Comportamiento común entre Objetos

- Volvamos al ejemplo del banco: ¿Cuántos objetos Caja de Ahorro habrá?
- ¿Es necesario especificar el comportamiento de *cada* Caja de Ahorro? ¿O el comportamiento debería ser común a todas?
- ¿Qué cosas son comunes a todas las Cajas de Ahorro y qué cosas son particulares de cada una?
- Entonces … ¿Cómo representamos este comportamiento común, de manera que cada Caja de Ahorro pueda reutilizarlo?

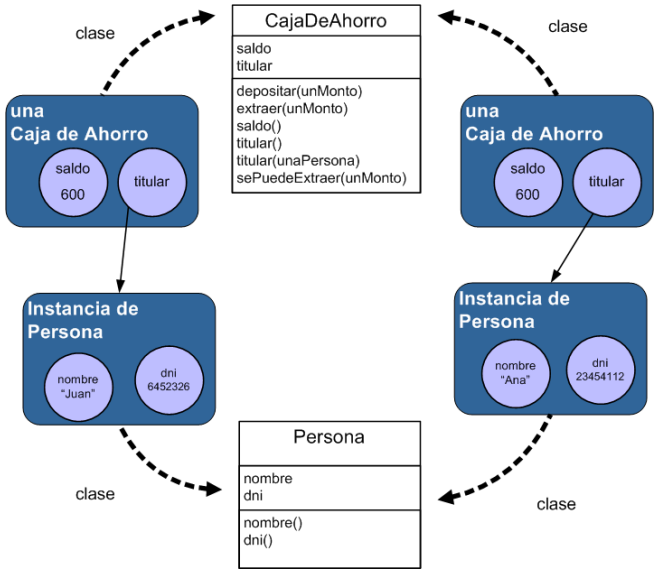
Lifia

Clases e instancias

- Una clase es una descripción abstracta de un conjunto de objetos.
- Las clases cumplen tres roles:
  - Agrupan el comportamiento común a sus instancias.
  - Definen la ***forma*** de sus instancias.
  - *Crean objetos que son instancia de ellas*
- En consecuencia todas las instancias de una clase se comportan de la misma manera.
- Cada instancia mantendrá su propio estado interno.

Lifia

Ejemplo de clases e instancias

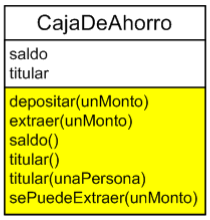


Lifia

Especificación de Clases

- Las clases se especifican por medio de un nombre, el estado o estructura interna que tendrán sus instancias y los métodos asociados que definen el comportamiento
- Gráficamente:

Variables de Instancia
Los nombre de las v.i. se escriben en minúsculas y sin espacios

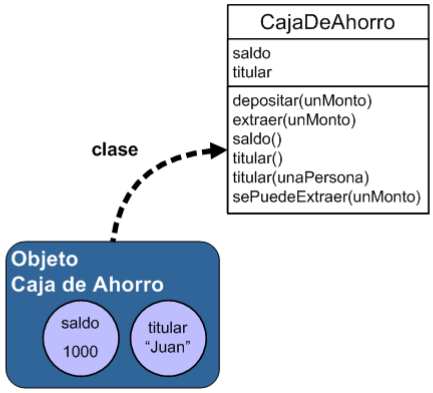


Nombre de la Clase
Comienzan con mayúscula y no posee espacios

Protocolo
Para cada mensaje se debe especificar como mínimo el nombre y los parámetros que recibe

