Clases y subclases

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

  → **Subclasificación**

Lifia

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.

Lifia

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 (y a veces algo mas)***.
- Es facil encontrar jerarquias? Cuando las encontramos?

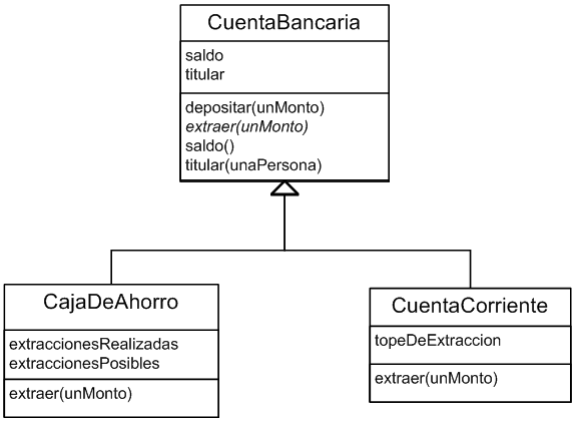
Lifia

Para que buscar jerarquias?

- Entender mejor el dominio
- Reducir algo de codigo y descripciones redundantes
- Aprovechar mejor el polimorfismo

Lifia

Ejemplo de una Jerarquía de Clases



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

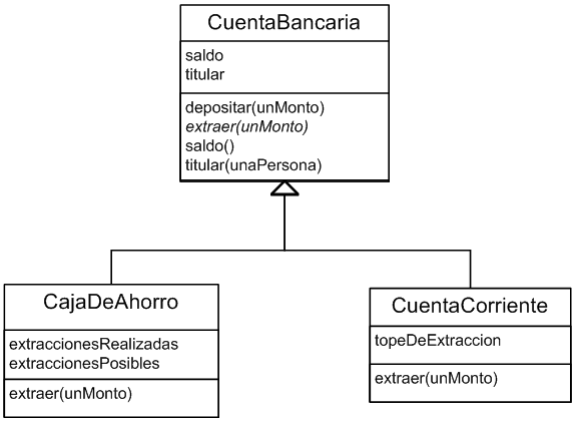
Herencia

- 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.



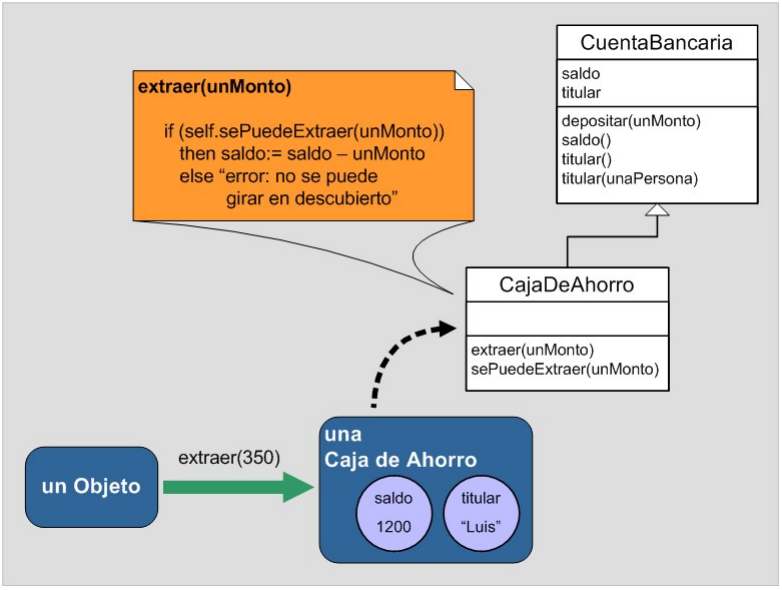
Lifia

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***

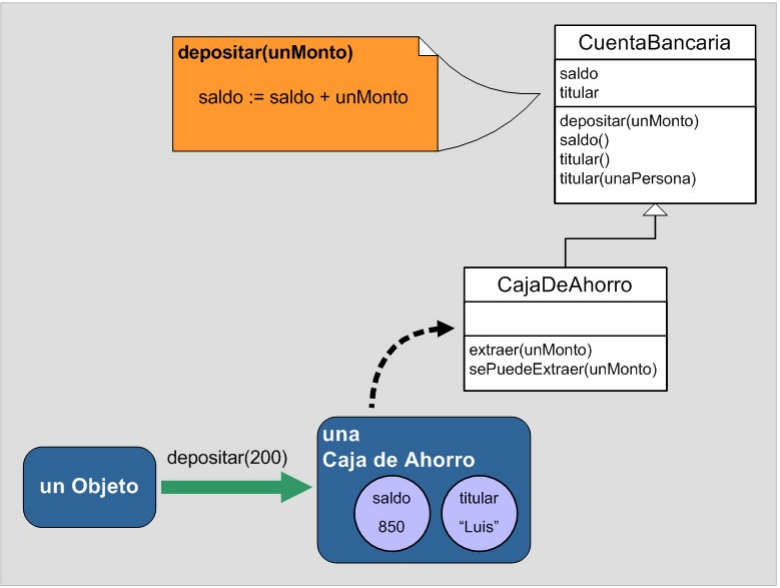
Lifia

Method Lookup



Lifia

Method Lookup



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.

