第9章 包图

9．1 包图的概念

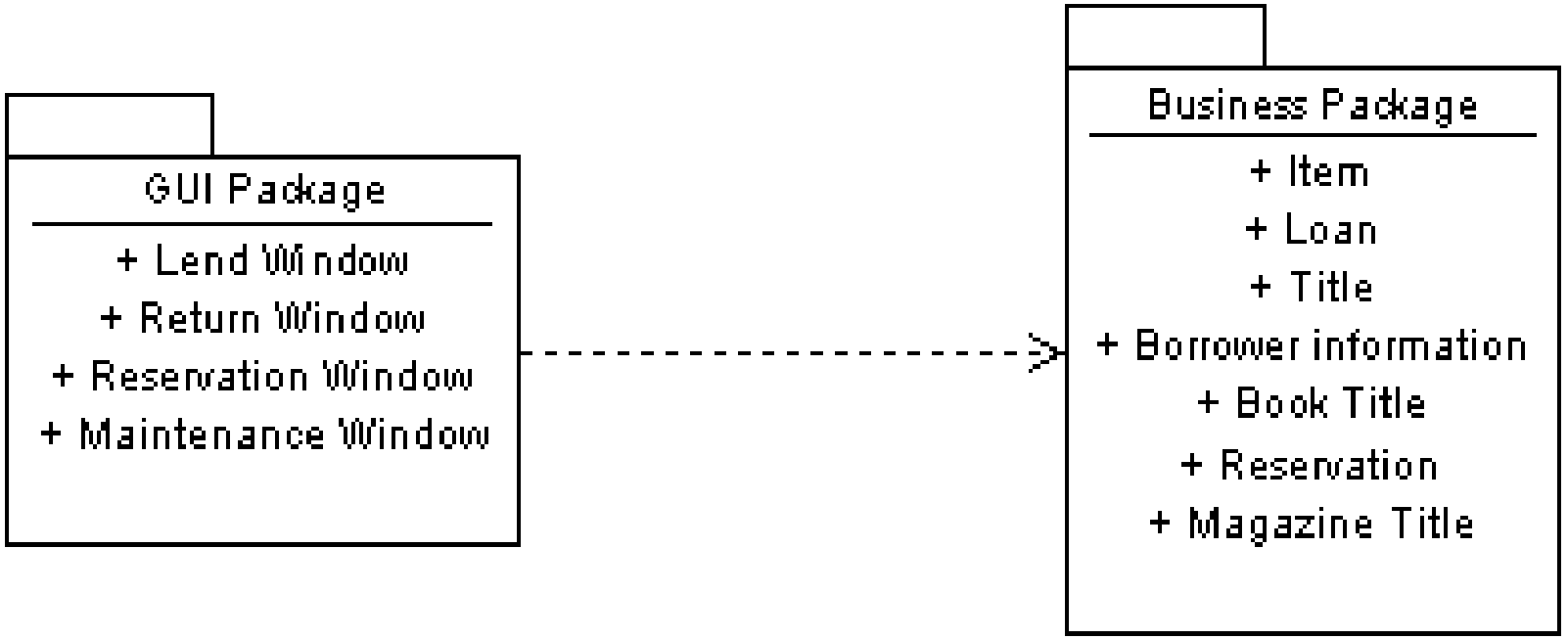
1．包

- 包就像一个“容器”，可用于组织模型中的相关元素，以便容易理解。
- 将相关的各种类型的模型元素组织成组的通用机制。
- 包的实例没有任何语义。
- 仅在建模时有意义，而不必转换到可执行的系统中。

