

# Programación III

## Clase IX

*Andrés Fortier*

# La pregunta de la Rhodesia

- Defina polimorfismo y de un ejemplo de su aplicación.

# Repaso - Polimorfismo

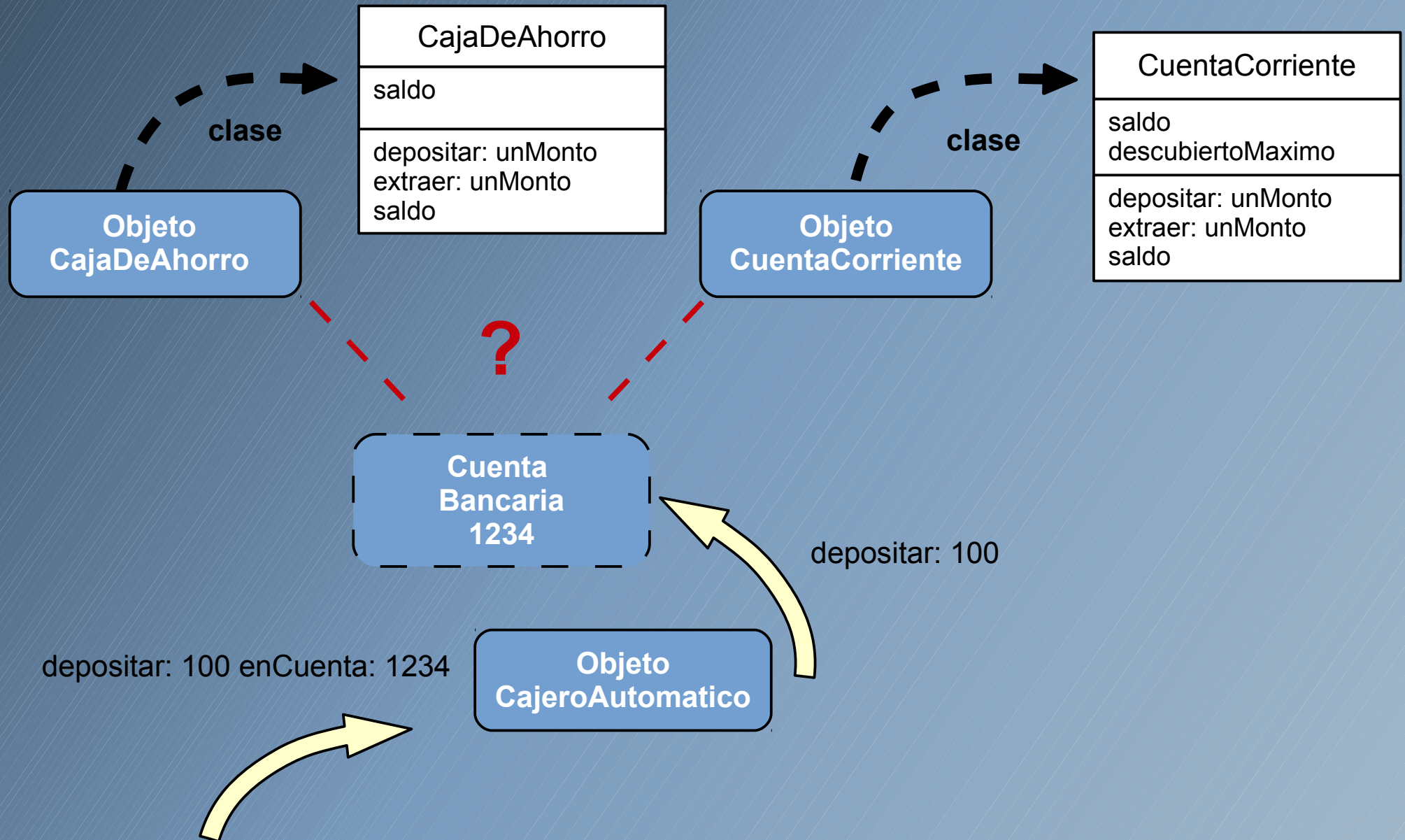
- Dos o más objetos son polimórficos con respecto a un mensaje si pueden entender ese mensaje, aún cuando cada uno lo haga de un modo diferente.
  - Mismo mensaje puede ser enviado a diferentes objetos.
  - Distintos receptores reaccionan diferente (diferentes métodos).



# Cuentas bancarias

- Un banco tiene dos clases de cuentas bancarias:
  - Cajas de ahorro.
  - Cuentas corrientes.
- Un usuario accede a su cuenta por medio de un cajero automático.

