UNTDF
Universidad Nacional de Tierra del Fuego,
Antártida e Islas del Atlántico Sur

Patrón Template Method

motivación

- Definir el esqueleto de un algoritmo en una operación, delegando algunos pasos a las subclases. El Template Method permite a las subclases redefinir ciertos pasos de un algoritmo sin cambiar la estructura del algoritmo.

- La clase base declara los 'comodines' (placeholders) del algoritmo, y las clases derivadas implementan los comodines.

UNTDF
Universidad Nacional de Tierra del Fuego,
Antártida e Islas del Atlántico Sur

Patrón Template Method

Problema

Dos componentes diferentes tienen similitudes significativas, pero demuestran no reutilizar interfaces comunes o implementación. Si es necesario un cambio común a ambos componentes, se duplica el esfuerzo.

UNTDF
Universidad Nacional de Tierra del Fuego,
Antártida e Islas del Atlántico Sur

Patrón Template Method

Discusión

El diseñador del componente decide que pasos de un algoritmo son invariantes (o estandar), y cuales son variantes (o personalizables). Los pasos invariantes son implementados en la clase base abstracta, mientras que los pasos variantes se les puede dar una implementación por defecto o ninguna implementación. Los pasos variantes representan "ganchos", o "comodines", que pueden, o deben, ser provistos por los clientes del componente en una clase concreta derivada.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Patrón Template Method

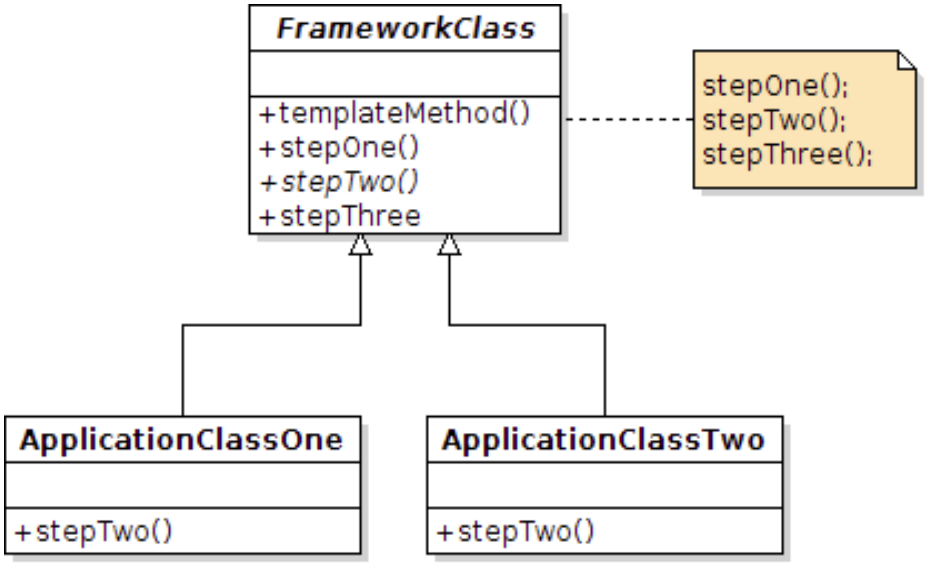
Discusión (cont.)

El diseñador del componente gobierna los pasos requeridos de un algoritmo, y el ordenamiento de los pasos, pero permite a los clientes del componente extender o reemplazar algunos de estos pasos.
El Template Method es muy usado en frameworks. Cada framework implementa las partes invariantes de la arquitectura del dominio, y define "comodines" para todas las opciones necesarias o interesantes de las personalizaciones del cliente. Esta estructura de inversión de control ha sido llamada afectuosamente "el principio Hollywood" – "no nos llame, nosotros lo llamaremos".