9．1 包图的概念

2．包图

- 描述包及其关系的图。图**9-1**是一个典型的包图。
- 包图**(Package Diagram)** 是维护和控制系统总体结构的重要建模工具。



9．1 包图的概念

3．包的作用

- 对语义上相关的元素进行分组。
- 提供配置管理单元。
- 提供并行工作的单元。
- 提供封装的命名空间，同一个包中，其元素的名称必须唯一。

9．1 包图的概念

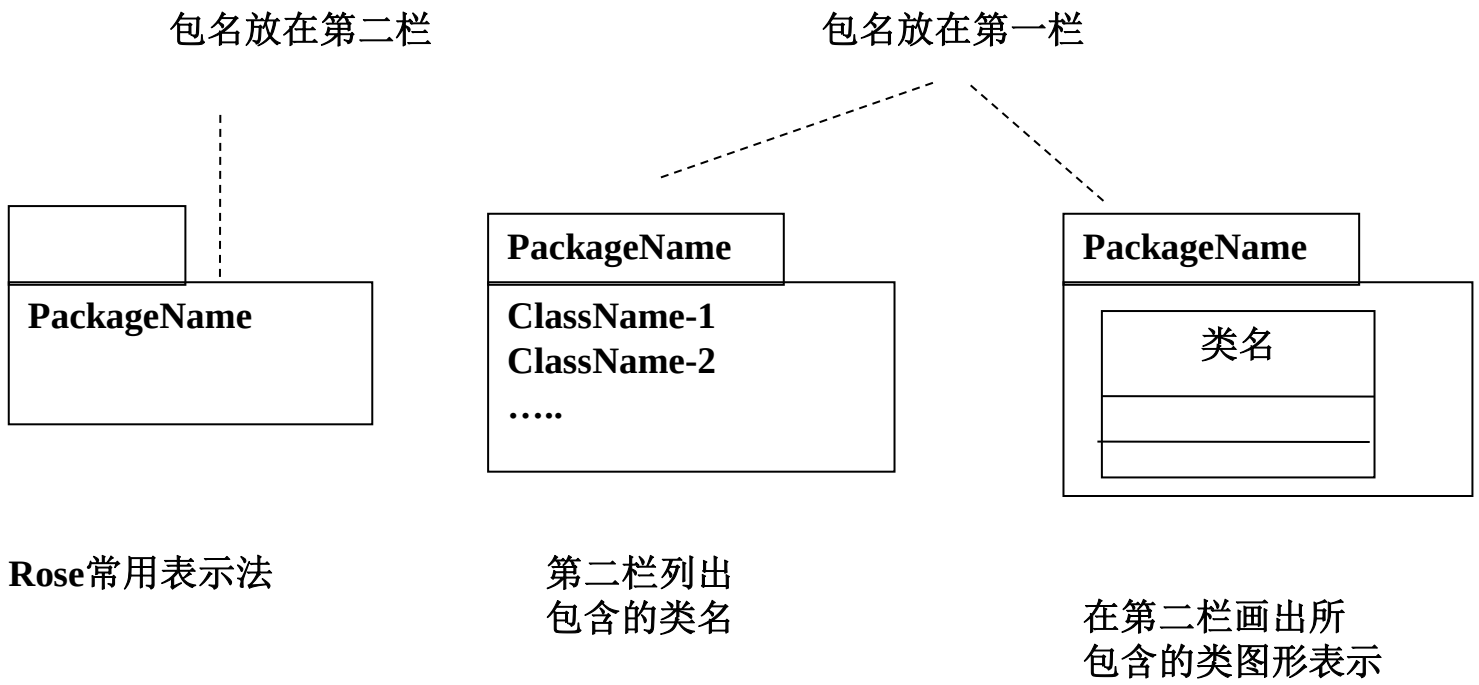
3．包中的元素

- 包中的元素：类、接口、组件、节点、协作、用例、图以及其他包。
- 一个模型元素不能被一个以上的包所拥有。
- 如果包被撤销，其中的元素也要被撤销。

9．2 包的表示

1．包的符号表示

- **UML**中，用文件夹符号来表示一个包。包由一个矩形表示，它包含**2**栏。
- 最常见的几种包的表示法，如图**9-2**所示：

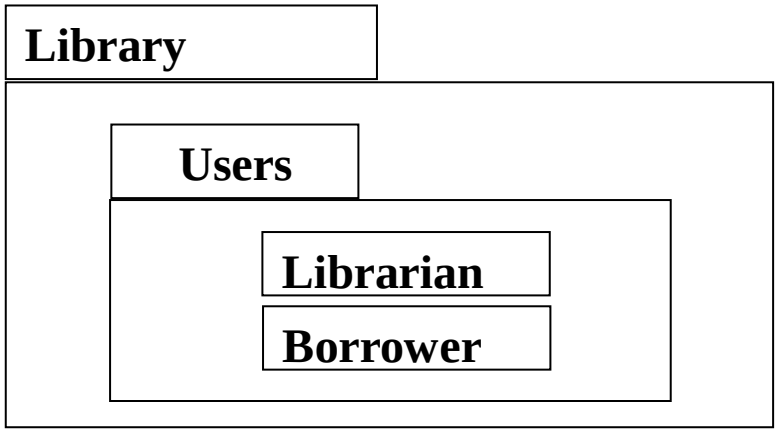


9．2 包的表示

1．包的符号表示

- 嵌套包及其表示

✓一个包可以包含其他的包；
✓嵌套包可以访问自身的元素；
✓应尽量避免使用嵌套包，一般2-3层最好。

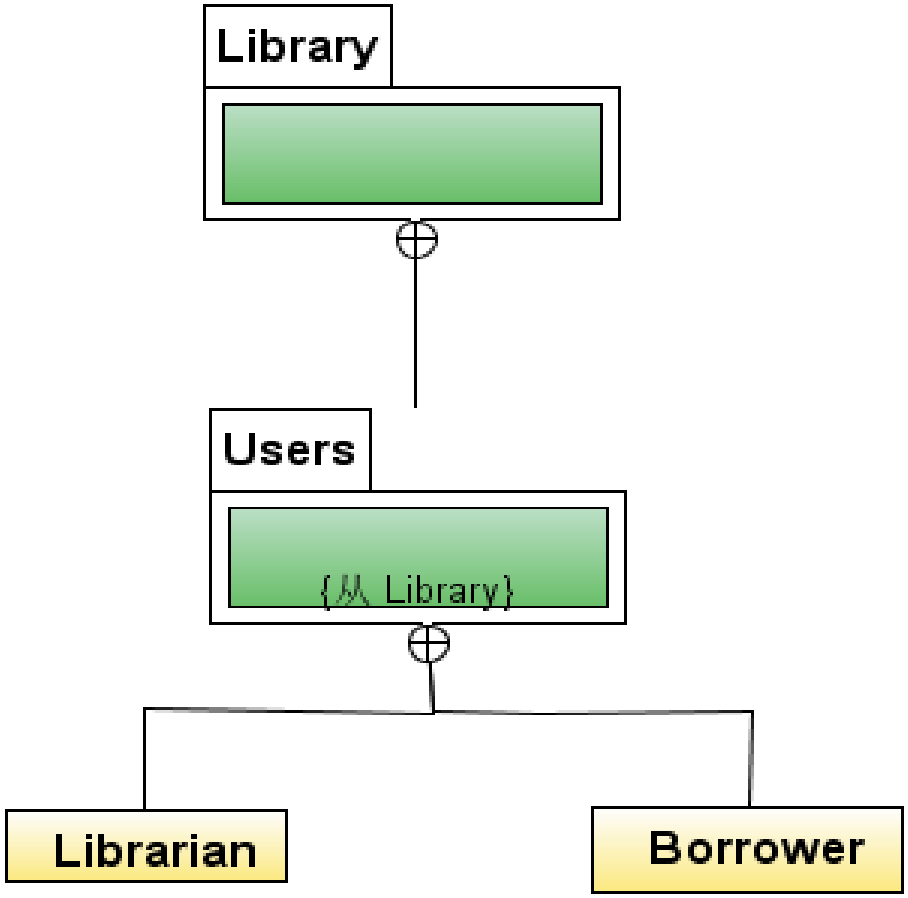


嵌套包

9．2 包的表示

1．包的符号表示