# Cuentas bancarias



# 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.



# Cuentas bancarias

## CajaDeAhorro

saldo

depositar: unMonto  
extraer: unMonto  
saldo

depositar: unMonto

saldo := saldo + unMonto.

saldo

^saldo.

extraer: unMonto

saldo >= unMonto  
If True: [saldo := saldo - unMonto]

## CuentaCorriente

saldo  
descubiertoMaximo

depositar: unMonto  
extraer: unMonto  
saldo

depositar: unMonto

saldo := saldo + unMonto.

saldo

^saldo.

extraer: unMonto

saldo + descubiertoMaximo >= unMonto  
If True: [saldo := saldo - unMonto]

# Cuentas bancarias

depositar: unMonto

saldo := saldo + unMonto.

CajaDeAhorro

saldo

depositar: unMonto  
extraer: unMonto  
saldo

saldo

^saldo.

extraer: unMonto

saldo >= unMonto  
If True: [saldo := saldo - unMonto]

CuentaCorriente

saldo  
descubiertoMaximo

depositar: unMonto  
extraer: unMonto  
saldo

saldo

^saldo.

extraer: unMonto

saldo + descubiertoMaximo >= unMonto  
If True: [saldo := saldo - unMonto]



# Cuentas bancarias

depositar: unMonto

saldo := saldo + unMonto.

saldo

^saldo.