Lifia

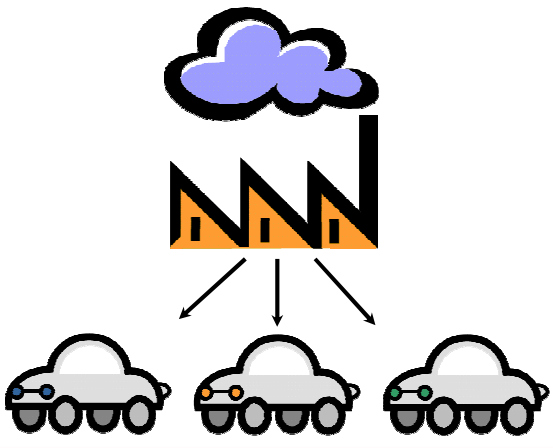
Envío de mensajes con clases I



Lifia

Creación de Objetos

- ¿Cómo creamos nuevos objetos?
- Instanciación



Lifia

Creación de Objetos

- Comúnmente se utiliza la palabra reservada ***new*** para instanciar nuevos objetos.
- Según el lenguaje
  - **new** es un mensaje que se envía a la clase.
  - **new** es una operación especial.
- Nosotros vamos a usar al new como mensaje de clase. En este caso, la ejecución del método retorna una nueva instancia de la clase a la que se le envía el mensaje.

Lifia

Instanciación

- Es el mecanismo de creación de objetos.
- Los objetos se *instancian* a partir de un molde.
- La **clase** funciona como molde.
- Un nuevo objeto es una ***instancia*** de una clase.
- Todas las instancias de una misma clase
  - Tendrán la misma estructura interna.
  - Responderán al mismo protocolo (los mismos mensajes) de la misma manera (los mismos métodos).

Lifia

Una instancia recién creada …

- ¿está lista para poder colaborar con otros objetos?
  - Pensemos en la creación de un objeto *fecha*.
  - ¿Podemos sumar un punto recién creado?
  - Una cuenta bancaria recién creada ¿sabe quién es su titular?

Lifia

Inicialización

- Para que un objeto esté listo para llevar a cabo sus responsabilidades hace falta *inicializarlo*.
- Inicializar un objeto significa darle valor a sus variables.
- ¿De dónde sacamos esos valores iniciales?

Lifia

Clases y subclases

- Una clase representa un concepto en el dominio de problema.
- ¿Qué sucede cuando las clases tienen comportamiento común?

  → **Subclasificación**

Ejemplo de cuentas bancarias

- Existen dos tipos de cuentas bancarias:
  - Cuentas corrientes.
  - Cajas de ahorro.
- Si revisamos el comportamiento nos encontraremos con las siguientes características en común:
  - Ambas llevan cuenta de su saldo.
  - Ambas permiten realizar depósitos.
  - Ambas permiten realizar extracciones.

Ejemplo de cuentas bancarias

- Pero cada una tiene distintas restricciones en cuanto a las extracciones:
  - Cuentas corrientes: permiten que el cliente extraiga en descubierto (con un tope pactado con cada cliente).
  - Cajas de ahorro: poseen una cantidad máxima de extracciones mensuales (para todos los clientes). No se permite extraer en descubierto.
- ¿Cómo podemos reutilizar las características en común?

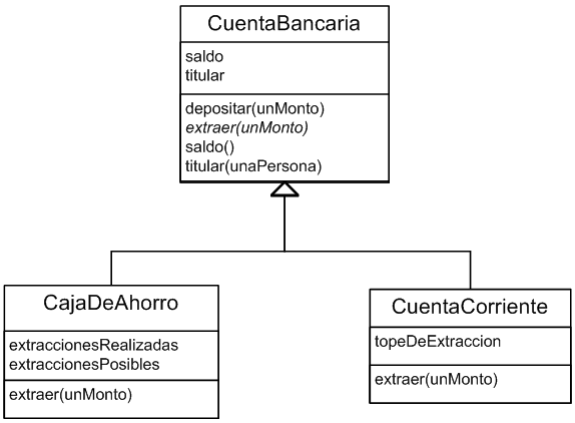
Lifia

Subclasificación

- Se reúne el comportamiento y la estructura común en una clase, la cual cumplirá el rol de ***superclase***.
- Se conforma una *jerarquía de clases*.
- Luego otras clases pueden cumplir el rol de subclases, ***heredando*** ese comportamiento y estructura en común.
- Debe cumplir la relación ***es-un***.