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Patrón Template Method

Estructura

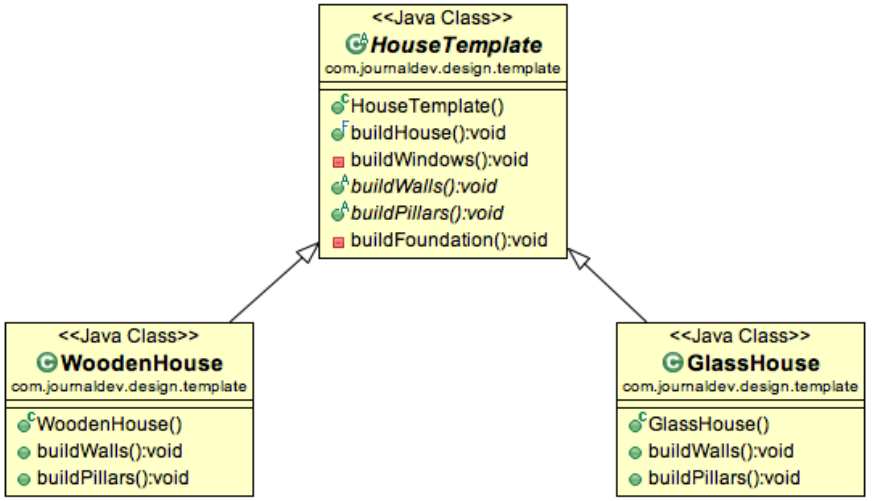


La implementación de template_method() es: llamar step_one(), llamar step_two(), y llamar step_three(). step_two() es un método “gancho”(placeholder). Es declarado en la clase base, y luego definido en las clases derivadas. Frameworks (infraestructuras de reuso a gran escala) utilizan mucho el Template Method. Todo el código reusable es definido en las clases bases del framework, y entonces los clientes del framework son libres para definir personalizaciones creando clases derivadas cuando sea necesario.

UNTDF
Universidad Nacional de Tierra del Fuego,
Antártida e Islas del Atlántico Sur

Patrón Template Method

Ejemplo



El Template Method define un esqueleto de un algoritmo en una operación, y delega algunos pasos a las subclases. Los constructores usan el Template Method cuando construyen un barrio. Un barrio típico consiste en un número de casas con diferentes variaciones en cada una de ellas. El plano, los cimientos, plomería y cableado será idéntico para cada casa. La variaciones son introducidas en las últimas etapas de construcción para producir una gran variedad de modelos.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Patrón Template Method

```
public abstract class HouseTemplate {
//template method, final para que las subclases no lo puedan
sobreescribir
 public final void buildHouse(){
 buildFoundation();
 buildPillars();
 buildWalls();
 buildWindows();
 System.out.println("House is built.");
}

//implementación por defecto
private void buildWindows() {
 System.out.println("Building Glass Windows");
}
private void buildFoundation() {
 System.out.println("Building foundation with
cement,iron rods and sand");
}
//metodos a ser implementados por las subclases
public abstract void buildWalls();
public abstract void buildPillars();
}
```

UNTDF
Universidad Nacional de Tierra del Fuego,
Antártida e Islas del Atlántico Sur

Patrón Template Method

```
public class WoodenHouse extends HouseTemplate {

@Override
public void buildWalls() {
 System.out.println("Building Wooden Walls");
}

@Override
public void buildPillars() {
 System.out.println("Building Pillars with Wood
coating");
}

}
```

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Patrón Template Method

```
public class GlassHouse extends HouseTemplate {

@Override
public void buildWalls() {
 System.out.println("Building Glass Walls");
}

@Override
public void buildPillars() {
 System.out.println("Building Pillars with glass
coating");
}

}
```

