

# 包的概念

- 语义
- 表示
- 包之间的关系
- 对成组的元素建模策略



## 6.2.2 包

为了组织类目,控制信息组织和文档组织的复杂性,UML引入了术语-包。

### 6.2.2.1 语义

包是模型元素的一个分组。一个包本身可以被嵌套在其它包中,并且可以含有子包和其它种类(model element)的模型元素。

一个包元素对外的可见性,可以通过在该元素名字前加上可见性符号(+ : 公共的, - : 私有的, # : 受保护的)来指示:

- + : 对其他包而言都是可见的;
- : 对其他包而言都是不可见的;
- # : 对子孙包而言是可见的;

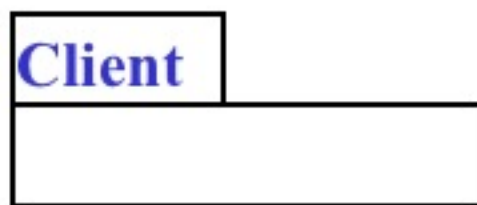


# 包的概念

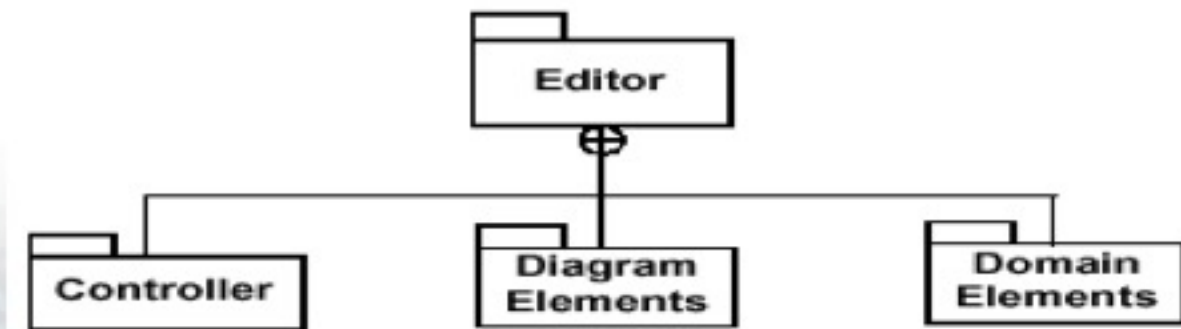


## 6.2.2.2 表示

①通常，在大矩形中描述包的内容，而把该包的名字放在左上角的小矩形中。



②可以把所包含的元素画在包的外面，通过符号 $\oplus$ ，将这些元素与该包相连。这时可把该包的名字放在大矩形中。



包拥有在其内所声明的模型元素,它们可以是类、接口、构件、协作、用况、节点,甚至可以是其他包。



北京大学

- 语义
- 表示
- 包之间的关系
- 对成组的元素建模策略

# 包的概念

- 语义
