

Sesión 20

Diagrama de Clases

Unidad 4

**Modelado del comportamiento
estático del sistema**

Mg. Gustavo G. Delgado Ugarte

Vista estática del sistema

- La vista estática es la base del UML
- Los elementos de la vista estática de un modelo son los conceptos significativos en una aplicación
 - Conceptos del mundo real
 - Conceptos abstractos
 - Conceptos de implementación
 - Conceptos de computación
 - Todo tipo de conceptos encontrados en el sistema

Vista estática del sistema

- La vista estática captura la estructura del objeto
- La vista estática incluye todo lo concerniente a las estructuras de datos tradicionales, así como la organización de las operaciones sobre los datos
 - Los datos y operaciones son cuantificadas en clases

Vista estática del sistema

- La vista estática describe las entidades de comportamiento como elementos de modelado discretos, pero no contiene los detalles del comportamiento dinámico
 - Los trata como elementos para ser nombrados, poseídos por las clases, e invocados
 - La ejecución dinámica es descrita por otras vistas que muestran los detalles internos de su faceta dinámica: diagramas de interacción

Vista estática del sistema

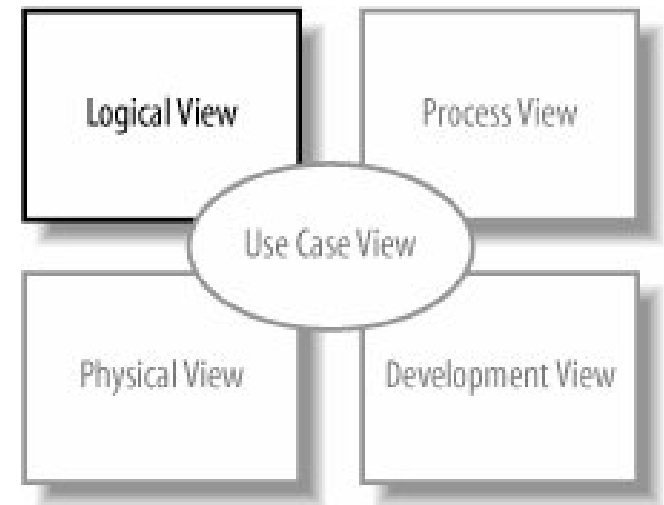
- Las vistas dinámicas requieren de la vista estática para describir las cosas que interactúan dinámicamente
 - No se puede decir *cómo* interactúa algo sin decir primero *qué* está interactuando
- La vista estática es la base sobre la que se construyen las otras vistas

Diagrama de Clases

- Las clases son el centro de cualquier sistema orientado a objetos, por lo tanto, se deduce que el diagrama UML más popular es el diagrama de clases
- La estructura de un sistema se compone de una colección de piezas referidas a menudo como objetos
- Las Clases describen los diferentes tipos de objetos que un sistema puede tener, y los diagramas de clases muestran las clases y sus relaciones

Diagrama de Clases

- Los casos de uso describen el comportamiento de un sistema como un conjunto de preocupaciones
- Las Clases describen los diferentes tipos de objetos que son necesarios dentro de un sistema para satisfacer esas inquietudes
- Las clases forman parte de la vista lógica del modelo



¿Qué es una Clase?

- Un guitarra es un ejemplo de un objeto
 - Tiene una identidad
- Sin embargo, la fábrica no hizo sólo una guitarra de un tipo
- La fábrica hace cientos de guitarras de un tipo o, para decirlo de otra manera, de una clase de guitarra
- Una clase es un tipo de algo
 - Se puede pensar en una clase como el modelo de objetos que se pueden construir

¿Qué es una Clase?

The Class



Used to define and construct



Instances, or objects,
of the class



¿Qué es una Clase?

- La descripción de una clase incluye dos tipos de información
 - La información del estado que contendrán los objetos de la clase
 - El comportamiento que van a soportar
- Esto es lo que diferencia a la OO de otras formas de desarrollo de sistemas
 - En la OO, el estado está estrechamente relacionado con el comportamiento y se combinan en las definiciones de clase, que luego se utilizan como los planos de los objetos que se pueden crear

¿Qué es una Clase?

