

Método Plantilla (Template Method)

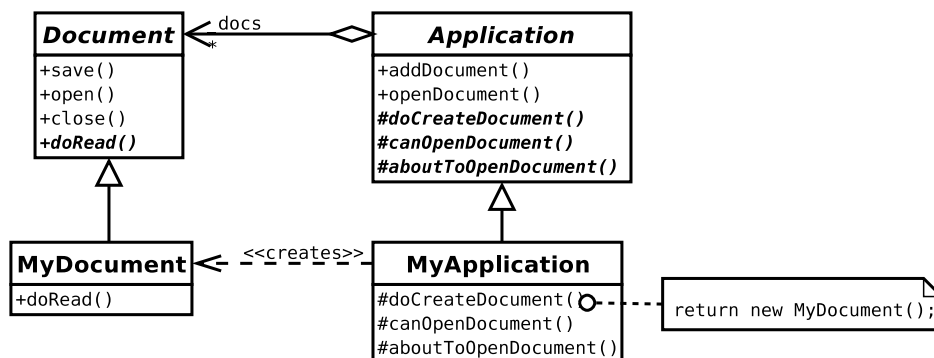
- *Patrón de Comportamiento*

- *Propósito*

Define el esqueleto de un algoritmo en una operación, pero difiere algunos pasos a las subclases.

- *Motivación*

- *Framework* para aplicaciones que permiten presentar múltiples documentos al usuario (abstracciones Aplicación y Documento)
- Para una aplicación específica, se especializan los conceptos de Aplicación y Documento
- Problema: La abstracción aplicación conoce la operación genérica a realizar (por ejemplo, abrir un documento), pero parte de la operación depende del documento concreto
- Solución: Implementar un método que difiera en otros métodos abstractos las partes del algoritmo que no se conocen o que son específicas



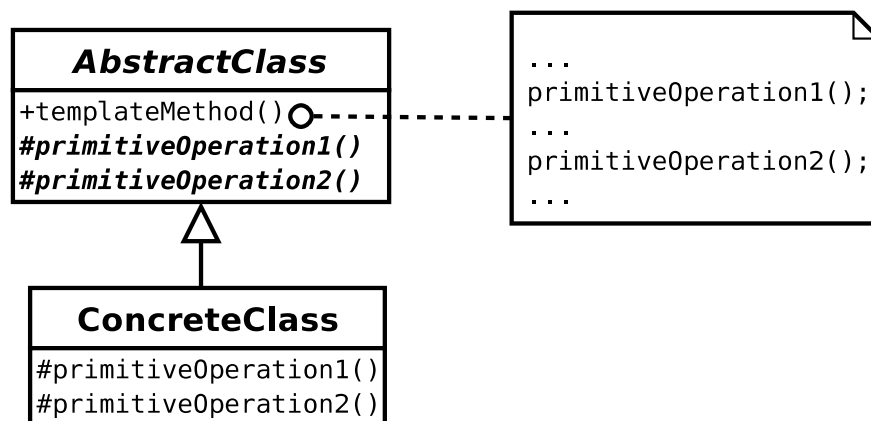
■ *Motivación (cont.)*

- Un *método plantilla* es un método que define un algoritmo en base a una serie de operaciones abstractas que son redefinidas por las subclases para obtener un comportamiento concreto

■ *Aplicabilidad*

- Implementar las partes invariables de un algoritmo una única vez y permitir que las subclases definan el comportamiento que cambia
- Factorización de comportamiento común de las subclases en la superclase (evitar replicación de código mediante generalización)
- Para controlar la extensiones de las subclases. El método plantilla utiliza métodos especiales (*hooks*) que son los únicos puntos que pueden ser redefinidos

■ *Estructura*



■ *Participantes*

- Clase Abstracta (AbstractClass)

Define *operaciones primitivas* (normalmente abstractas) que las subclases implementan

Implementa un método plantilla que define el esqueleto de un algoritmo y que utiliza las operaciones primitivas

- Clase Concreta (ConcreteClass)

Implementa las operaciones primitivas que definen el comportamiento específico del algoritmo para las subclases

■ *Colaboraciones*

Las clases concretas confían en la clase abstracta la responsabilidad de la parte invariable del algoritmo

■ *Consecuencias*

- Fundamental para la reutilización de código

- Invierte el control: la superclase es la encargada de llamar a las operaciones definidas en las subclases (*no nos llame, ya le llamaremos*)

- Distinción entre:

- Operación primitiva (implementada en la subclase)

- Operaciones de enganche (*Hooks*) (proporcionan código por defecto que puede refinarse en las subclases si es necesario)

■ *Patrones Relacionados*

- Métodos de fabricación normalmente utilizados por los métodos plantilla

- Estrategia vs. Método Plantilla