- 嵌套包及其表示

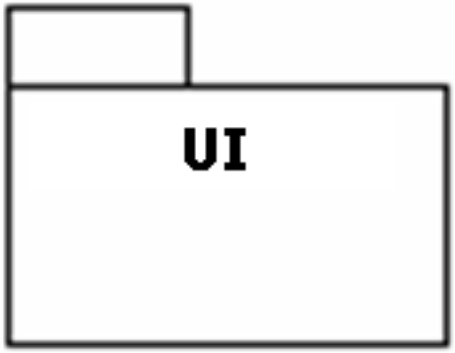


外部表示法

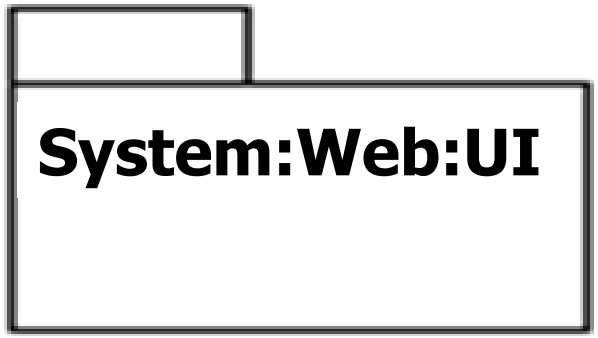
9．2 包的表示

2．包的名称

- 每个包必须有一个与其他包相区别的名称。
- 两种形式：简单名和路径。



简单名



路径名

System.Web.UI

9．2 包的表示

3．包的元素

- 在一个包中可以拥有各种其他元素，这是一种组成关系。
- 每一个包就意味着一个独立的命名空间，两个不同的包，可以具有相同的元素名。
- 在包中表示拥有的元素时，有两种方法：一种是在第二栏中列出所属元素名，一种是在第二栏中画出所属元素的图形表示。

9．2 包的表示

4．包的可见性

- 公有的（**public**）“＋”
- 受保护的（**protected**）“＃”

① 私有的（**private**）“—”

9．2 包的表示

4．包的可见性

包内元素的可见性控制了包外部元素访问包内部元素的权限。

可见性	含义	前缀符号
公有的 **Public**	此元素可以被任何引用该包的包中的元素访问。	**+**
受保护的 **Protected**	此元素可被继承该包的包中的元素访问。	**#**
私有的 **private**	此元素只能被同一个包中的元素访问。	**-**