- En el caso de la clase guitarra, el estado de la clase podría incluir información acerca de cómo están las cuerdas de la guitarra y en que condición está la guitarra
 - Estas piezas de información son los atributos de la clase
- Para completar la descripción, lo que necesitamos saber es lo que la guitarra puede hacer. Esto incluye comportamientos tales como afinación y tocar la guitarra
 - El comportamiento de una clase se describe como las diferentes operaciones que esta soporta

¿Qué es una Clase?

- Los atributos y operaciones son los pilares de la descripción de una clase
- Juntos, permiten a una clase describir un grupo de piezas dentro de un sistema que comparten características comunes
 - El estado es representado por los atributos de la clase
 - El comportamiento es representado por las operaciones de la clase

Diagrama de Clases

- Una clase en UML se dibuja como un rectángulo dividido en un máximo de tres secciones
 - La sección superior contiene el nombre de la clase
 - La sección central contiene los atributos o la información que contiene la clase
 - La última sección contiene las operaciones que representan el comportamiento que exhibe la clase
- Las secciones de los atributos y las operaciones son opcionales
 - Si las secciones de los atributos y las operaciones no se muestran, no implica necesariamente que están vacías, sólo que el esquema, es quizás más fácil de entender con la información oculta

Diagrama de Clases

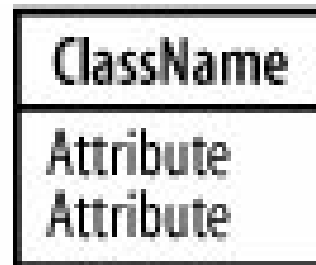
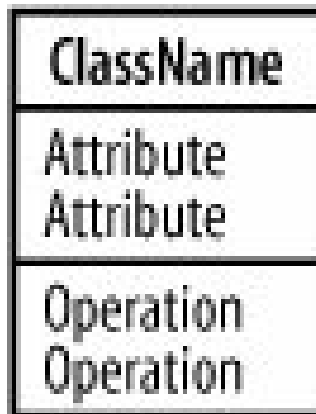


Diagrama de Clases

- El nombre de una clase establece el tipo de objetos, que creará una instancia basada en ella



```
classDiagram
    class BlogAccount
    class BlogEntry
```

