

Estrategia (Strategy)

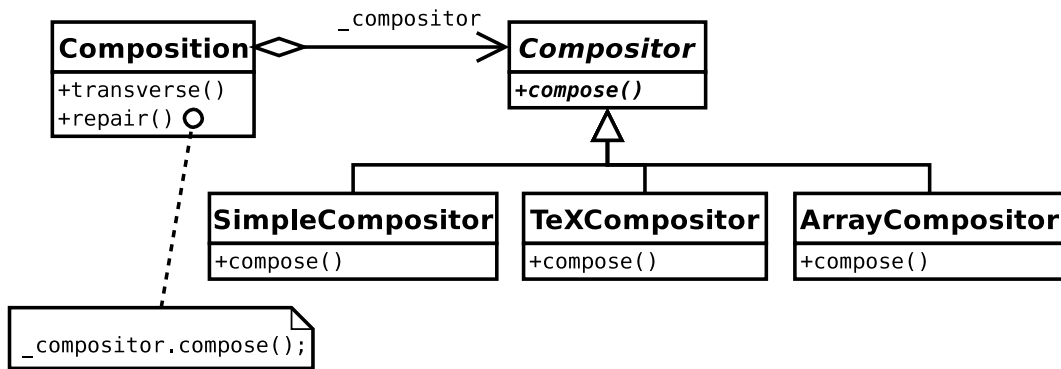
- *Patrón de Comportamiento*

- *Propósito*

Encapsula y hace intercambiables a una familia de algoritmos

- *Motivación*

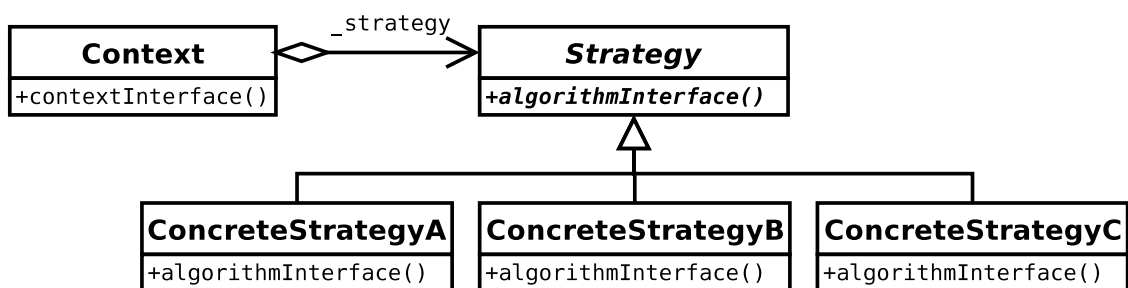
- Diversos algoritmos para particionar un texto en líneas
- Deseable separar clientes de los algoritmos de partición:
 - Los clientes que incluyen el código para particionar en líneas son más grandes y más difíciles de mantener
 - No todos los algoritmos son necesarios en todos los casos
 - Difícil añadir nuevos algoritmos o modificar los existentes si el algoritmo forma parte del cliente
- Solución: Definir clases que encapsulan a las diferentes *estrategias* de particionamiento.



■ *Aplicabilidad*

- Muchas clases relacionadas se diferencian únicamente en su comportamiento
- Necesarias distintas variantes de un algoritmo
- Un algoritmo utiliza información que los clientes no deberían conocer. El patrón evita la exposición de estructuras de datos dependientes del algoritmo
- Una clase define múltiples comportamientos mediante una serie de instrucciones condicionales.

■ *Estructura*



■ *Participantes*

- Contexto (Context)
Configurado con una estrategia concreta
Puede definir una interfaz que permita a la estrategia el acceso a sus datos
- Estrategia (Strategy)
Declara una interfaz común para todos los algoritmos soportados
El contexto utiliza esta interfaz para invocar el algoritmo definido por una estrategia concreta
- Estrategia Concreta (ConcreteStrategy_{*i*})
Implementa el algoritmo utilizando la interfaz definida por la estrategia

■ *Colaboraciones*

- Es necesario el intercambio de información entre Estrategia y Contexto.
 - Parámetros de los métodos de la estrategia
 - En ocasiones, como ocurría en el patrón Estado, el contexto se pasa a sí mismo a la estrategia.
- Los clientes del contexto normalmente configuran a éste con una estrategia concreta. A partir de ahí, sólo se interactúa con el contexto.

■ *Consecuencias*

- La herencia puede ayudar a factorizar las partes comunes de las familias de algoritmos
- Alternativa a la extensión de contextos (independencia, cambio dinámico)
- Reducción de instrucciones condicionales
- Diferentes opciones de implementación
- Clientes deben conocer las diferentes estrategias
- Penalización en la comunicación entre estrategia y contexto
- Incremento del número de objetos

■ *Implementación*

- Definición de la interfaz entre estrategia y contexto
 - Pasar como parámetro la información necesaria para la estrategia
 - Pasar como parámetro el contexto y dejar que la estrategia explícitamente pida la información que necesita
 - Mantener en la estrategia una referencia al contexto
- Objetos estrategia opcionales