9．3 包图中的关系

包之间可以有两种关系：

- 引用和访问依赖：在一个包中引入另一个包输出的元素

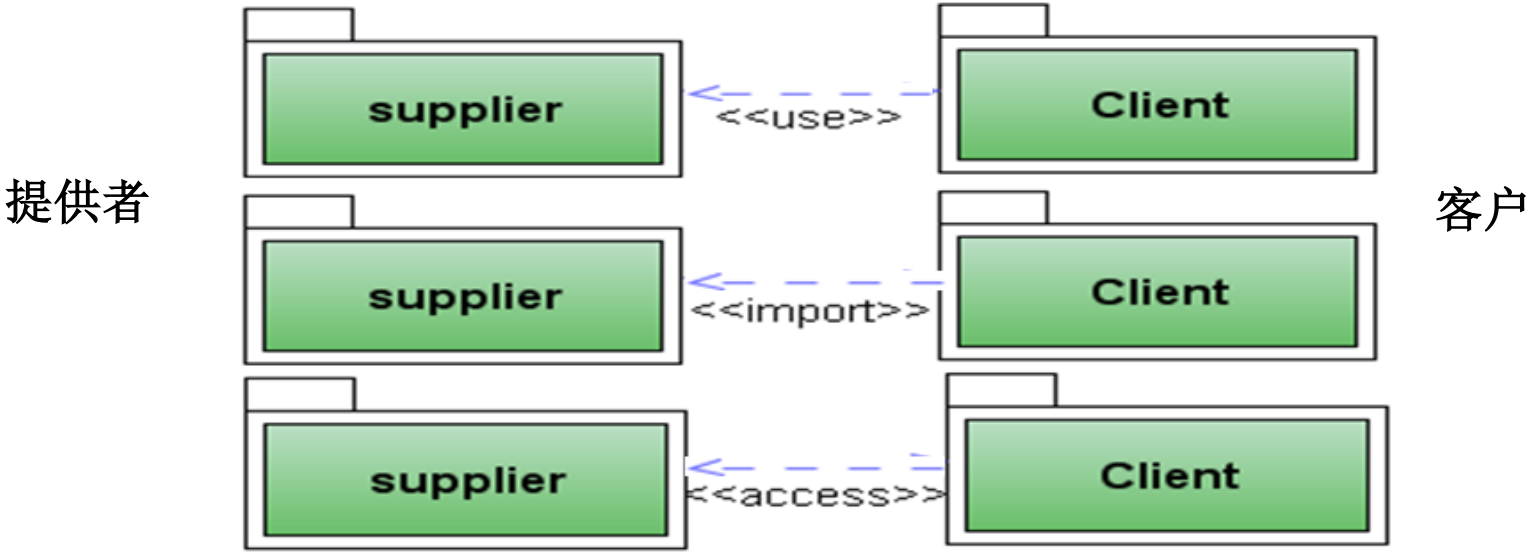
① 泛化：说明包的家族

9．3 包图中的关系

1. 依赖关系

- 使用关系《use》：说明客户包中的元素以某种方式使用提供者包的公共元素。
- 包含关系《import》：提供者包命名空间的公共元素被添加为客户包命名空间上的公共元素

① 访问关系《access》：提供者包命名空间的公共元素被添加为客户包命名空间上的私有元素。



9．3 包图中的关系

1．依赖关系

- ④ 跟踪关系《**trace**》：通常表示一个元素历史地发展成为另一个进化版本。

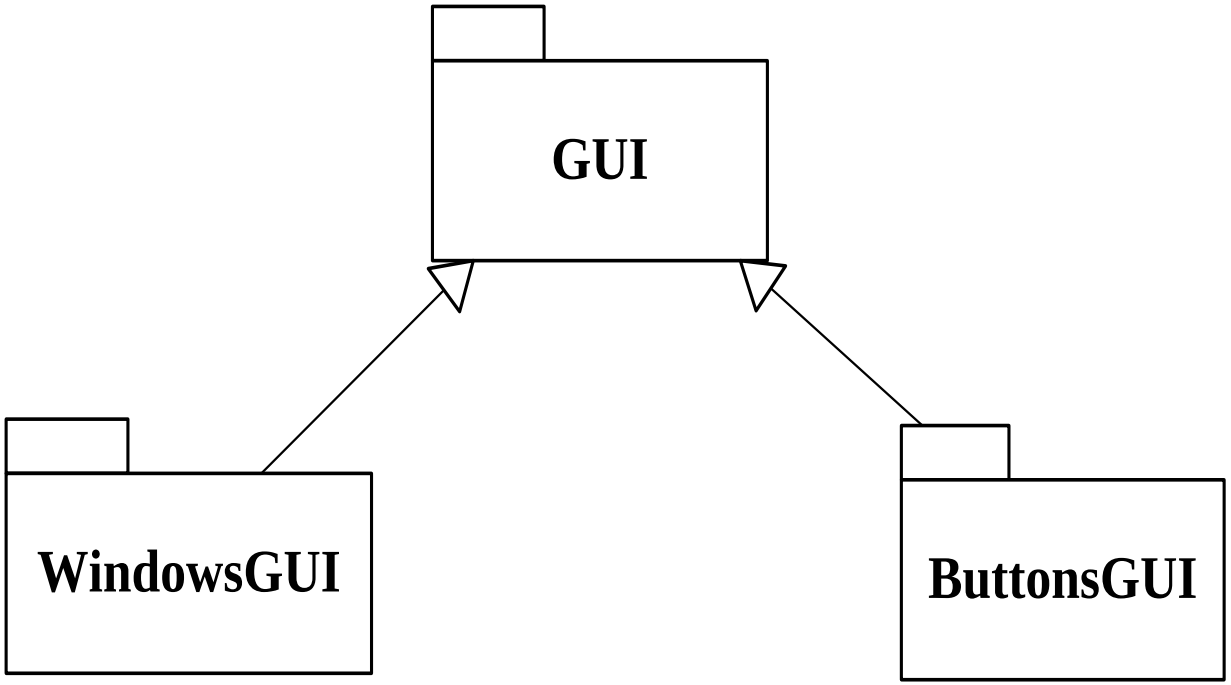
1. 分析模型是设计模型的元模型，元模型的元素进化为设计模型，这时指的是模型之间的关系，不是元素之间的关系。