BlogAccount

BlogEntry

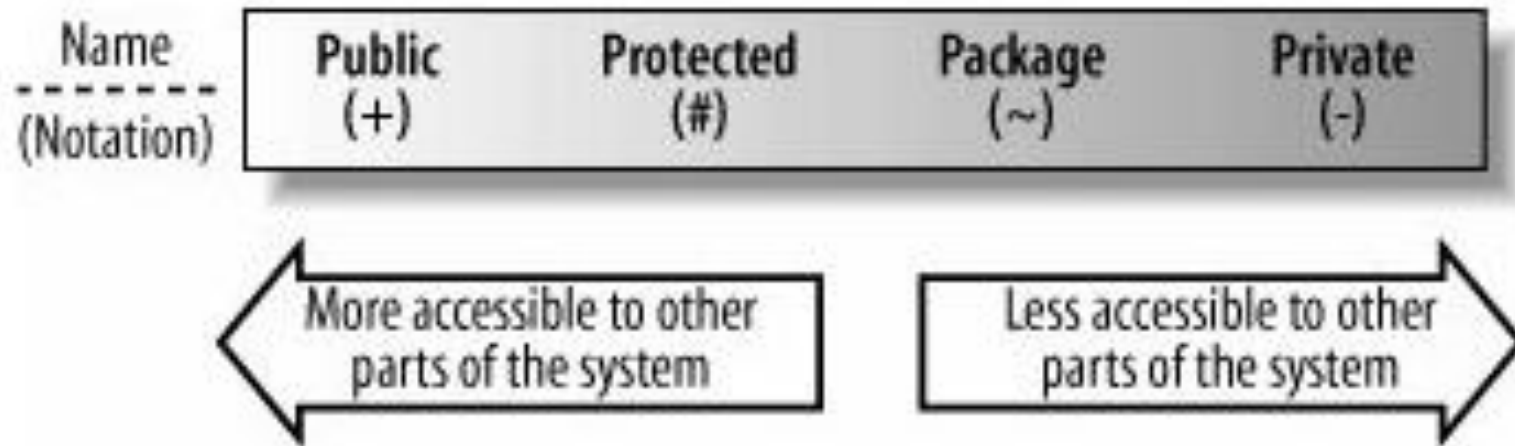
Visibilidad

- ¿Cómo es que una clase de forma selectiva revela sus operaciones y datos a otras clases?
 - Mediante el uso de la visibilidad
- Una vez que las características de la visibilidad se aplican, se puede controlar el acceso a los atributos, operaciones, e incluso clases enteras en la aplicación efectiva de encapsulación

Visibilidad

- Hay cuatro tipos diferentes de visibilidad que se pueden aplicar a los elementos de un modelo UML
 - Pública
 - Privada
 - Protegida
 - Paquete
- Generalmente, estas características de visibilidad se utilizará para controlar el acceso a los atributos, operaciones y, a veces incluso las clases

