**Template Method Design Pattern**

Template Method design pattern is used to get some steps of algorithm implemented in subclass.

**Intent**

* Define the skeleton of an algorithm in an operation, deferring some steps to subclasses. Template Method lets subclasses redefine certain steps of an algorithm without changing the algorithm structure.

The intent is to defer some steps of algorithm to subclass without changing the algorithm structure.

**Problem**

* The algorithm has to be generic but some steps are require to be defined by user.

There may be a situation where steps of algorithm are not known. For example, the framework has some steps of algorithm which are supposed to be provided by user. There may be many scenarios where some steps may vary but the algorithm is generic.

**Solution**

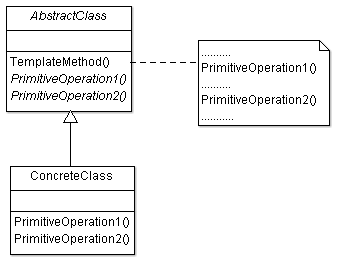
* Identify the invariant and variant part of the algorithm.
* Have the implementation of invariant part in abstract class and variant part in derived class.

So the algorithm structure is not changed and some steps are deferred to be implemented by subclasses.

**Where it is applicable?**

* The algorithm is generic with invariant and variant parts.
* The variant parts are required to be deferred to derived class.
* There is a need to provide extension points.

**Structure**



**Participant classes**

* **AbstractClass** has method TemplateMethod() which defines a generic algorithm. It also has abstract methods PrimitiveOperation1() and PrimitiveOperation2(). These methods are variant parts of generic algorithm().
* **ConcreteClass** implements the abstract methods PrimitiveOperation1() and PrimitiveOperation2().

**How they work together?**

* Client uses the TemplateMethod(). The generic algorithm defined in TemplateMethod() calls the method of Concrete class to perform the deferred steps.
* The invariant part of algorithm is executed in template method of abstract class and variant part of algorithm is executed in concrete class. This pattern is very useful for code reuse and generally used in class libraries and frameworks.