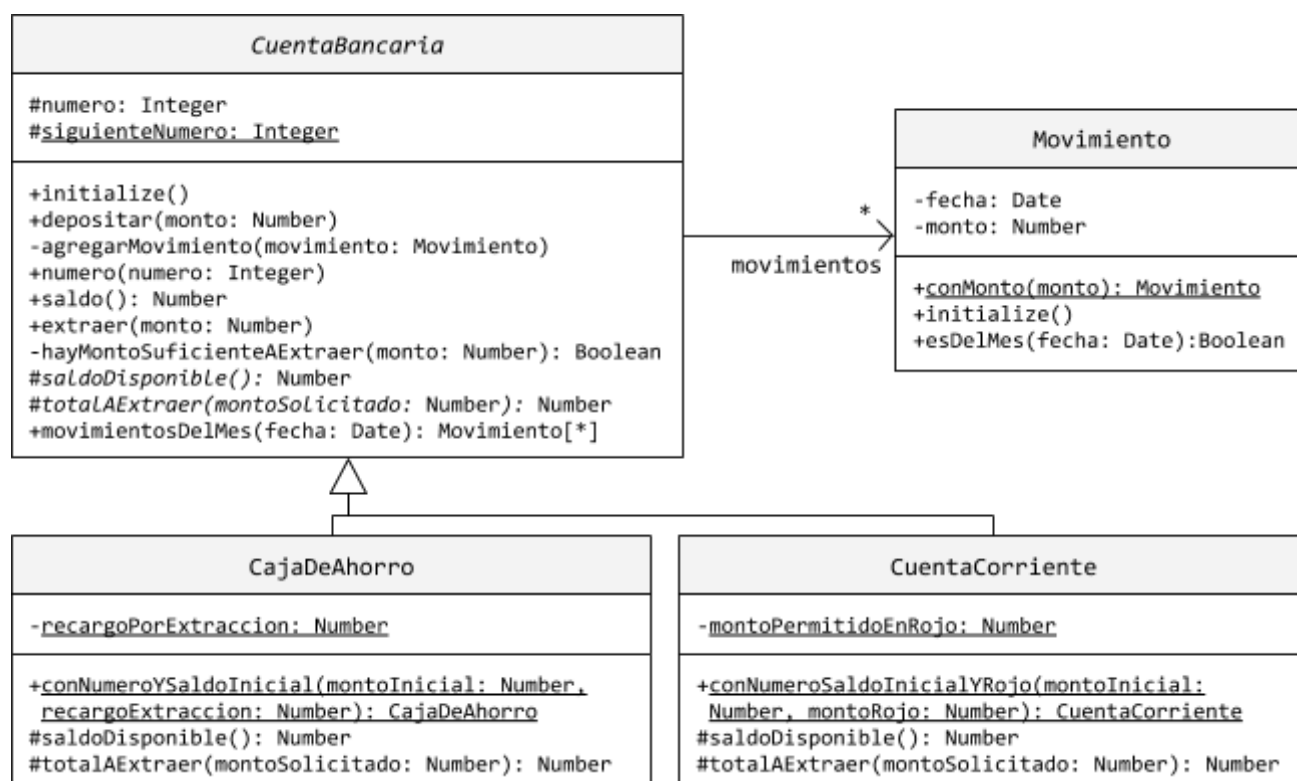


Recomendaciones de sintaxis UML: Diagrama de clases

Si bien es válido especificar una clase sólo con su nombre (sin atributos ni operaciones), los diseños trabajados en clase deben contar con los siguientes tres compartimentos:

NombreDeClase
atributos
operaciones

Por ejemplo:



Nombre de la clase

- Expresé su nombre en **CamelCase**, comenzando en mayúscula.
- Expresé su nombre en cursiva o con la notación «Abstract» si se trata de una clase abstracta. Haciendo uso de herramientas de modelado, lo más común es utilizar la primera forma, mientras que cuando se diseña en papel, lo más legible y recomendable es hacer lo segundo.

Atributos/Propiedades

- Expresé su nombre en **camelCase**, comenzando en minúscula.
- Subrayé los **atributos de clase**.
- Indiqué visibilidad privada (-) o protegida (#) para los atributos, según corresponda. No tengo atributos públicos.
- No incluí como **atributos** aquello que ya fue expresado como final de asociación con otras clases.
- Incluí el tipo de atributo.

Operaciones

- Expresé el nombre en **camelCase**, comenzando en minúscula .
- Puse () al final del nombre aunque no tenga parámetros.
- Indiqué los tipos de parámetros.
- Indiqué visibilidad pública (+), privada (-) o protegida (#) para las operaciones, según corresponda.
- Expresé los **parámetros dentro de paréntesis separados por comas**. Ejemplo:

En Smalltalk:

dash: totalDistance spacing: spaceLength

En UML:

+dashSpacing(totalDistance: Integer, spaceLength: Integer)

- Indiqué su **retorno**, en los casos en que devuelven algún valor. Si no retorna nada, no se especifica el retorno.
- Utilicé un tipo/clase indicando su multiplicidad en los casos en que se devuelve una colección. **No poner "Collection"**. Ejemplo:

+obtenerOfertasDelDia(): Oferta[*]

- Indiqué las operaciones **abstractas** en cursiva o con la notación «Abstract». Haciendo uso de herramientas de modelado, lo más común es utilizar la primera forma, mientras que cuando se diseña en papel, lo más legible y recomendable es hacer lo segundo.

+calcularSueldo(): Float

«abstract» +calcularSueldo(): Float




- Subrayé las **operaciones de clase**.

Respecto a la visibilidad de los miembros (atributos y operaciones), recuerde:

- Se utiliza visibilidad **pública** (+) cuando el miembro es accesible a todos los objetos del sistema.
- Se utiliza visibilidad **protegida** (#) cuando el miembro es accesible a las instancias de la clase que lo implementa y de sus subclases.
- Se utiliza visibilidad **privada** (-) cuando el miembro es sólo accesible a las instancias de la clase que lo implementa.

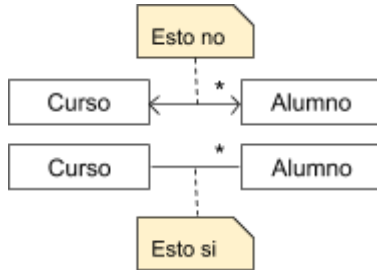
Asociaciones

- Indiqué la:

Asociación	con	
Agregación	con	
Composición	con	

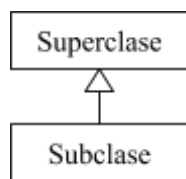
- Indiqué el nombre del **rol** en cada extremo del destino navegable que tenga la asociación, opuesto a la clase origen.
- Indiqué la multiplicidad en todos los casos, salvo que la misma sea "1".
- Indiqué la dirección en las relaciones unidireccionales.

- **No** indiqué la dirección en las asociaciones bidireccionales (conocimiento mutuo).



Generalización

- Indiqué la relación de generalización con una flecha sin relleno, apuntando en dirección a la superclase.



- **No** incluí nombre ni multiplicidad a esta relación.

Interfaces

- Al incluir una interfaz, la identifiqué mediante el estereotipo «Interface».
- Indiqué el nombre de la interfaz con cursiva, ya que su comportamiento es abstracto.
- Especifiqué todos los métodos de la interfaz como públicos y abstractos.
- Me aseguré que la clase que implementa la interfaz implementa todos los métodos de la interfaz.
- La relación con la clase que la implementa, se representa con una flecha sin relleno y línea punteada, apuntando en dirección a la interfaz. Como la generalización, pero con línea punteada.