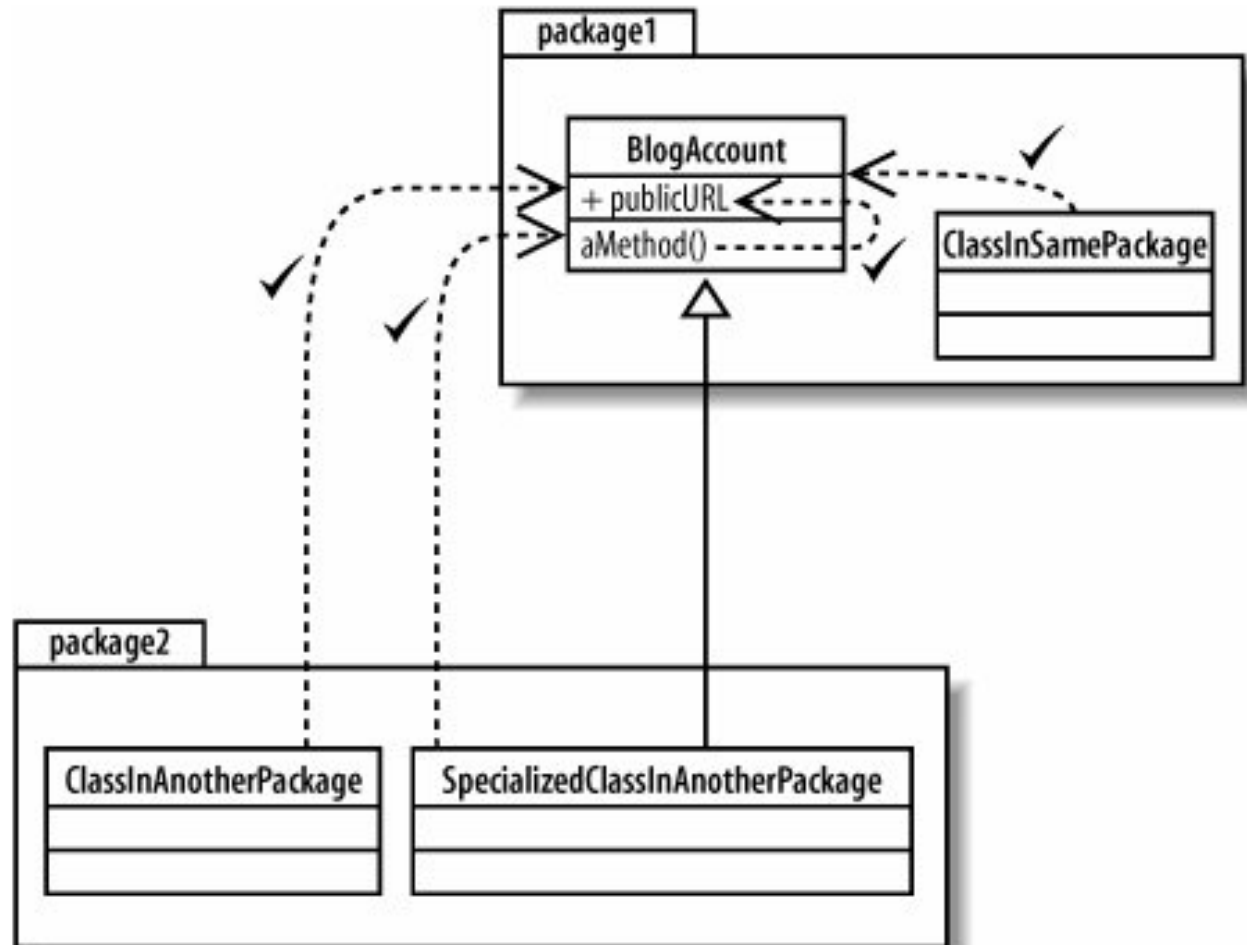
Visibilidad



Visibilidad Pública

- Comenzando con la más accesible de las características de visibilidad, la visibilidad pública se especifica con el signo más (+) delante de los atributos u operaciones asociados
 - Declare un atributo u operación públicos si desea que sea accesible directamente por cualquier otra clase

