

# 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:

Objeto

Intérprete de partituras

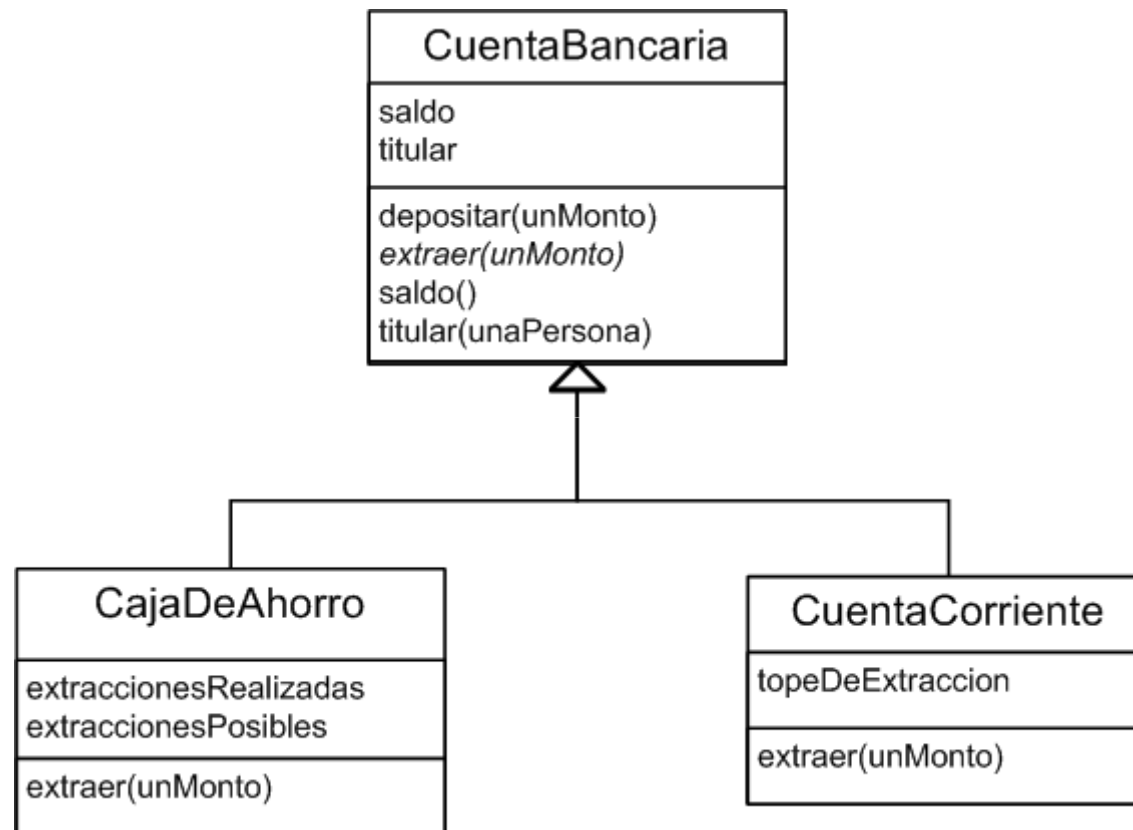
interpretar: unaPartitura con: unInstrumento

(Por cada nota de la partitura)

unInstrumento tocar: unaNota

interpretar: unaPartitura con: unInstrumento

# Ejemplo: Cuentas Bancarias



# Ventajas del uso del polimorfismo

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

# 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? (Que pasa con el control de tipos)

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

## El problema del “if”

- El if es a la Programación Orientada a Objetos lo que el **goto** era para la Programación Estructurada
  - Los lenguajes OO proveen if pero...
  - **El mal uso de If** significa la falta de objetos polimórficos
    - Las decisiones son tomadas por los programadores
    - Mala asignación de responsabilidades
- ¿Cuándo se justifica su uso?

## ¿Qué pasaría sin polimorfismo?

- Veamos el siguiente método.

```
if (unaCuenta.esCajaDeAhorro())  
    then unaCuenta.extraerDeCajaDeAhorro(100)  
    else unaCuenta.extraerDeCuentaCorriente(100)
```

- ¿Qué problema presenta?
- ¿Qué sucede si aparece un nuevo tipo de cuenta, por ejemplo CuentaUniversitaria?