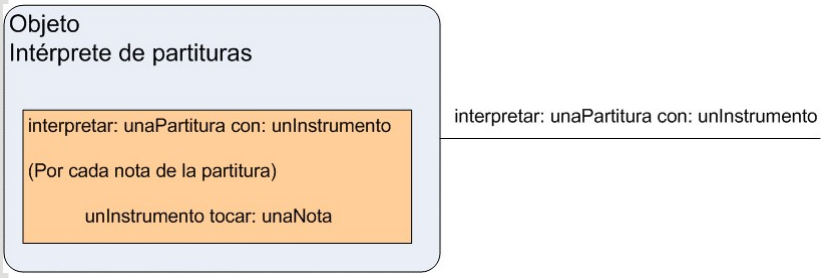
Polimorfismo

- Dos o más objetos son ***polimórficos*** con respecto a un mensaje, si todos pueden entender ese mensaje, aún cuando cada uno lo haga de un modo diferente
  - *Mismo mensaje puede ser enviado a diferentes objetos*
  - *Distintos receptores reaccionan diferente (diferentes métodos)*

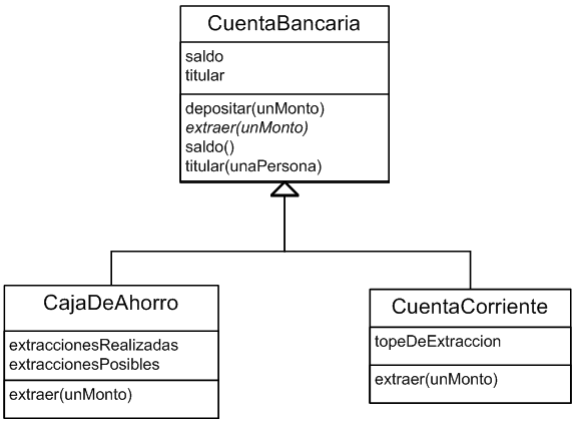
Ejemplo de objetos polimórficos

Polimorfismo

- Ejemplo:



Ejemplo: Cuentas Bancarias



Lifia

Ventajas del uso del polimorfismo

- *Código* genérico
- Objetos desacoplados
- Objetos intercambiables
- Objetos reutilizables
- Programar por protocolo, no por implementación

- Bien usado, implica menos codigo

Lifia LIFIA

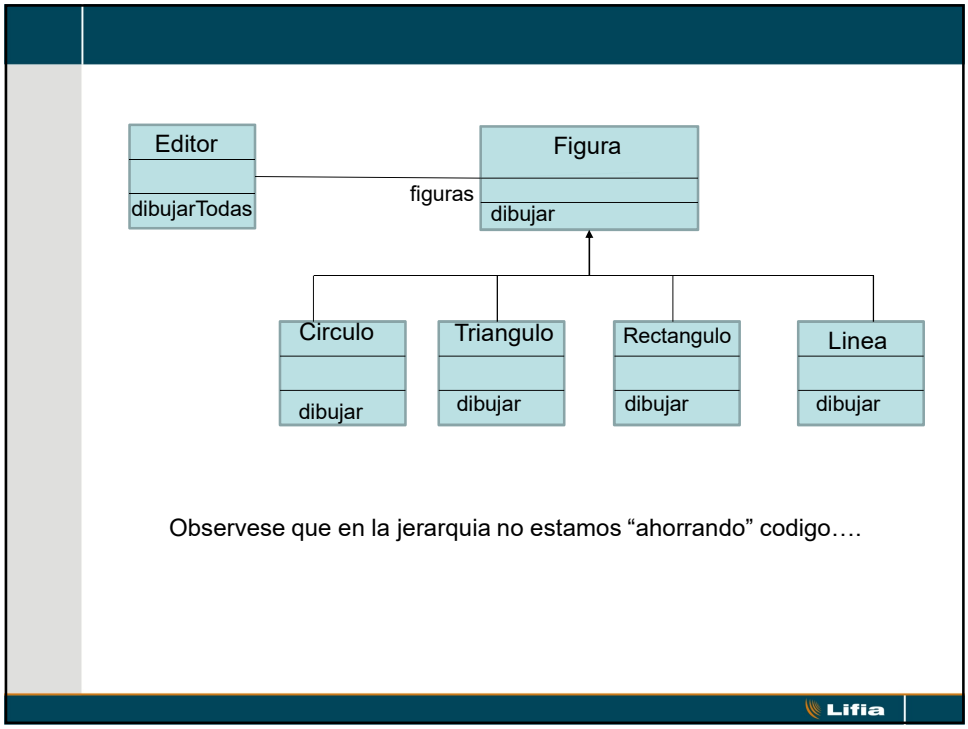
Polimorfismo + Binding Dinamico

- Que es binding? Asociacion de operadores y operandos
- Ej: X= A + B….. Que significa?
- Ej (Procedural): Dibujar (X) Que significa?
- Ej: (OO): x dibujar…..Que significa?
- Por que es diferente?

Cuando se hace el binding? Por que?
Procedural?
OO?

Ejercicio – Figuras

- Supongamos que tenemos que diseñar un editor gráfico para figuras en dos dimensiones
- Las figuras pueden ser
  - Rectángulos
  - Triángulos
  - Círculos
- El editor debe poder dibujar las figuras, pero naturalmente dibujará cada figura de distinta manera



Observese que en la jerarquia no estamos “ahorrando” codigo….

Lifia

a1

El problema del “if”

- Analicemos el siguiente método del editor: Dibujar todas las figuras definidas

```
For i= 1 to N
  Si (figuras[i] es rectangulo) entonces
        dibujarrectangulo.
  Si (figuras [i] es circulo) entonces
        dibujarcirculo.
  Si (figuras [i] es triangulo) entonces
        dibujartriangulo.
```

Lifia LIFIA

Diapositiva 89

a1 podria ir antes que el ejemplo de figuras
agustinm; 06/12/2005

Con Polimorfismo y Binding dinamico

For i= 1 to N

Figuras [i] dibujar