Visibilidad Pública



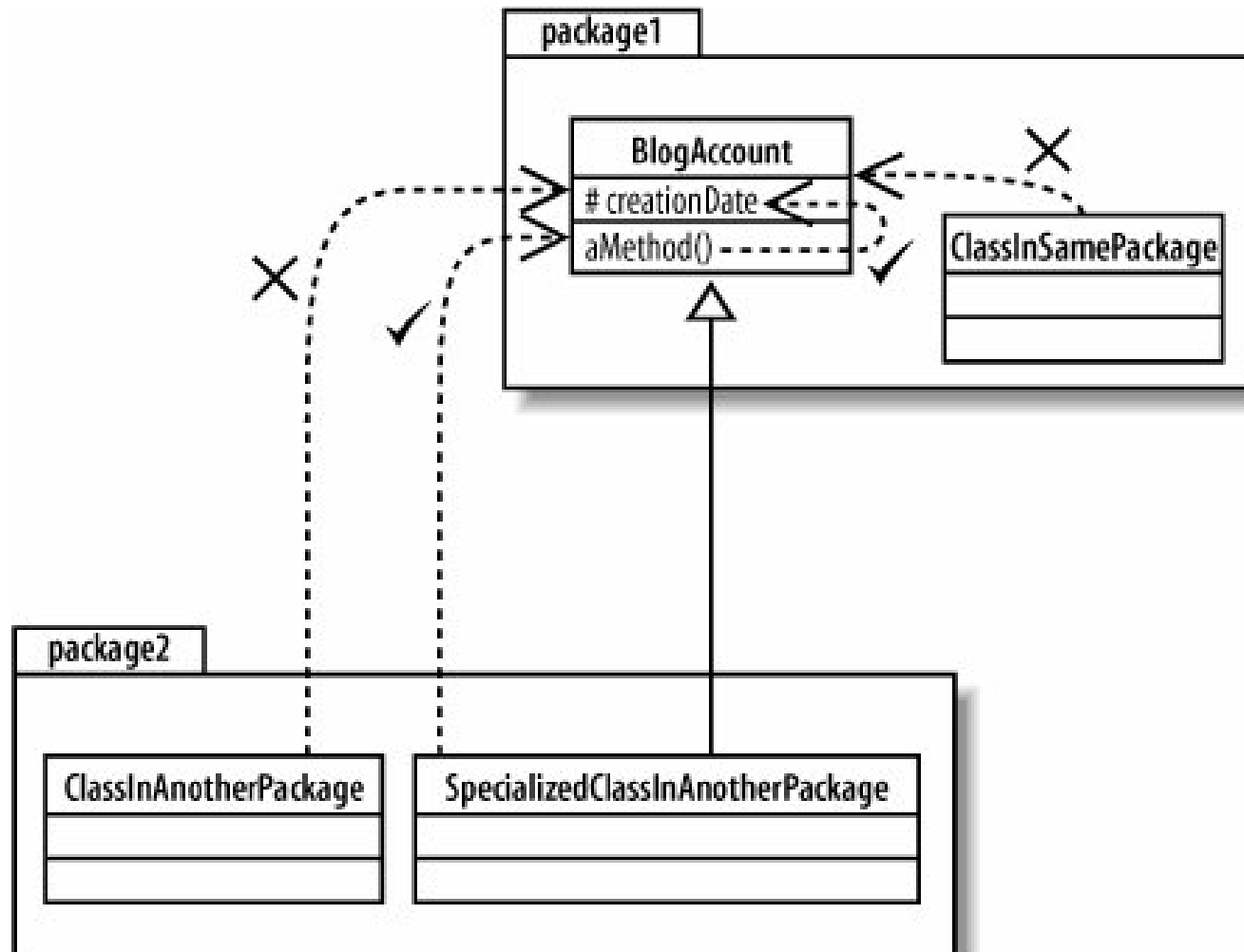
Visibilidad Pública

- La colección de atributos y operaciones que se declaran públicos en una clase crean la interfaz pública de esa clase
- La interfaz pública de una clase se compone de los atributos y las operaciones que pueden ser accedidos y utilizados por otras clases
 - Esto significa que la interfaz pública es la parte de la clase de la que la mayoría de las otras clases dependerá
- Es importante que los cambios en la interfaz pública de las clases sea lo menos posible, para evitar cambios innecesarios cada vez que la clase se utiliza

Visibilidad Protegida

- Los atributos y las operaciones protegidas se especifican mediante el símbolo de la almohadilla (#)
- Es más visible para el resto del sistema que atributos y operaciones privados, pero son menos visibles que los públicos
- Los elementos protegidos de las clases se puede acceder por métodos que son parte de la clase y también por los métodos que se declaran en una clase que hereda de la clase
- Los elementos protegidos no se puede acceder por una clase que no hereda de su clase, ya sea en el mismo paquete o no

Visibilidad Protegida



Visibilidad Protegida

- Visibilidad protegida es crucial si quiere permitir que clases especializadas accedan a un atributo u operación en la clase base sin necesidad de abrir ese atributo u operación a todo el sistema
- Usando visibilidad protegida, es como decir: "Este atributo u operación es útil dentro de mi clase y las clases que se extienden a partir de mi clase, pero nadie más debe usarlo."