CajaDeAhorro

saldo

depositar: unMonto  
extraer: unMonto  
saldo

CuentaCorriente

saldo  
descubiertoMaximo

depositar: unMonto  
extraer: unMonto  
saldo

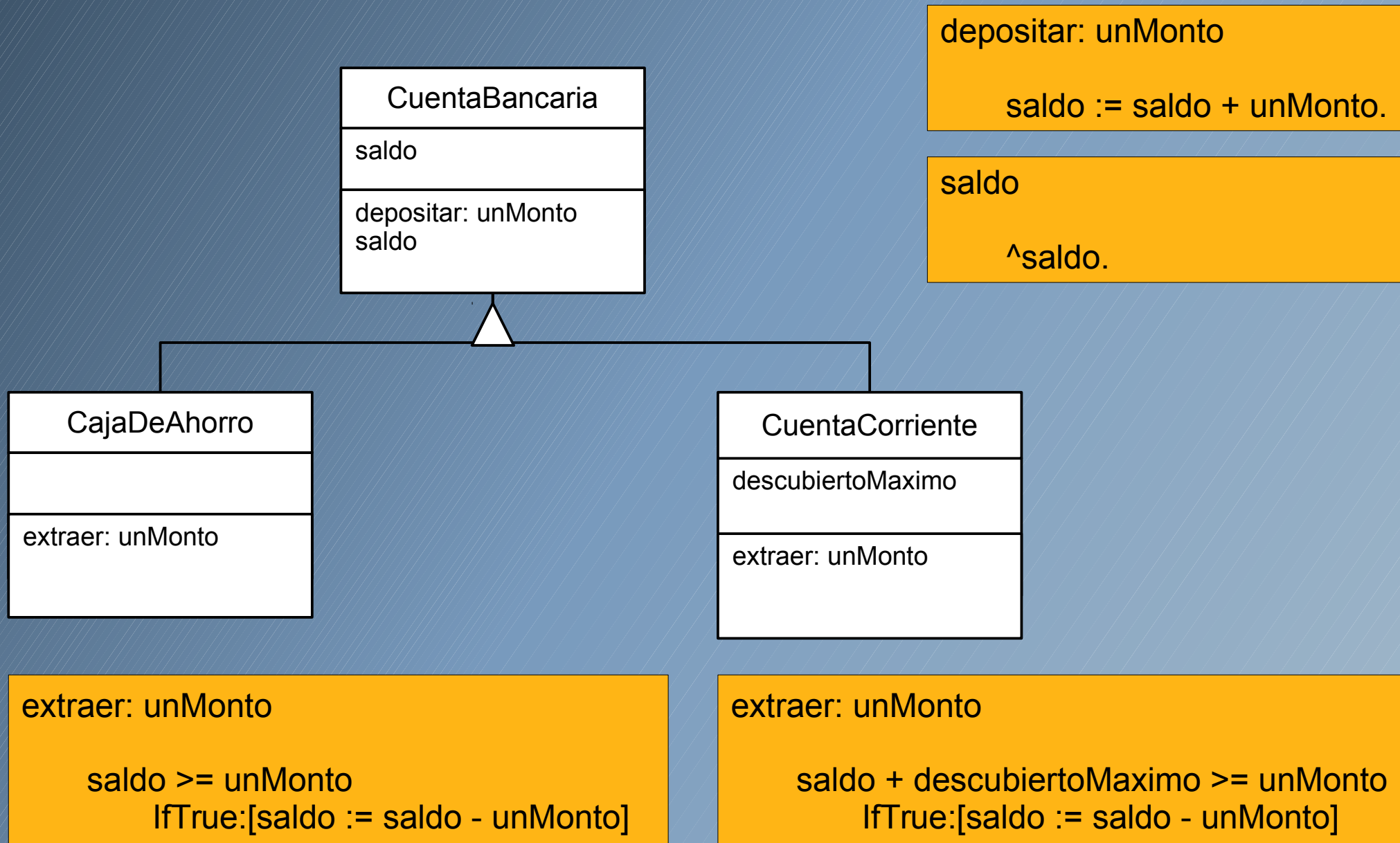
extraer: unMonto

saldo >= unMonto  
If True: [saldo := saldo - unMonto]

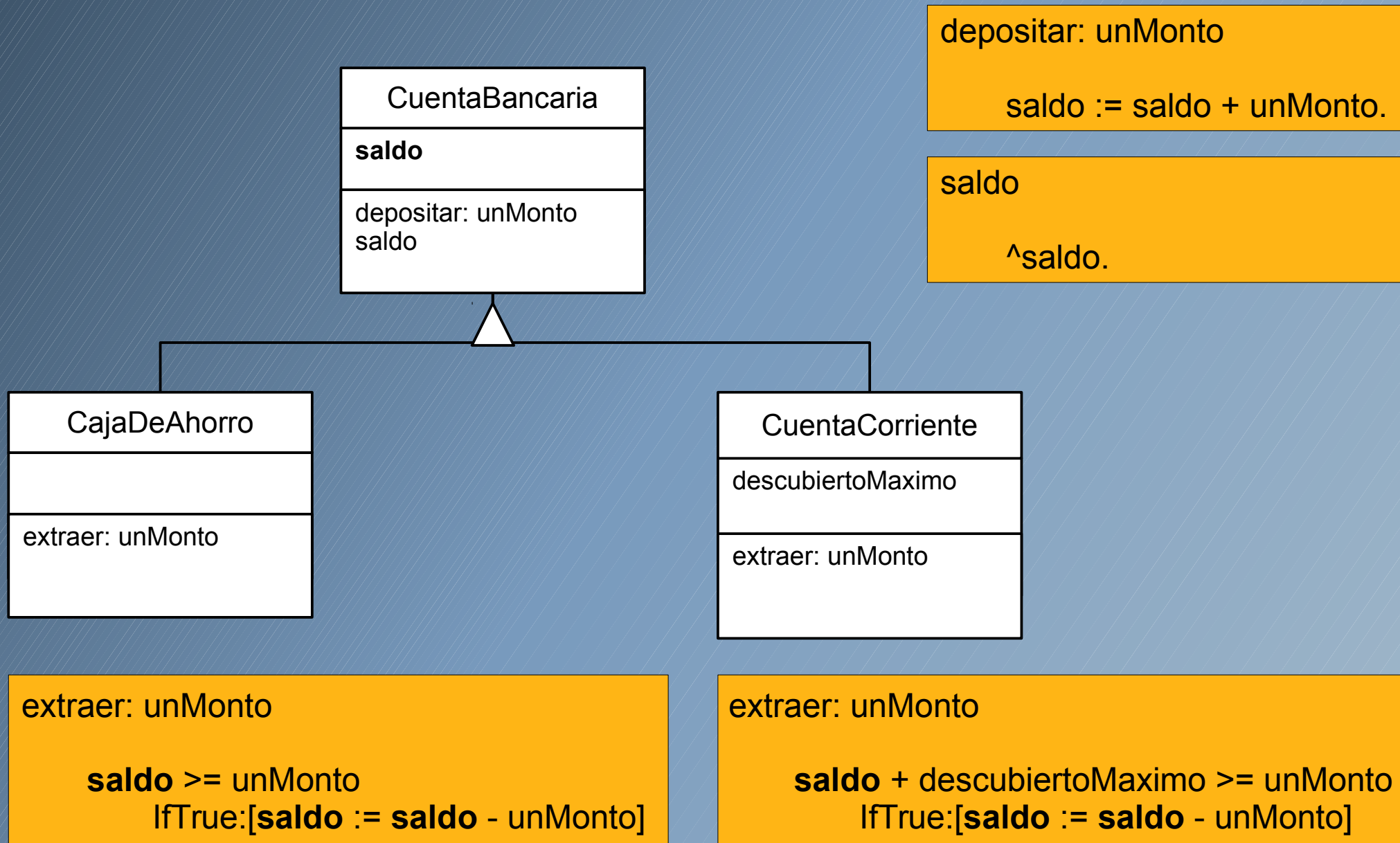
extraer: unMonto

saldo + descubiertoMaximo >= unMonto  
If True: [saldo := saldo - unMonto]