Lifia

Ejemplo de una Jerarquía de Clases



Lifia

Relación *es-un*

- En toda jerarquía de clases, se debe respetar la relación *es-un* entre una clase y su superclase.
- Verdadero o falso?
  - Una **CajaDeAhorro** *es-una* **CuentaBancaria**
  - Una **CuentaBancaria** *es-un* **Banco**
  - Un **TitularDeCuenta** *es-un* **EmpleadoDelBanco**

Lifia

Herencia

- Es el mecanismo por el cual las subclases reutilizan el comportamiento y estructura de su superclase.
- La herencia permite:
  - Crear una nueva clase como refinamiento de otra.
  - Diseñar e implementar sólo la diferencia que presenta la nueva clase.
  - Factorizar similitudes entre clases.

Lifia

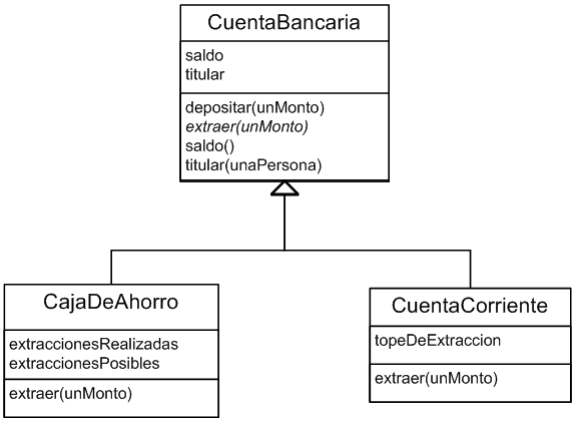
Herencia

- Toda relación de herencia implica:
  - Herencia de comportamiento
    - Una subclase hereda ***todos*** los métodos definidos en su superclase.
    - Las subclases pueden ***redefinir*** el comportamiento de su superclase.
  - Herencia de estructura
    - No hay forma de restringirla.
    - No es posible redefinir el nombre de un atributo que se hereda.

Lifia

Ejercicio - Cuenta Bancaria

- Implementar el mensaje **`extraer(unMonto)`** en cada una de las subclases de CuentaBancaria.



Lifia

Ejercicio - Cuenta Bancaria

- Recordemos las restricciones:
  - Cuentas corrientes: permiten que el cliente extraiga en descubierto (con un tope pactado con cada cliente).
  - Cajas de ahorro: poseen una cantidad máxima de extracciones mensuales (para todos los clientes). No se permite extraer en descubierto.

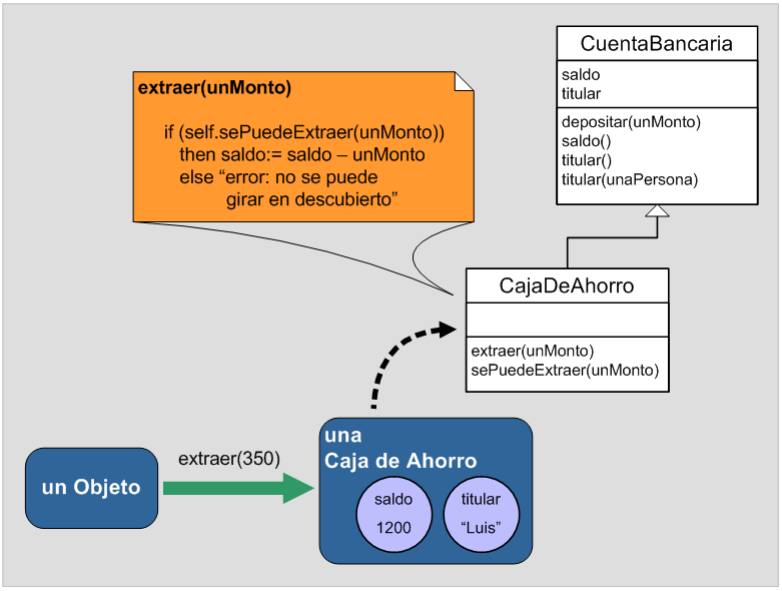
Lifia

Method Lookup

- Al enviarse un mensaje a un objeto:
  - Se determina cuál es la clase del objeto.
  - Se busca el método para responder al envío del mensaje en la jerarquía, comenzando por la clase del objeto, y subiendo por las superclases hasta llegar a la clase raíz (Object)
- Este proceso se denomina ***method lookup***

