

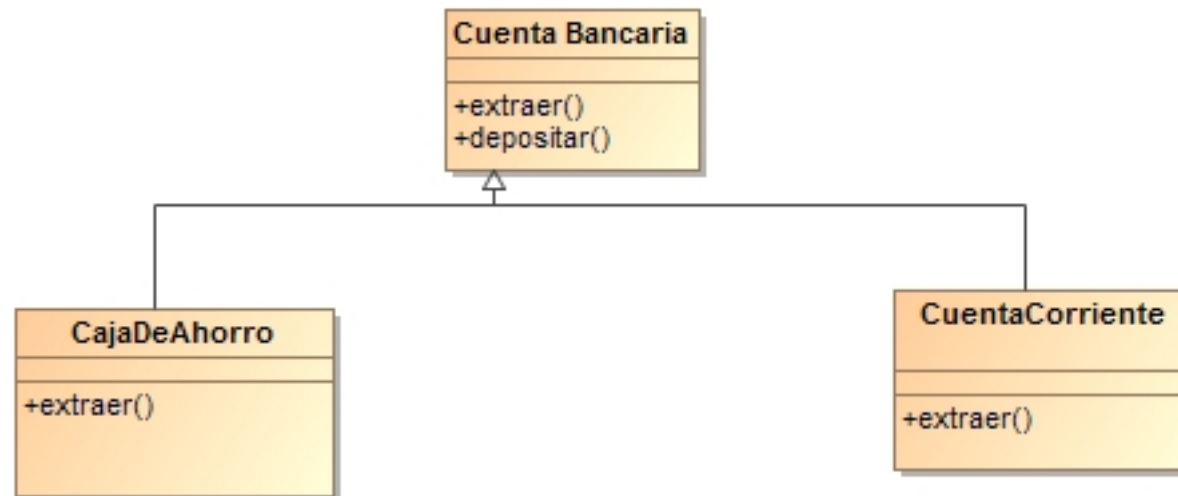


Pattern Composite

- ✓ Hace mas fácil el agregado de nuevos tipos de componentes porque los clientes no tienen que cambiar cuando aparecen nuevas clases componentes.
- ✓ Puede hacer difícil restringir las estructuras de composición cuando hay algún tipo de conflicto (por ejemplo ciertos compuestos pueden armarse solo con cierto tipo de atómicos)

Problema

- ✓ Supongamos una jerarquía de cuentas bancarias y una operación con “variantes” de acuerdo a la cuenta



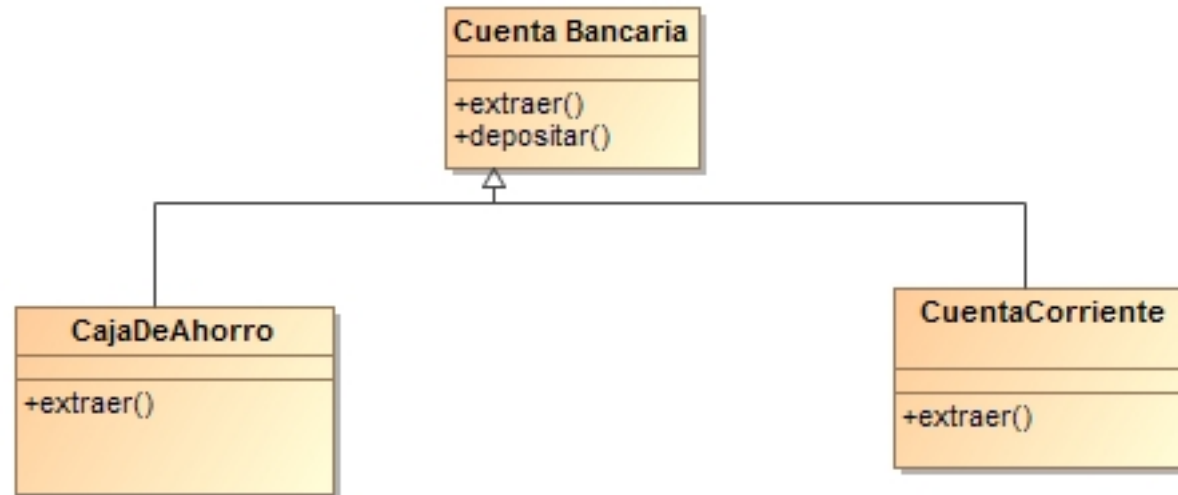


El Metodo Extraer

- ✓ En la Clase Caja de Ahorro tiene que controlar el saldo contra 0 y la cantidad de extracciones
- ✓ En la Clase Cuenta Corriente tiene que controlar el saldo contra un “rojo” permitido y la situacion impositiva del cliente



Solucion



extraer: cant
If `cantExtracciones < 5` and `saldo > cant` then `saldo:=saldo-cant`

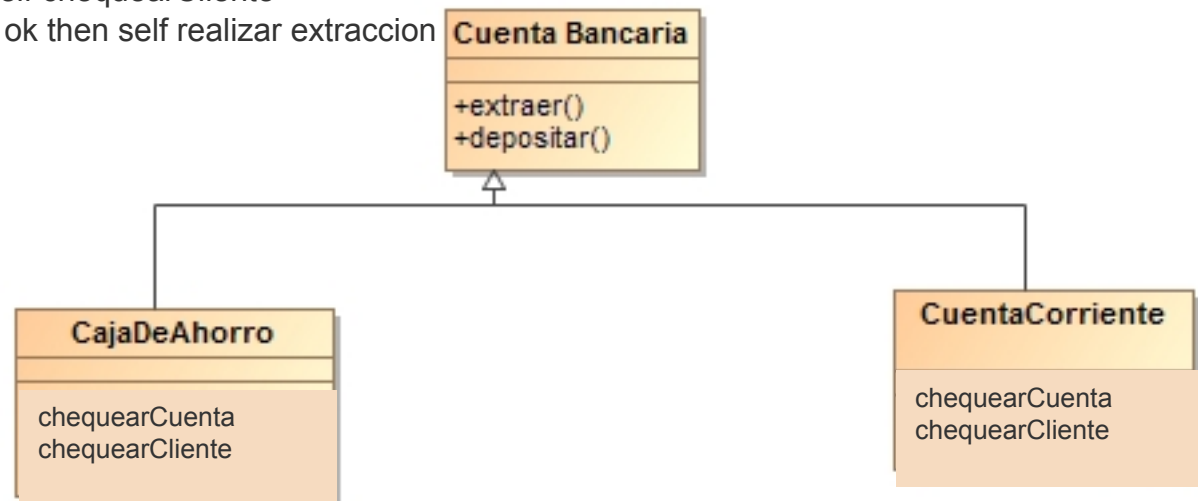
extraer: cant
If `saldo-cant > rojoPermitido` and `cliente pagoImpuestos` then `saldo: saldo -cant`

Problemas con esta Solucion



Template Method

Extraer: cant
Self chequearCuenta
Self chequearCliente
If ok then self realizar extraccion





Template Method

✓ Intent:

- ✓ Definir el esqueleto de un algoritmo en un metodo, difiriendo algunos pasos a las subclases. El template method permite que las subclases re definas ciertos aspectos de un algoritmo sin cambiar su estructura

✓ Aplicabilidad

- ✓ Para implementar las partes invariantes de un algoritmo una vez y dejas que las sub-clases implementen los aspectos que varian

✓ Structure