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Patrón Template Method

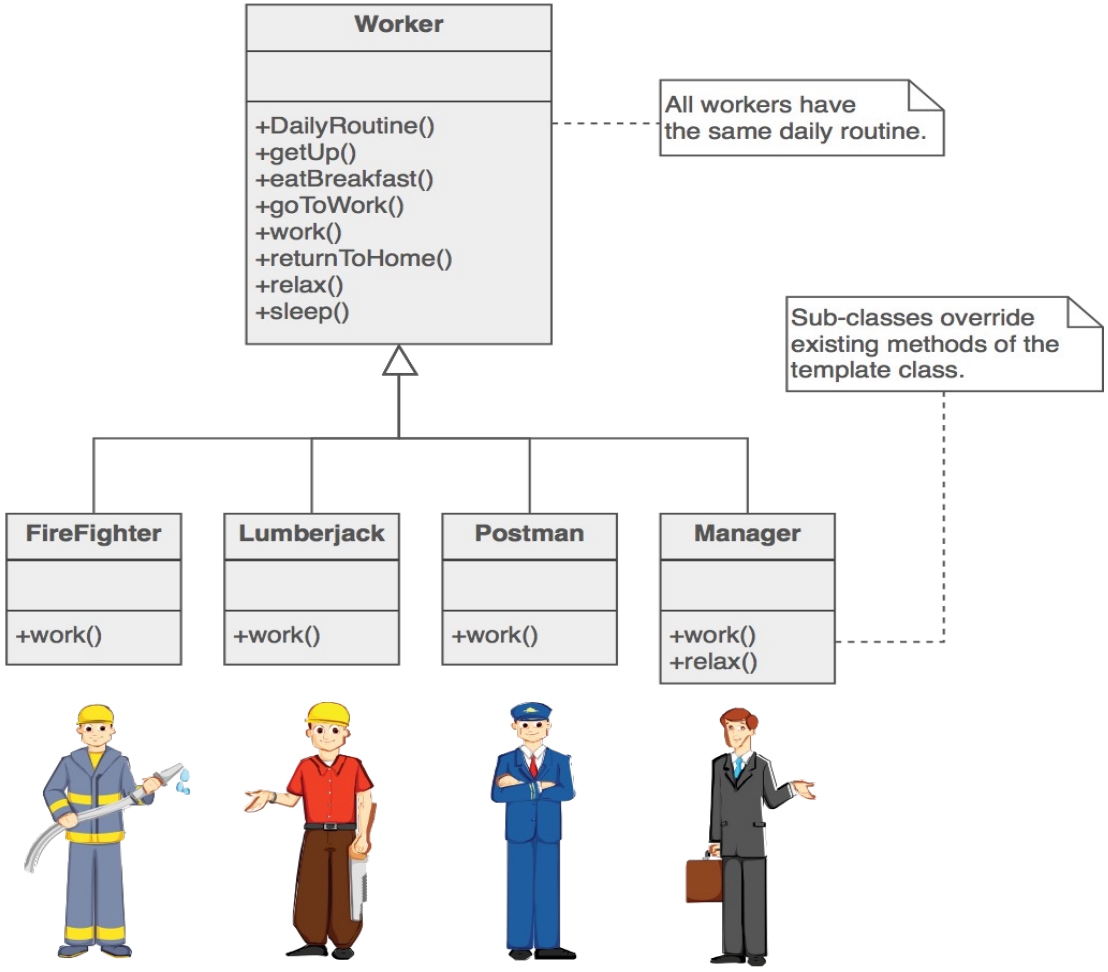
```
public class HousingClient {

public static void main(String[] args) {
 HouseTemplate houseType = new WoodenHouse();
 //utilizando el template method
 houseType.buildHouse();
 System.out.println("************");
 houseType = new GlassHouse();
 houseType.buildHouse();
}
}
```

```
Building foundation with cement,iron rods and sand
Building Pillars with Wood coating
Building Wooden Walls
Building Glass Windows
House is built.
************
Building foundation with cement,iron rods and sand
Building Pillars with glass coating
Building Glass Walls
Building Glass Windows
House is built.
```

UNTDF
Universidad Nacional de Tierra del Fuego,
Antártida e Islas del Atlántico Sur

Patrón Template Method

Otro ejemplo: rutina diaria de un trabajador



UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Patrón Template Method

Check list

1) Examine cada algoritmo, y decida que pasos son estándar y cuales son particulares de la clase correspondiente.
2) Defina una nueva clase base abstracta para alojar el "no nos llame, nosotros lo llamaremos".
3) Mueva la cascara del algoritmo (ahora llamado "template method") y la definición de todos los pasos estándar a la clase base.
4) Defina un método comodín o "gancho" en la clase base para cada paso que requiera muchas diferentes implementaciones. Este método puede contener una implementación por defecto o puede ser definido abstract (Java) o pure virtual (C++).
5) Invoque los métodos “gancho” desde el template method.
6) Cada una de las clases existentes declara una relación “es-una” con la nueva clase base abstracta.
7) Remover de las clases existentes todos los detalles de implementación que hayan sido movidos a la clase base.
8) Los únicos detalles que seguirán existiendo en las clases existentes serán los detalles de implementación particulares a cada clase derivada.

UNTDF
Universidad Nacional de Tierra del Fuego,
Antártida e Islas del Atlántico Sur

Patrón Template Method

Patrones relacionados

- Strategy es como el Template Method excepto en su granularidad.
- Template Method usa herencia para variar parte del algoritmo. Strategy usa delegación para variar un algoritmo entero.
- Strategy modifica la lógica de objetos individuales. Template Method modifica la lógica de una clase completa.
- Factory Method es una especialización de Template Method.