元模型

9．3 包图中的关系

2．泛化关系

包间的泛化关系与类之间的泛化关系类似。



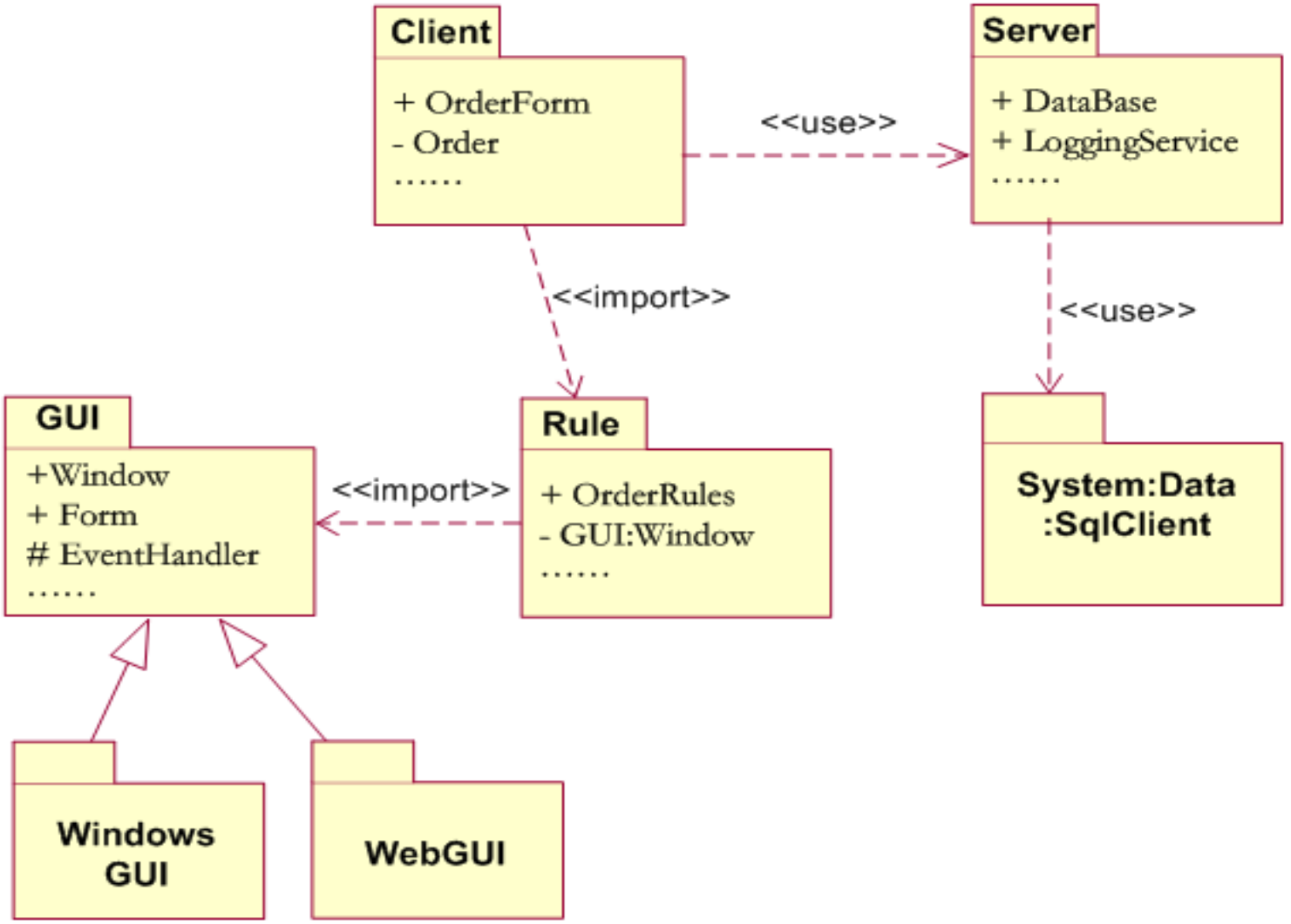
9．4 阅读包图

阅读包图的方法如下：

- 了解每个包的语义及其包含元素语义。
- 理解包间的关系。

① 找到依赖复杂的包，从最复杂的包开始阅读，然后依次是简单的包。

9．4 阅读包图


Client
+ OrderForm
- Order
……
<<use>>
Server
+ DataBase
+ LoggingService
……
<<import>>
<<use>>
GUI
+Window
+ Form
# EventHandler
……
<<import>>
Rule
+ OrderRules
- GUI:Window
……
System:Data
:SqlClient
Windows
GUI
WebGUI

