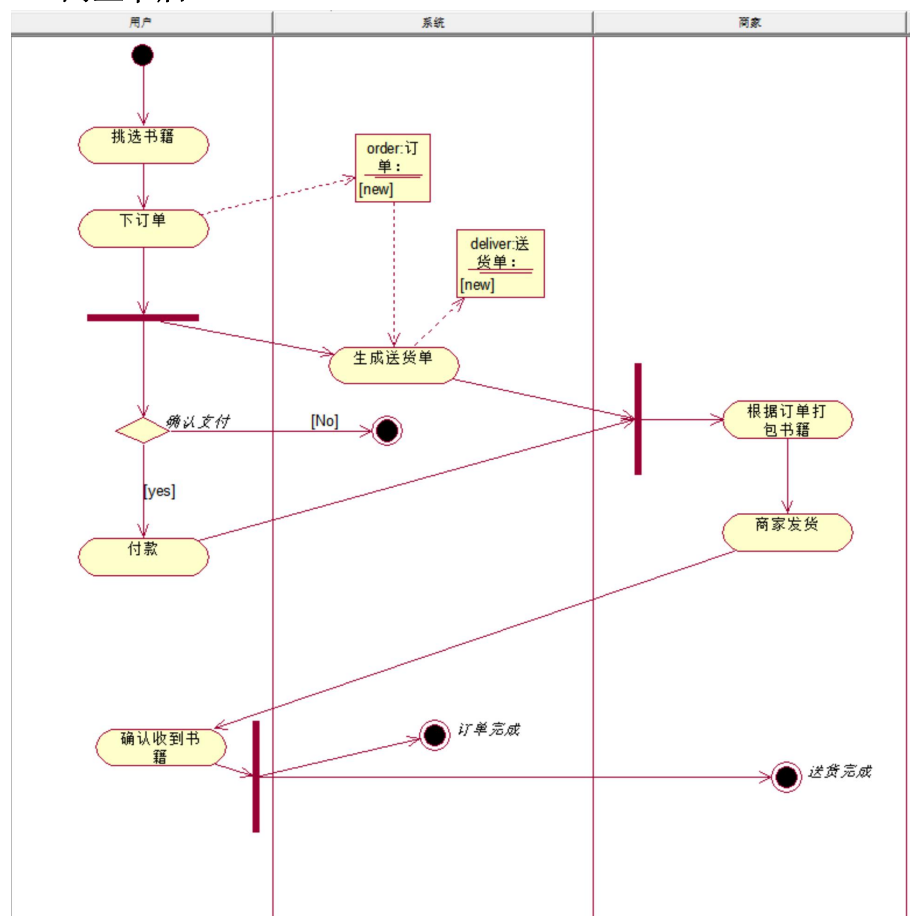
8. 总结 UML 在软件工程中的作用以及使用 UML 建模的必要性

UML 是一种功能强大的, 面向对象的可视化系统分析的建模语言, 它的各个模型可以帮助开发人员更好地理解业务流程, 建立更可靠, 更完善的系统模型. 从而使用户和开发人员对问题的描述达到相同的理解, 以减少语义差异, 保障分析的正确性。建立模型可以帮助开发者更好地了解正在开发的系统。便于开发人员展现系统, 提供指导开发人员构造系统的模板, 记录开发人员的决策。 每个项目可以从建模中受益。甚至在自由软件领域, 模型可以帮助开发小组更好地规划系统设计更快地开发。

9. 网上购物系统





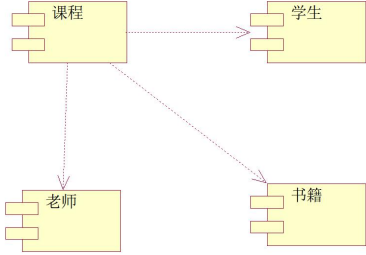
10. 网上书店



实 验 心 得 体 会	<p>1. 实验中遇到的问题以及解决方法</p> <p>一开始对活动图中的分支合并与分叉汇合区分不清楚，后来通过翻看书籍并查阅 ppt 后直到二者的区别在于，分支只会根据监护条件选择一条控制流程进行，而分叉则是所有输出端并发执行。合并是多种不同情况的合并，而汇合是所有并发流程的同步。</p> <p>2. 实验体会</p> <p>通过本实验我了解了活动图中个元素表示的意义和表示方法，掌握了分支、监护条件、分岔、汇合和泳道的概念，能够通过给定的情形绘制出活动图。</p>
----------------------------	---

课程名称	高级语言程序设计	实验课次	UML 第 3 次
实验项目	实验九 组件图	实验时间	2020/5/13
指导老师评议	<p>成绩评定：（满分为 100 分）</p> <p>指导教师评语：</p> <p>指导教师签名：</p>		
实验内容（要求、算法、步骤和方法）	请参见 实验说明书		
实 验	<p>具体代码请参见附件中电子版的 mdl 源程序。源程序的简要说明如下：</p> <hr/> <p>UML 实验： 构件图.mdl 说明：</p>		