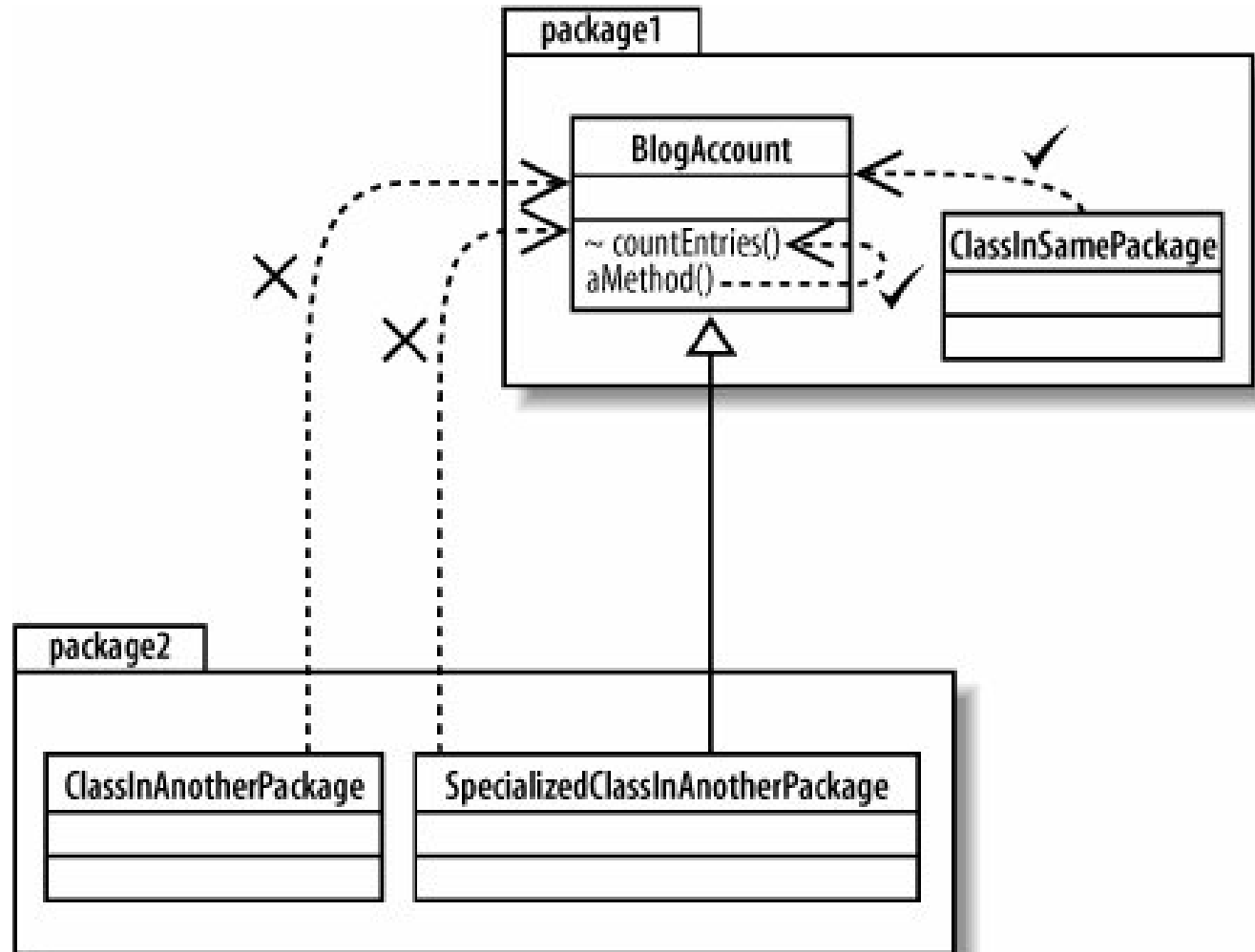
Visibilidad de Paquete

- La visibilidad de paquete, es especificado con una tilde (~), cuando se aplica a los atributos y operaciones, se sitúa entre protegida y privada
- Como era de esperar, los paquetes son el factor clave para determinar qué clases puede ver un atributo u operación que se declara con visibilidad de paquete

Visibilidad de Paquete

- La regla es bastante simple: si se agrega un atributo u operación que se declara con visibilidad de paquete a una clase, entonces cualquier clase en el mismo paquete pueden acceder directamente a ese atributo u operación
- Las clases fuera del paquete no pueden acceder a los atributos u operaciones protegidas, incluso si se trata de una clase que hereda
- En la práctica, la visibilidad de paquete es más útil cuando se desea declarar un conjunto de métodos y atributos a través de sus clases que sólo se puede utilizar dentro de un paquete