- 表示
- 包之间的关系
- 对成组的元素建模策略



## 6.2.2.3 包之间的关系

两种依赖：访问依赖和引入依赖。作用：使一个包可以访问和引入其它包。

注:包间的依赖通常隐含了各包中元素之间存在着的一个或多个依赖。

### (1) 引入依赖：《import》

从源包到目标包的引入依赖表明：目标包中有适当可见性的内容被加入到**源包的公共命名空间**中，这相当于源包对它们做了声明（即对它们的引用可不需要一个路径名）



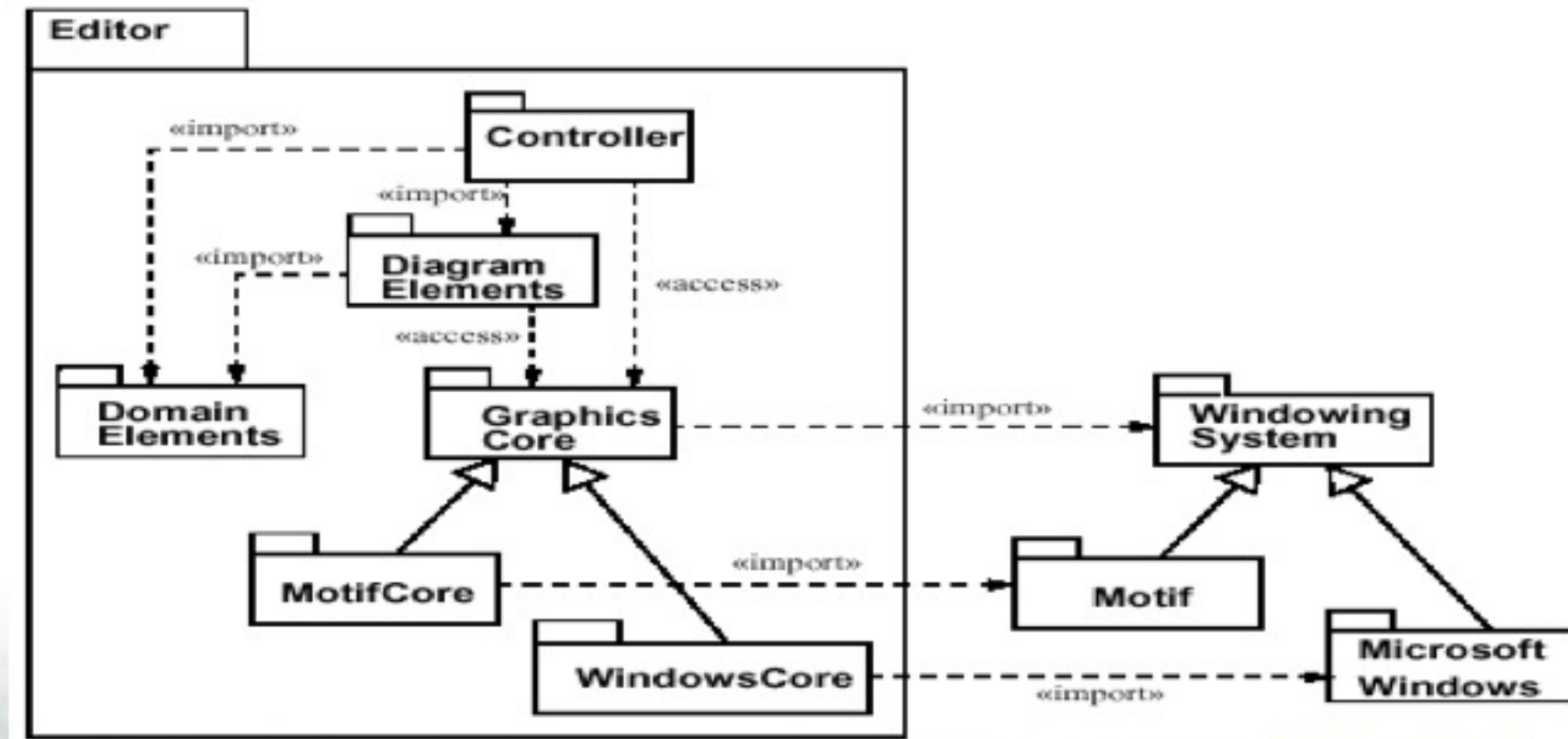


# 包的概念

- 语义
- 表示
- 包之间的关系
- 对成组的元素建模策略

## 引入: 《import》

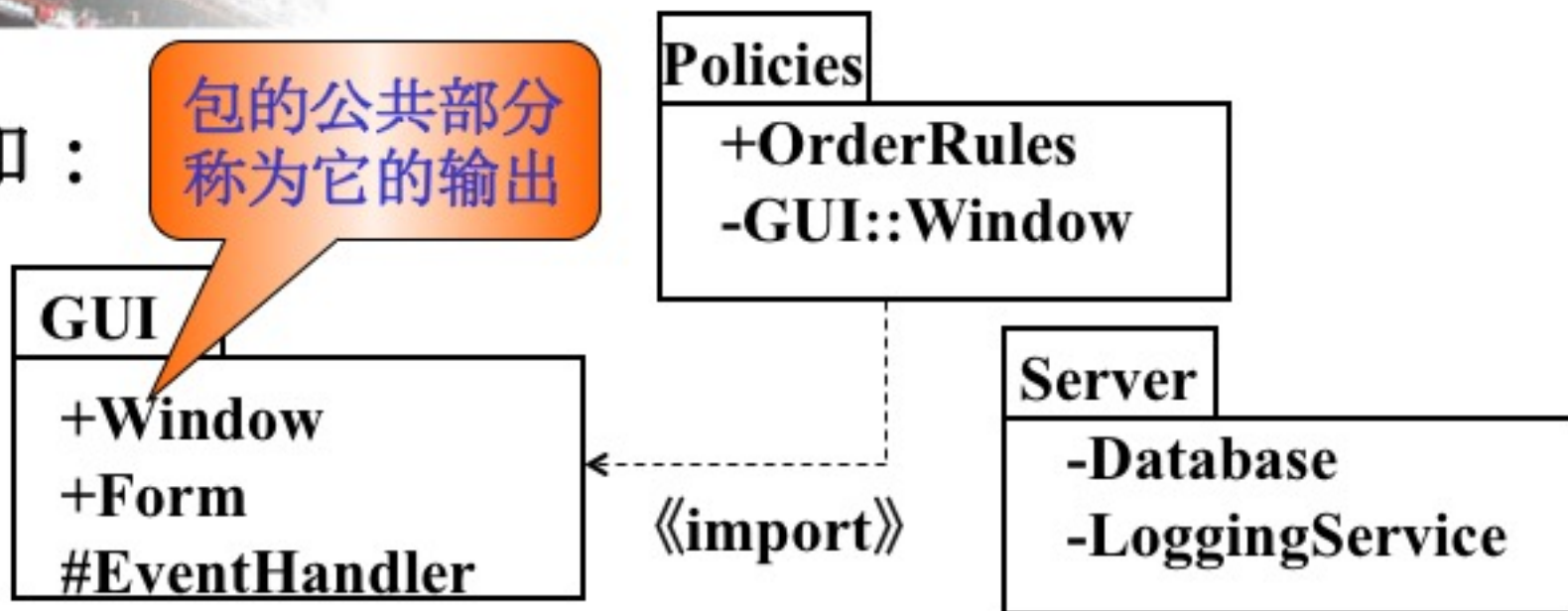
表示为从源包到目标包的一条带箭头的线段，并标记为《import》，如下图所示：



# 包的概念

- 语义
- 表示
- 包之间的关系
- 对成组的元素建模策略

例如：



注：包Policies引入包GUI，因此，对于类GUI::Window和类GUI::Form，包Policies的内容使用简单名Window和Form就能访问它们，然而，由于GUI::EventHandler是受保护的，因此它是不可见的。由于包Server没有引入包GUI，Server中的内容必须用限定名才能访问GUI的公共内容，如GUI::Window。由于Server的内容是私有的，GUI的内容无权访问Server中的任何内容，即使用限定名也不能访问它们。