<p>结果</p>	<p>分析可知，课程给老师、学生和书籍提供了相应的接口。</p> <hr/> <p>实验结果:(包括运行结果, UML 图, 以及实验要求中需要书面回答的内容)</p> <p>1. 什么是组件? 组件有何作用? 组件是定义了良好接口的物理实现单元。组件是系统中可替换的物理部件, 它包装了实现而且遵从并统一提供一组接口的实现。组件常用于对可分配的物理单元建模, 这些物理单元包含模型元素, 并具有身份表示和明确定义的接口。它所表现的是一种系统静态实现的结构, 能够帮助开发人员对系统组成达成一致的认识。</p> <p>2. 组件的要素是什么? 组件的要素包括: (1) 规格说明: 一个组件所提供服务的抽象描述。(每个组件都必须提供特定的服务) (2) 一个或多个实现: 组件是一种物理概念, 它必须被一个或多个实现所支持。 (3) 受约束的构造标准: 每一个组件在实现时必须遵从某种构造标准。 (4) 封装方法: 组件遵从的封装方法。 (5) 部署方法: 组件要运行, 必须先部署, 一个组件可以有多个部署。</p> <p>3. 组件和类的关系是怎样的? 它们有何异同之处? 组件在很多方面与类相同: 两者都有名称; 都可以实现一组接口; 可以参与依赖关系; 可以被嵌套; 可以有实例; 可以参与交互。 它们之间的区别: 类表示逻辑抽象(软件设计的逻辑组织和意图), 而组件表示物理实现(每个组件体现了系统设计中特定类的实现); 类可以直接拥有属性和操作, 而一般情况下, 组件只拥有只能通过其接口访问的操作。</p> <p>4. UML 中组件有哪些表示方法, 分别代表什么含义? UML 中可以使用不同图标表示不同类型的组件。他们包括: (1) 构件。可以用 ActiveX, Applet, Application, DLL, EXE 以及自定义构造型来指定构件的类型。 (2) 子程序规范和子程序体。子程序规范和子程序体是用来表示子程序的规范和实现体。子程序是一个单独处理的元素的包, 通常用它来代指一组子程序集。 (3) 主程序。主程序是指组织起来以完成一定目的的连接单元的集合。 (4) 数据库。在 Rose 中, 数据库也被认为是一种组件。 (5) 虚包。虚包是一种只包含对其他包所具有的元素进行引用的构件, 它被用来提供一个包的某些内容的公共视图。 (6) 包规范和包体。包规范和包体分别用于放置声明文件和实现文件。 (7) 任务规范和任务体。他们用来表示那些拥有独立控制线程的构件的规范和实现体。</p>
-----------	---

	<p>5. 组件中的接口有哪些？分别如何表示？ 组件中的接口包含导出接口和导入接口。表示方法如下：</p> <div style="display: flex; align-items: center; margin: 10px 0;"> <div style="margin-right: 20px;">  </div> <div>用无箭头的直线连接的接口表述导出接口</div> </div> <div style="display: flex; align-items: center; margin: 10px 0;"> <div style="margin-right: 20px;">  </div> <div>用虚线箭头指向接口，表示导入接口</div> </div> <p>6. 在一张基本组件图中，组件之间最常见的关系是什么？ 组件之间最常见的关系是依赖关系。</p> <p>7. 讨论组件图适用于哪些建模需求 组件图一般用于对面向对象系统的物理方面建模，显示了组成系统的构件之间的组织及其依赖关系。在以构件为基础的开发中，构件图为架构设计师提供了一个系统解决方案模型的自然形式，并且，它还能够系统在完成后允许一个架构设计师验证系统必须功能是由构件实现的，这样确保了最终系统将会被接受。</p> <p>8. 分析实验二(4)的选课系统类图，并绘制其组件图</p> <div style="text-align: center; margin: 20px 0;">  </div>
<p style="text-align: center;">实 验 心 得 体 会</p>	<p>1. 实验中遇到的问题以及解决方法</p> <p>一开始我并不知道组件与类图之间的区别的联系，对组件图的作用也不甚至明晰，不知道组件中的接口有哪些。后来通过上网查阅资料，知道了组件其实就是定义了良好接口的物理实现单元，它包装了实现而且遵从并统一提供一组接口的实现。组件中的接口包含导出接口和导入接口。</p> <p>2. 实验体会</p> <p>通过此次实验我了解了 UML 中引入组件的原因，知道了组件与类之间的异同，掌握了阅读和绘制组件图的方法。组件图所表现的是一种系统静态实现的结构，它能够帮助开发人员对系统组成达成一致的认识。</p>