# Polimorfismo

- Hay que modificar el código:

```
if (unaCuenta.esCajaDeAhorro())  
    then unaCuenta.extraerDeCajaDeAhorro(100);  
    else if (unaCuenta.esCuentaCorriente())  
        then unaCuenta.extraerDeCuentaCorriente(100);  
        else unaCuenta.extraerDeCuentaUniversitaria(100);
```

- Si delegamos la lógica en cada cuenta, el código del método sería:

```
unaCuenta.extraer(100);
```

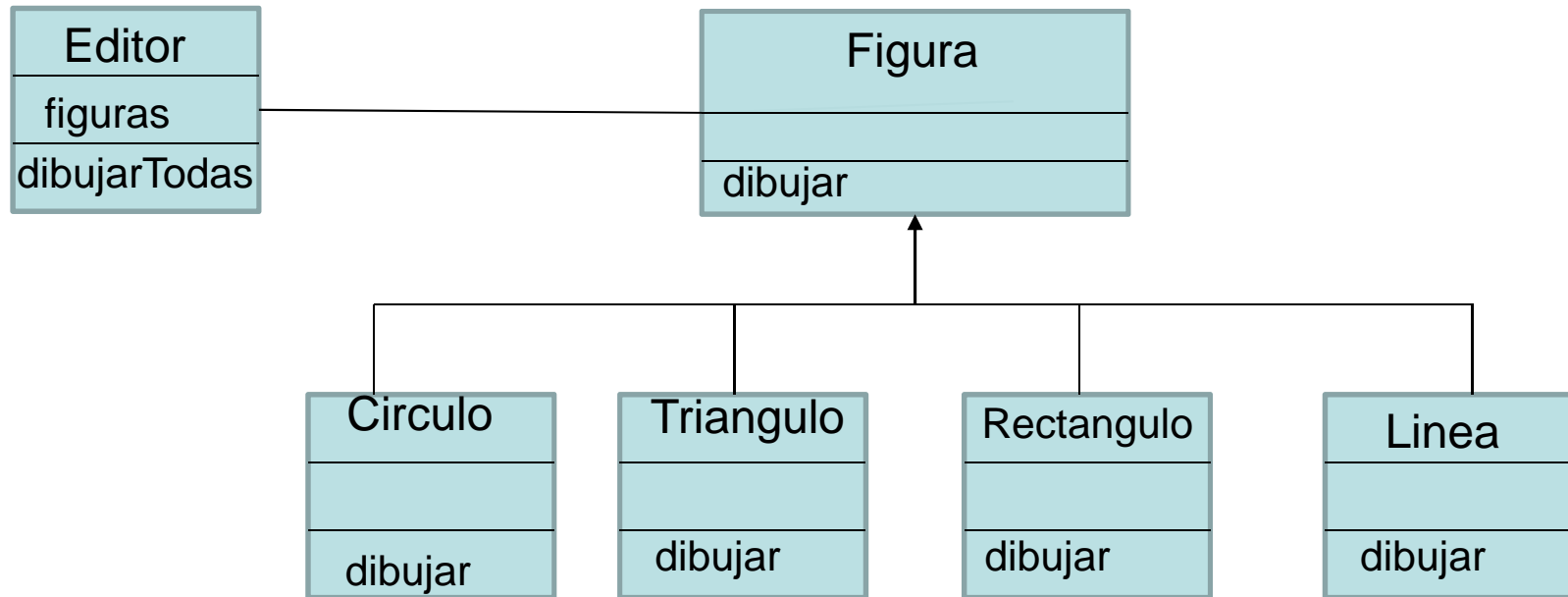
## En el Banco

- Como seria el Metodo *chequearCuenta (tarjeta)*

```
chequearCuenta (tarjeta)  
  
t:=tarjeta.titular  
...  
...
```

## *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



## 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  
dibujarCuadrado.*

*Si (figuras [i] es circulo) entonces  
dibujarCirculo.*

*Si (figuras [i] es triangulo) entonces  
dibujarTriangulo.*

## Diapositiva 74

---

**a1**

podria ir antes que el ejemplo de figuras

agustinm; 06/12/2005

For i= 1 to N

Figuras [i] dibujar

# Ventajas?

- Por que es diferente?
- Por que es mejor?