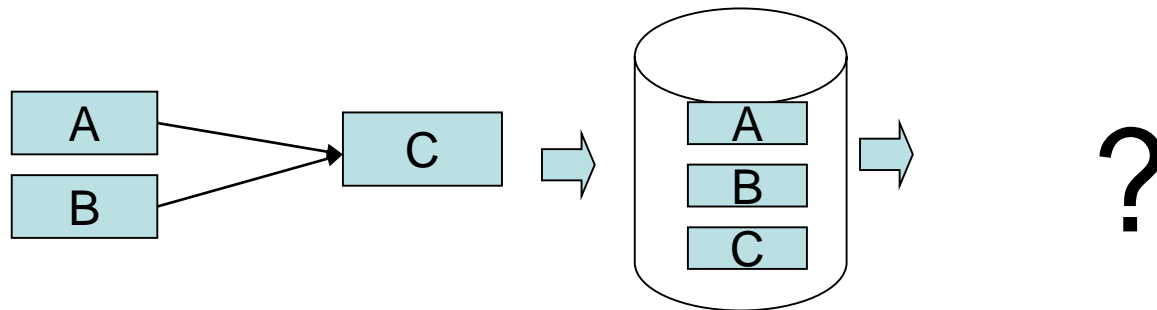


# Persistencia

Clase 2

# Conceptos Importantes

- Identidad de los objetos
  - `A=1, B= 1; A == B?`
  - `A=new Integer(1), B=new Integer(1); A==B?`
- Las Instancias son únicas
- Las referencias pueden ser múltiples



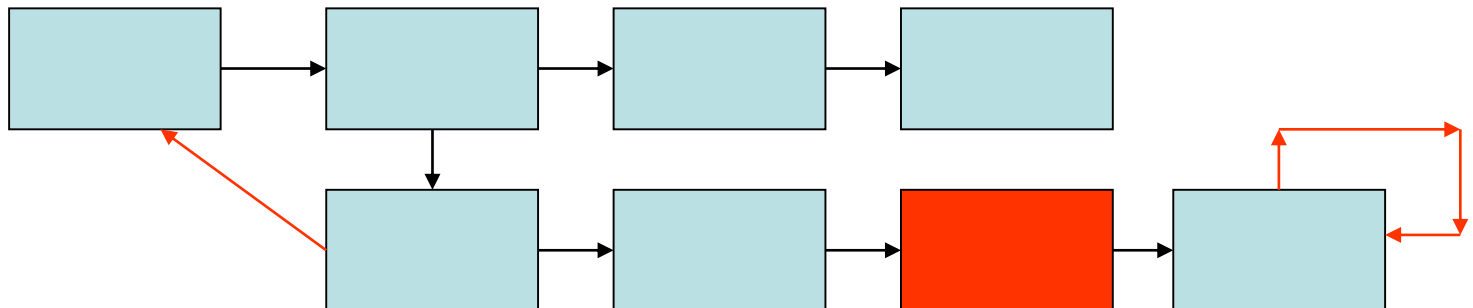
- Es necesario definir un equivalente a la referencia para poder persistir las relaciones entre objetos

# Conceptos Importantes

- Responsabilidad
  - La clase sabe persistirse
    - En contra del principio de responsabilidad única
  - Un agente externo se encarga de persistir
    - Toma los objetos y los examina
    - Necesita información externa sobre qué persistir y como
    - Necesita acceder al estado interno del objeto (posible problema con el encapsulamiento)

# Conceptos Importantes

- Orden de Persistencia
  - Por donde empiezo
  - Loops y Referencias Circulares
  - ¿Qué pasa si en el camino hay objetos no persistibles?



# Persistencia por Transformación

- Persistencia implica mantener un objeto más allá de su contexto de creación.
- Requiere un repositorio destino (archivo, sistema, base de datos), externo al proceso donde se origina el objeto.
- Es necesaria una transformación para pasar el objeto al repositorio persistente.
- Existen dos alternativas:
  - Persistencia por transformación explícita: El desarrollador debe configurar la transformación.
  - Persistencia por transformación implícita: El desarrollador no debe configurar la transformación.

# Transformación de Objetos

- Los componentes que participan del proceso de transformación son los siguientes:
  - Objeto
  - Clase
  - Protocolo de transformación
  - Descriptor de la transformación
  - Repositorio destino
  - Componente destino
  - Herramientas de manipulación

# Transformación de Objetos

- Objeto:
  - Relaciones con otros objetos (cardinalidad)
  - ¿Qué campos se persisten?
  - ¿Cómo se identifica?
  - ¿Qué características debe tener?
  - ¿Cómo se manejan grafos con ciclos?
  - ¿Qué ocurre con el polimorfismo?

# Transformación de Objetos

- Clase:
  - ¿Qué ocurre con la herencia?
  - ¿A qué versión de la clase corresponde el objeto?
  - ¿Qué campos se persisten?
  - ¿Cómo se crean nuevos objetos?



# Transformación de Objetos

- Protocolo de transformación:
  - ¿Binario o de texto?
  - ¿Define el language de manipulación?
  - ¿Es multiplataforma?
  - ¿Estándar público o propietario?
  - ¿Es nativo o no nativo?
  - ¿Es implícito o explícito?

# Transformación de Objetos

- Descriptor de transformación (depende del protocolo):
  - ¿Binario o de texto?
  - ¿Incluído en el código fuente o externo?
  - ¿Permite múltiples esquemas para una misma clase?
  - ¿Es multiplataforma?

# Transformación de Objetos

- Repositorio destino:
  - ¿De qué tipo es (archivo/BDR/BDOO)?
  - ¿Permite manipulación estándar?
  - ¿Qué características de desempeño tiene?

# Transformación de Objetos

