Template

Objective

Define the structure of an algorithm in an operation by allowing subclasses to redefine certain steps without changing the structure of the algorithm.

Function

Redefine subclasses by following certain steps of an algorithm without changing the structure of the algorithm.

Structure

As shown in figure 1

- Client: It's the component that drives the execution of the template.

- AbstractTemplete: Abstract class with a series of operations that define the steps to carry out the execution of the algorithm. The class has the templeteMethod that executes in order the methods step1, step2, step3.

- Implementation: A class that represents a concrete temper, for which it must inherit from AbstractTemplete and implement its methods.

The structure that meets this pattern is shown in Figure 1

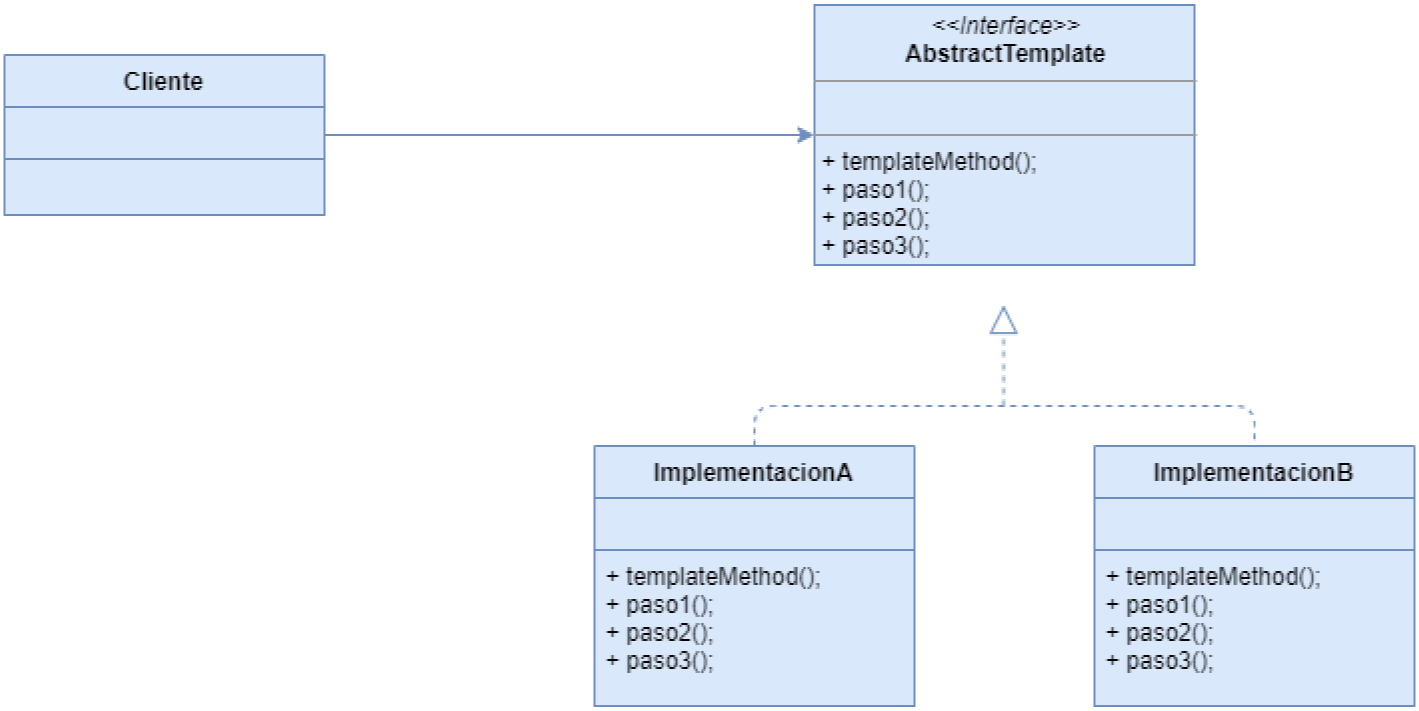

Figure 1: UML Diagram Template method Pattern

Applications

The use of the Template pattern is recommended when:

- It is required to implement the invariable part of an algorithm in the class and to delegate the variable parts to the subclasses.
- A common behavior between subclasses should be factored and located in a common class avoiding duplicate code.
- The application needs to control the extensions of the subclasses.

Design Patterns Collaborators

- The Template pattern presents a similar performance to the Strategy pattern; using the inheritance to vary the part of an algorithm.

Scope of action

Applied at the object level.

Problem

More effort is needed to provide a common interface to components that have come to have significant similarities; this would involve doubling the effort to be able to detail as much as possible the relationship between these components.

Solution

The Template pattern writes a parent class in which one or more methods to be implemented are assigned to the derived classes, formalizing the idea of defining an algorithm in a class; but delegating the implementation details to the subclasses.

Diagram or Implementation

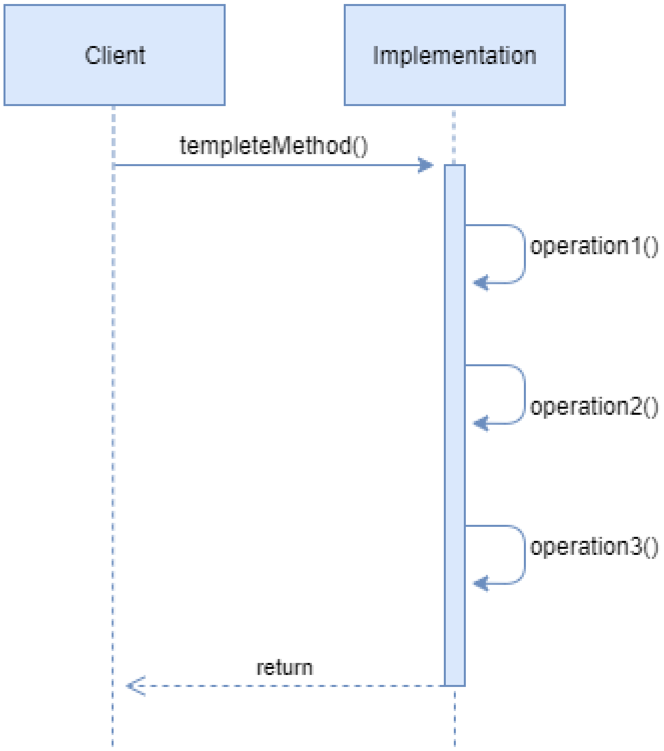

Figure 2: UML Diagram Template method Pattern

Figure 2 explains the behaviour of the pattern by means of a sequence diagram.

- The client class creates or obtains an instance of a template implementation.

- The client class runs the public templete method of the templete.

- The default implementation class of the templete method executes in order the methods step1, step2, step3.

- The implementation class returns a result.