# Cuentas bancarias



# Cuentas bancarias - herencia de v.i.





# Envío de mensaje

depositar: unMonto

saldo := saldo + unMonto.

superclase

CuentaBancaria

saldo

depositar: unMonto  
saldo

clase

CajaDeAhorro

extraer: unMonto

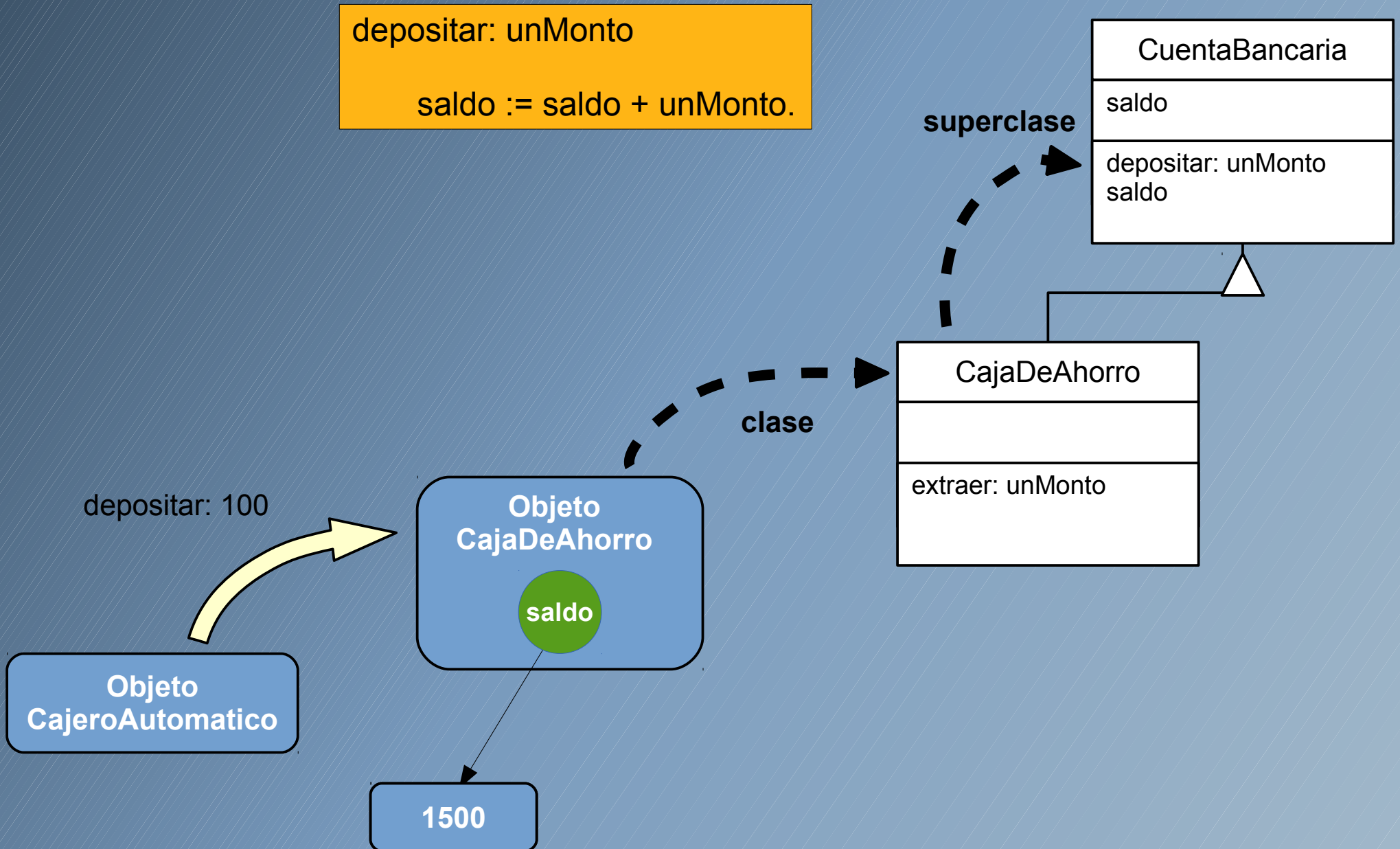
depositar: 100

Objeto  
CajeroAutomatico