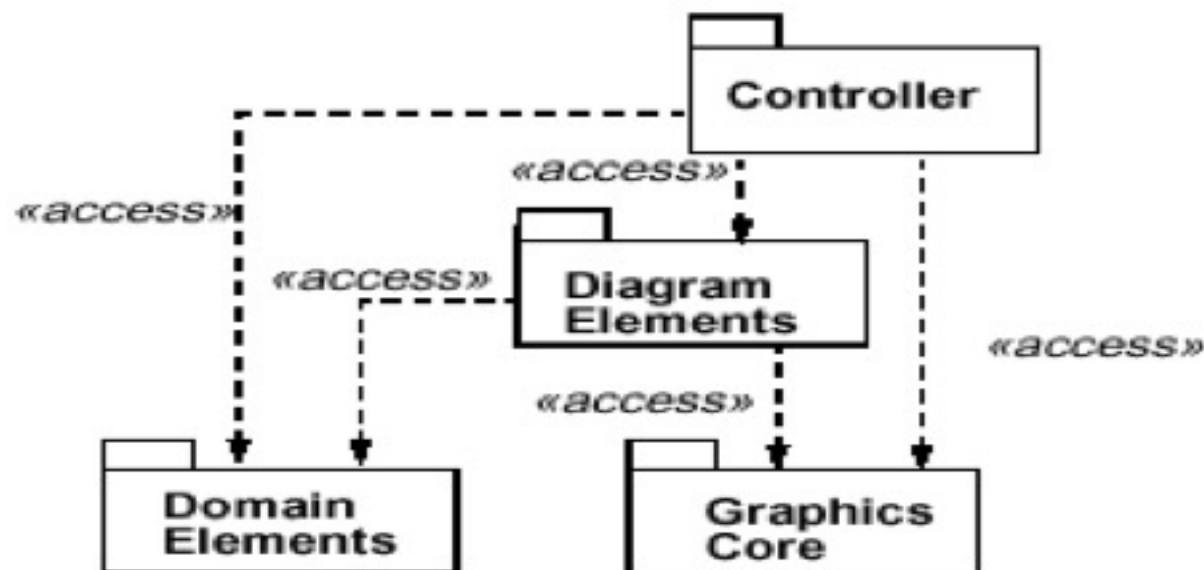


# 包的概念

- 语义
- 表示
- 包之间的关系
- 对成组的元素建模策略

## (2) 访问依赖: 《access》

从源包到目标包的访问依赖表示: 目标包中具有可见性的内容增加到源包的私有命名空间里 (即源包可以不带限定名来引用目标包中的内容, 但不可以输出之, 即如果第三个包引入源包, 就不能再输出已经被引入的目标包元素)。



注:如果在提出访问的那个包中还存在包, 那么嵌套在其中的包能得到与外层包同样的访问。





# 包的概念



## 6.2.2.4 对成组的元素建模策略:

- 语义
  - 表示
  - 包之间的关系
  - 对成组的元素建模策略
- 浏览特定体系结构视图中（如类图）的建模元素，找出概念或语义上相互接近的元素所定义的组块。
  - 把每一个这样的组块围在一个包内。
  - 对每一个包判别哪些元素要在包外访问，把这些元素标记为公共的，把所有其他元素标记为受保护的或私有的。
  - 用引入依赖显示地连接建立在其他包之上的包。
  - 在包的家族中，用泛化关系把特殊包连接到它们的较一般的包。