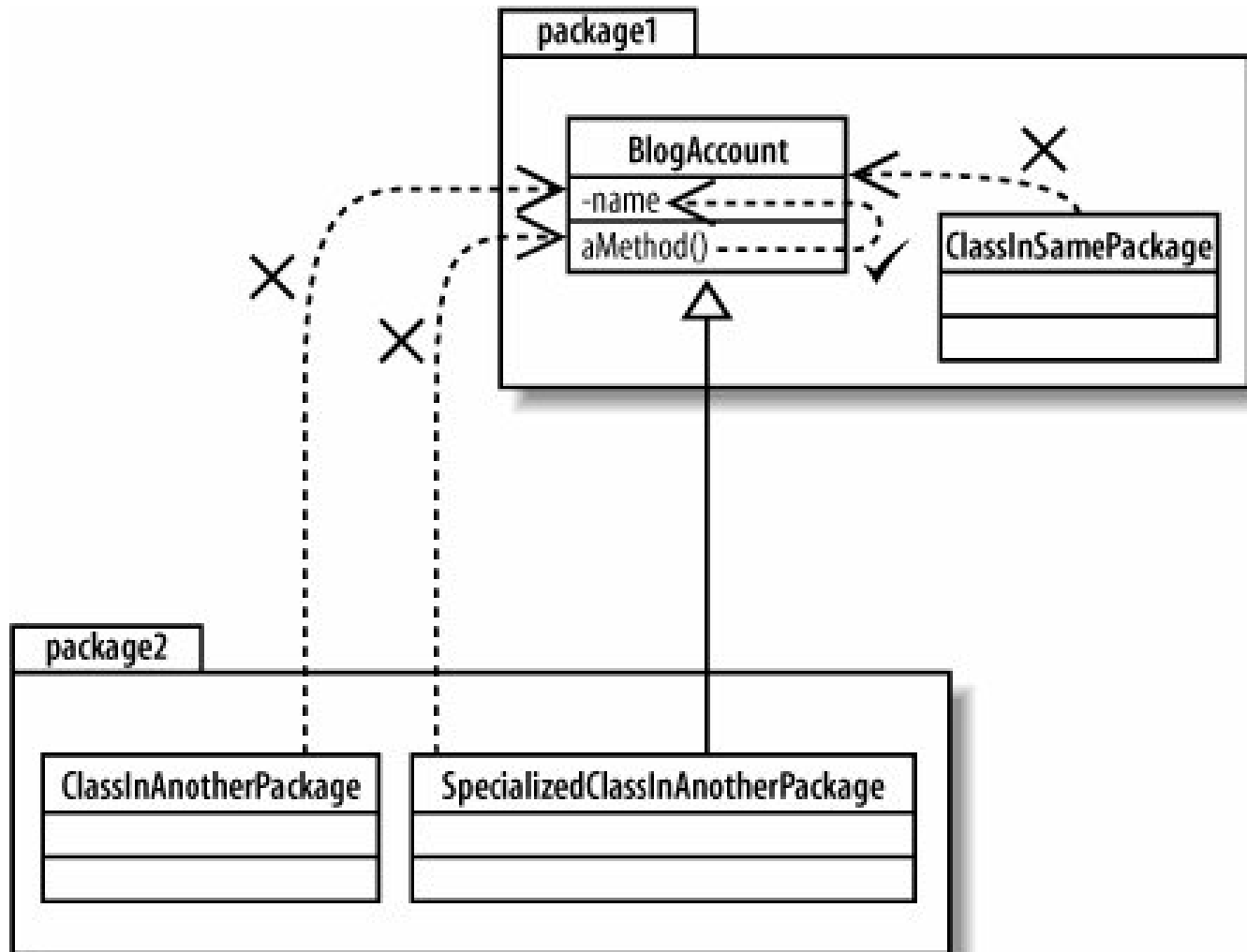
Visibilidad de Paquete



Visibilidad Privada

- Último de la fila en la escala de visibilidad de UML es visibilidad privada
- Visibilidad privada es el tipo más bien limitado de la clasificación de la visibilidad, y se muestra mediante la adición de un signo menos (-) delante del atributo u operación
- Sólo la clase que contiene el elemento privado puede ver o trabajar con los datos almacenados en un atributo privado o hacer una llamada a una operación privada

Visibilidad Privada



Visibilidad Privada

- Visibilidad privada es más útil si se tiene un atributo u operación que desea que ninguna otra parte del sistema dependa de él
- Esto podría ser el caso, si tiene intención de cambiar un atributo u operación en un momento posterior, pero no quiere que las otras clases tengan acceso a ese elemento que va a cambiar