举例

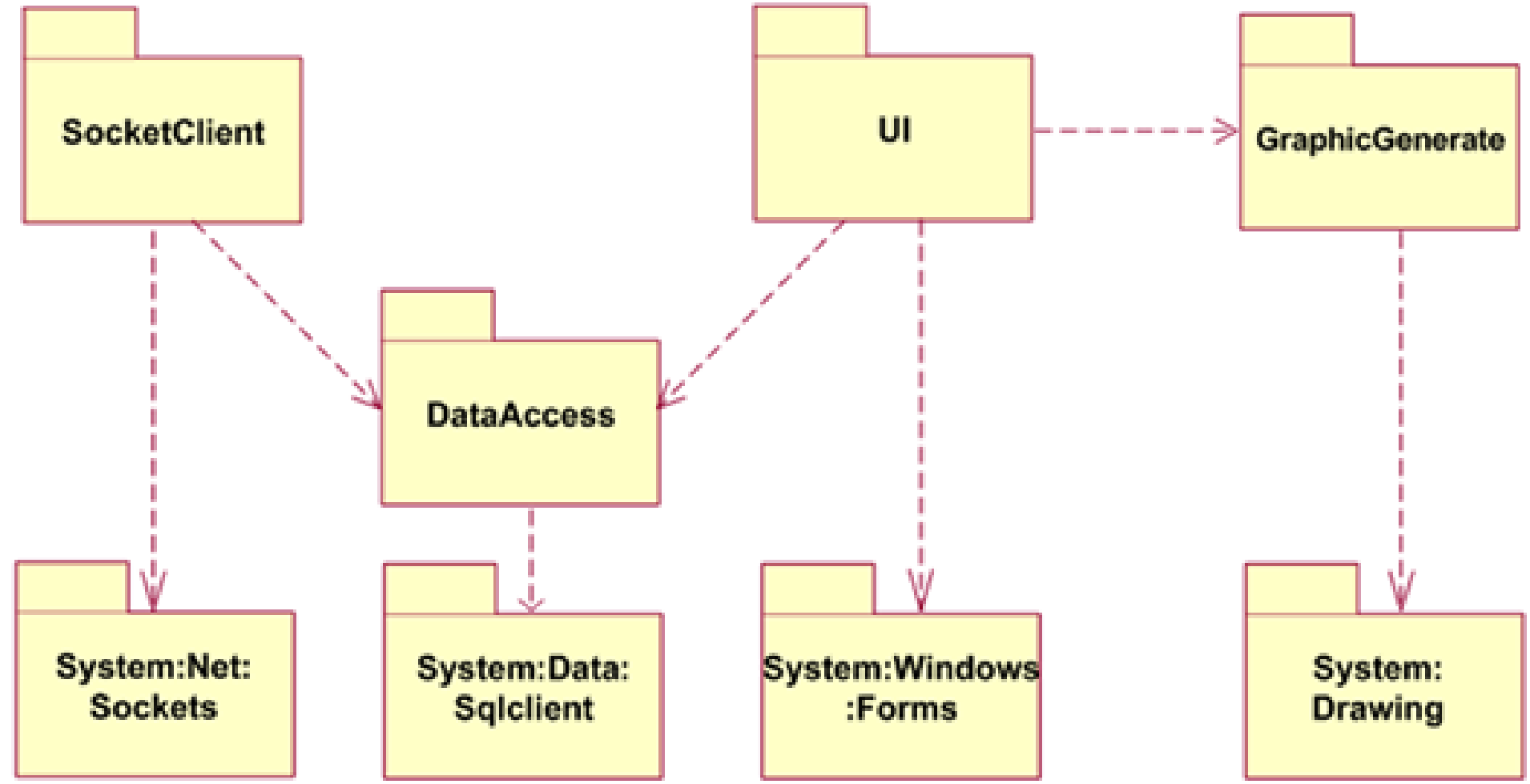
例子描述

- 分析系统工作流程：
  1）通过Internet连接到股票信息服务器，获取实时的股票信息，并存入数据库中。
  2）根据用户的输入和选择，从数据库中获取相应的信息，展现在屏幕中。
  3）在数据的展现过程中，将需要绘制大量的图表。

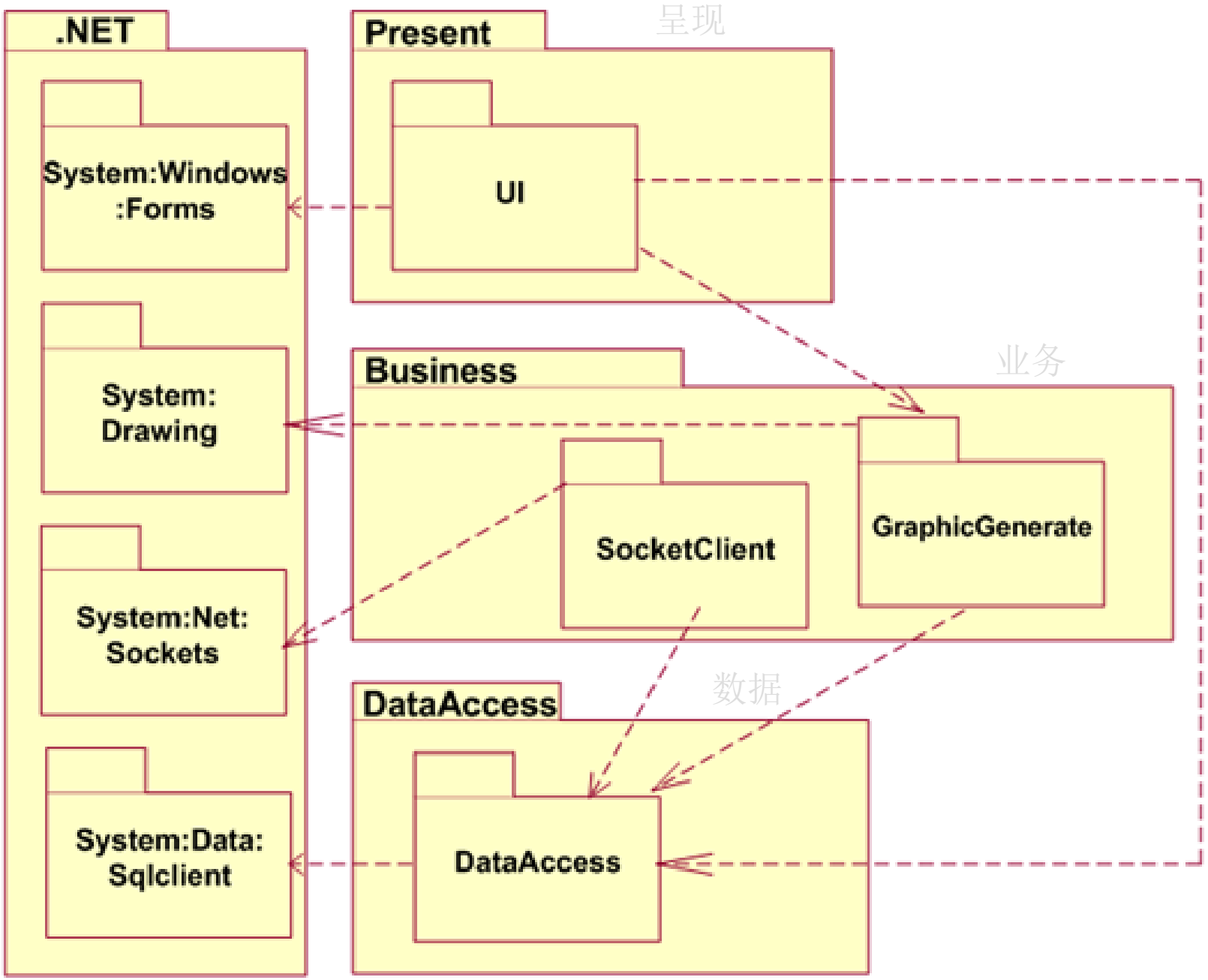
举例

根据功能模块组织包：

包	分析与功能	.NET支持包
SocketClient	负责连接Internet服务器，获取实时股票信息	System.Net.Sockets
DataAccess	负责从数据库读写实时股票信息	System.Data.Sqlclient
UI	负责响应用户输入和选择，并展现信息	System.Windows.Forms
GraphicGenerate	负责根据数据库的信息生成相应的图表	System.Drawing