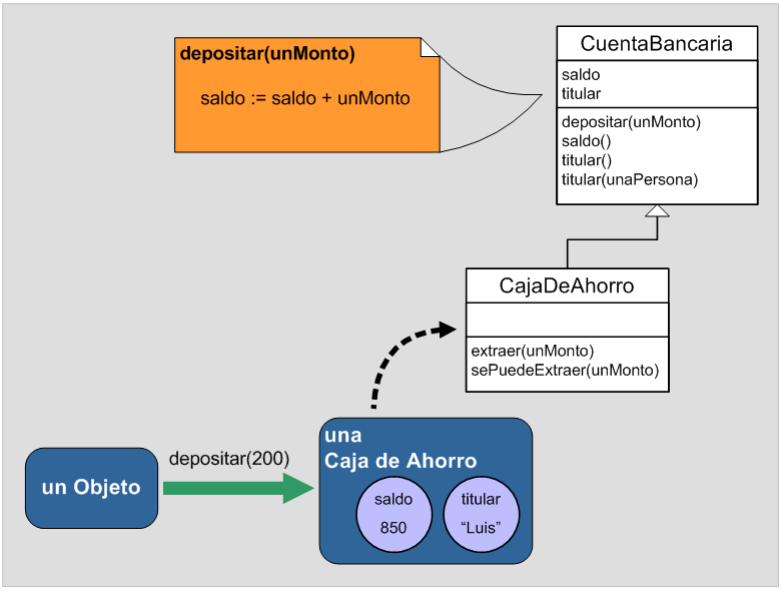
Lifia

Method Lookup



Lifia

Method Lookup



Métodos abstractos

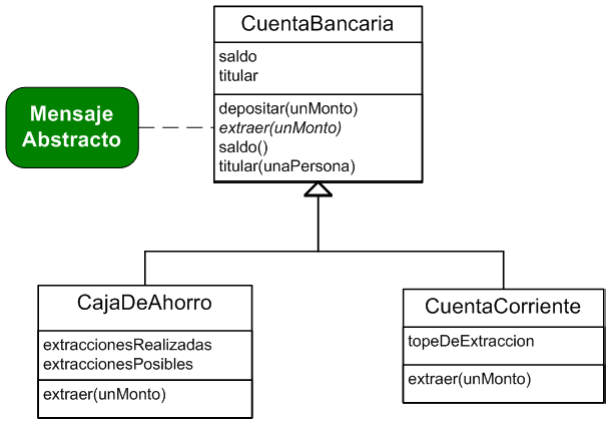
- Son importantes para establecer el protocolo de una jerarquía de clases
- No se especifica el comportamiento, ya que a nivel de la superclase no se puede prever.
- En los diagramas de clase UML los métodos abstractos se escriben en *letra cursiva*.

Métodos abstractos

- Las subclases concretas deben tener implementaciones de los métodos abstractos, ya sea dentro de la clase o en alguna de sus superclases.

Lifia

Mensajes abstractos



Lifia

Clases Abstractas

- Una **clase abstracta** es una clase que no puede ser instanciada.
- ¿Entonces, para qué sirven?
  - Se diseña sólo como clase padre de la cual derivan subclases.
  - Representan conceptos o entidades abstractas.
  - Sirven para factorizar comportamiento común.
  - Usualmente, tiene partes incompletas.
    - Las subclases completan las piezas faltantes, o agregan variaciones a las partes existentes.

Lifia

Relaciones entre herencia y encapsulamiento

- Toda subclase hereda estructura y comportamiento.
- Entonces puede acceder a:
  - Las variables de instancia.
  - Los métodos.

  definidos en la superclase.

Lifia