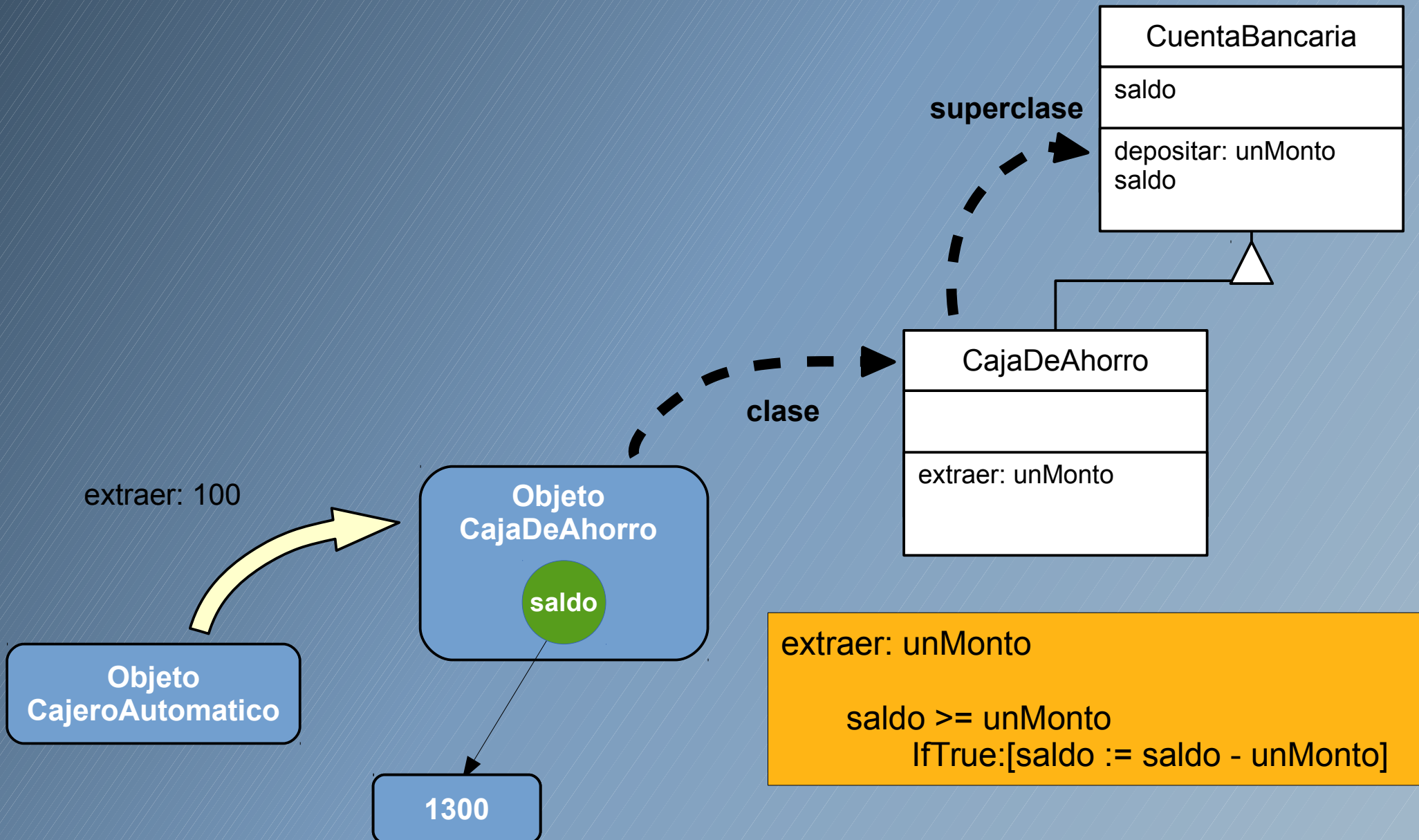
Objeto  
CajaDeAhorro

saldo

1500



# Envío de mensaje



# Herencia de comportamiento

- Ambas cuentas bancarias heredan los métodos
  - #saldo.
  - #depositar: unMonto.
- Cada cuenta implementa el mensaje #extraer: unMonto.



# Ejercicio - Modelo reducido de ruleta