SocketClient
UI
GraphicGenerate
DataAccess
System:Net:
Sockets
System:Data:
Sqlclient
System:Windows
:Forms
System:
Drawing

举例


.NET
System:Windows
:Forms
System:
Drawing
System:Net:
Sockets
System:Data:
Sqlclient
Present
呈现
UI
Business
业务
SocketClient
GraphicGenerate
DataAccess
数据
DataAccess

设计包的原则

在考虑如何对类进行分组并放入不同的包时，主要依据类之间的依赖关系进行分组。包中的类应该是功能相关的，在建包时，应把概念上和语义上相近的模型元素纳入一个包。依赖关系其实是耦合的一种体现，如果两个包中的类之间存在依赖关系，那么这两个包之间就有了依赖关系，也就存在耦合关系。好的设计要求体现高内聚、低耦合的特性。

设计包的原则

设计包时，应遵循以下原则：

- 重用等价原则
- 共同闭包原则
- 共同重用原则

① 非循环依赖原则

设计包的原则

- 重用等价原则
- ——把类放入包中时，应考虑把包作为可重用的单元。
- 共同闭包原则

——把那些需要同时改变的类放在同一个包中。

  - 若一个类的行为或结构的改变要求另一个类做相应的改变；
  - 删除了一个类后，另一个类成多余的；
  - 两个类之间有大量的消息发送。

设计包的原则

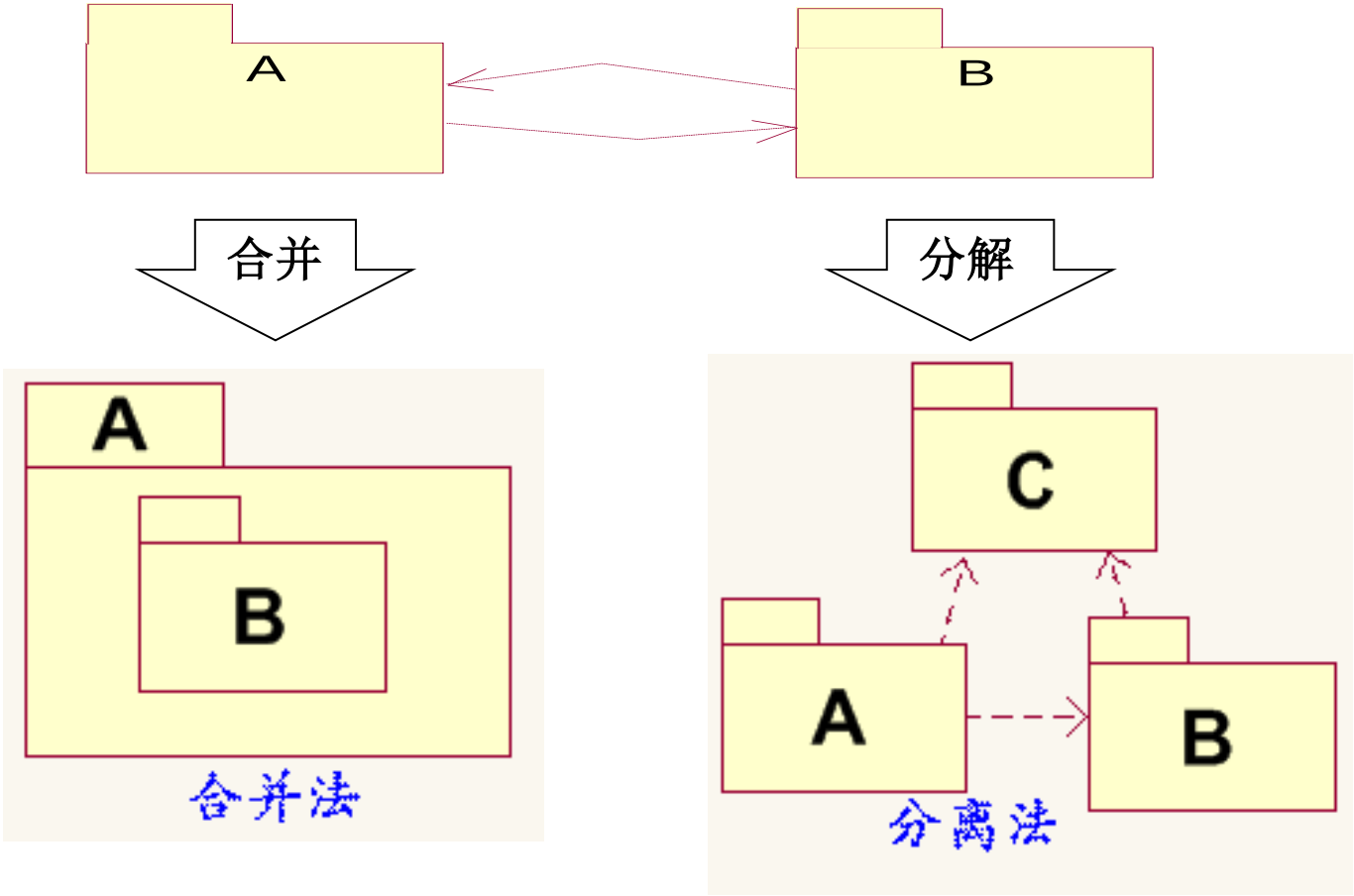
- 共同重用原则

——把不会一起使用的类不要放在同一个包中。

- 非循环依赖原则

——包之间的依赖关系不要形成循环。

设计包的原则



消除循环依赖的示例

9．4 创建包图

绘制包图的基本步骤如下：

① 分析系统的模型元素（通常是对象类），把概念上或语义上相近的模型元素归入同一个包。

② 对于每一个包，标出其模型元素的可视性，确定包内每个元素的访问属性，是公共、保护或私有。

③ 确定包与包之间的依赖联系，特别是“引入”关系。

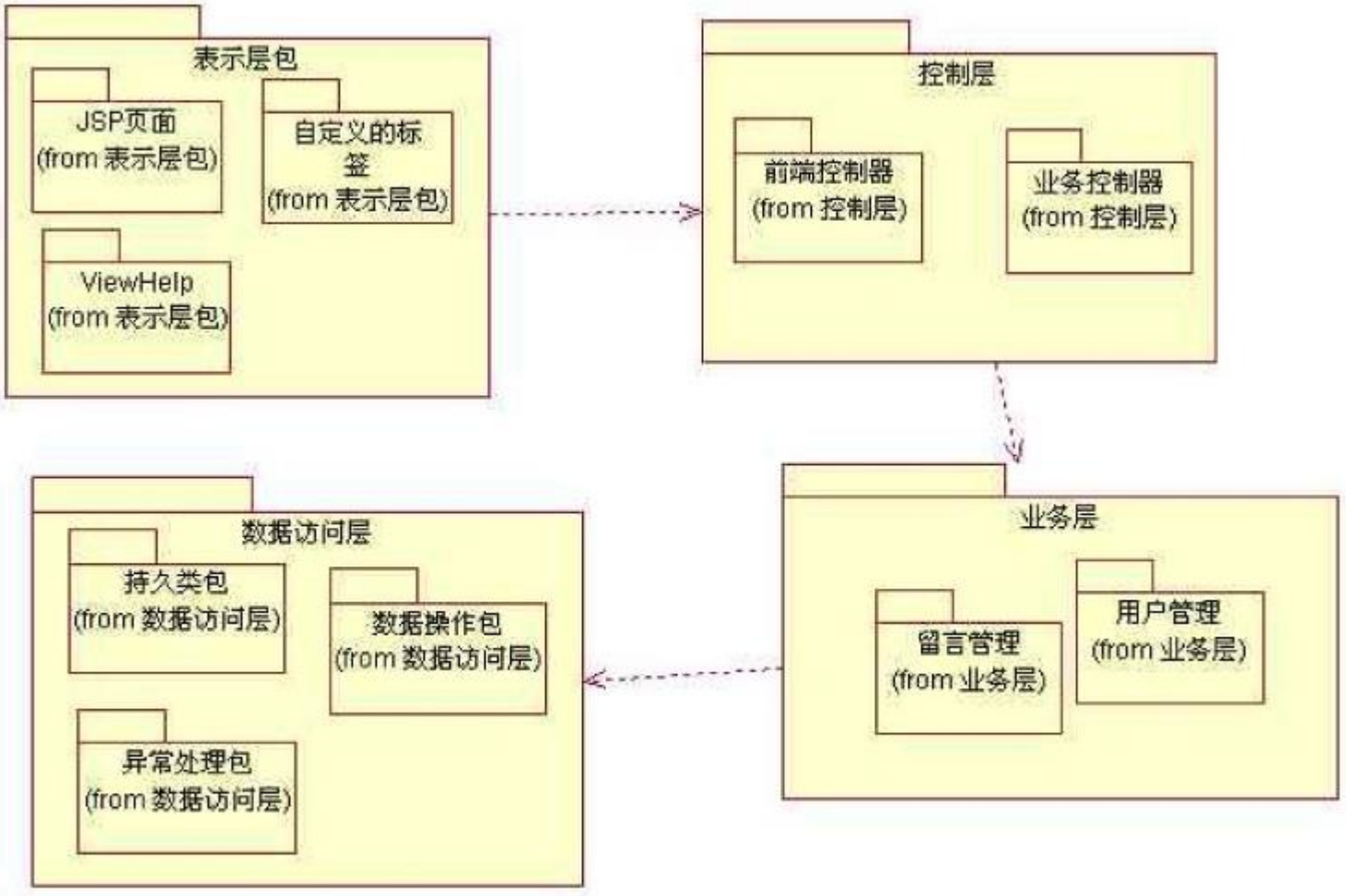
④ 确定包与包之间的泛化关系。

⑤ 绘制包图。

⑥ 对结果进行精化和细化。

9．4 包图示例

某Web系统包图：



小结

- 首先解释了几种常见的包图表示法之后，通过了一个简单的例子来说明包的可见性、依赖关系、泛化等概念；其次，概要地说明了五种包的构造型 。
- 说明如何寻找包、确定包之间的依赖关系，从而绘制了出一个表明软件体系结构的包图。