# Patrones de Diseño

# Patrones de Diseño

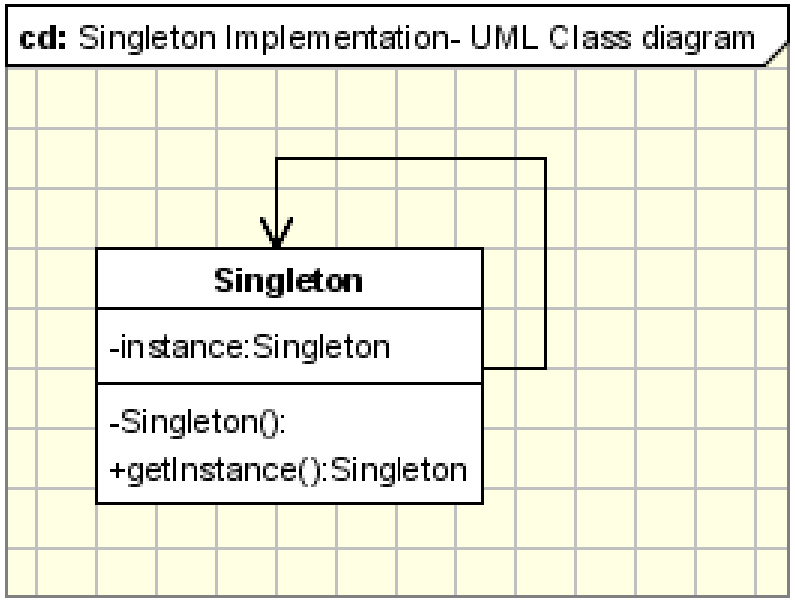
- Un patrón de diseño es una solución a un problema de diseño.
- Para ser considerado un patrón, debe poseer ciertas características.
  - Se debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores
  - Debe ser reutilizable, o sea, debe ser aplicable a diferentes problemas de diseño en distintas circunstancias

# Patrones de Diseño

- Patrones creacionales
  - Corresponden a patrones de diseño software que solucionan problemas de creación de instancias. Nos ayudan a encapsular y abstraer dicha creación
- Patrones estructurales
  - Son los patrones de diseño software que solucionan problemas de composición (agregación) de clases y objetos
- Patrones de comportamiento
  - Se definen como patrones de diseño de software que ofrecen soluciones respecto a la interacción y responsabilidades entre clases y objetos

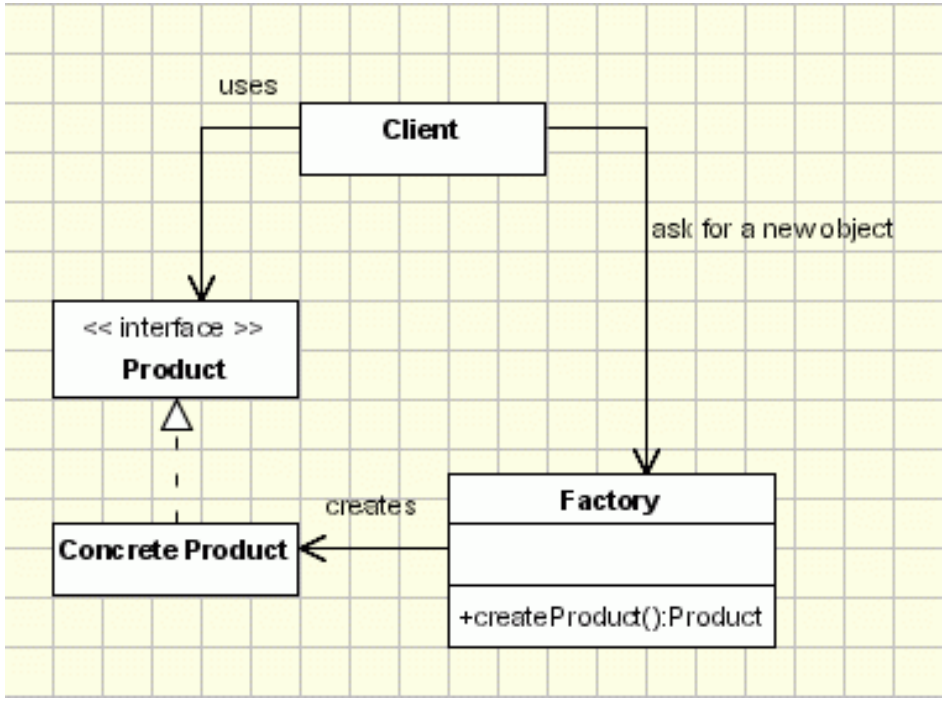
# Patrones creacionales - **Singleton**

- Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia
- Restringe la instanciación de una clase o valor de un tipo a un solo objeto
- La implementación implica un miembro estático en la clase 'Singleton', un constructor privado y un método publico que devuelve una referencia al miembro estático



# Patrones creacionales - **Factory**

- Crea un objeto sin exponer la lógica de instanciación al cliente
- Se refiere al nuevo objeto creado a través de una interfaz común



# Sistema de Archivos

Manipulando Archivos

# Streams - ¿Qué son?

- Un ‘stream’ es una abstracción que representa un dispositivo en el cual se producen operaciones de entrada y salida.
- Básicamente se puede definir como una fuente o destino de caracteres con longitud indefinida.
- Los ‘stream’ generalmente se asocian con fuentes o destinos físicos, como un archivo en disco, un teclado, o la pantalla.

# Manejo de Archivos en Java

- Java provee dos clases dentro de la librería **java.io** para el manejo de archivos:
  - **FileInputStream** - permiten leer una serie secuencial de caracteres en un archivo de a 1 byte por vez
    - Read
  - **FileOutputStream** - permiten escribir una serie secuencial de caracteres en un archivo de a un byte por vez
    - Write
  - **FileReader** - permiten leer una serie secuencial de caracteres en un archivo de a 2 bytes por vez
    - Read
  - **FileWriter-** permiten escribir una serie secuencial de caracteres en un archivo de a 2 bytes por vez
    - Write

# Manejo de Archivos en Java

- Java provee adicionalmente el Scanner para el manejo de archivos grandes de una forma compacta
  - Solo permite lectura

CAPTTON
IT CONSULTING