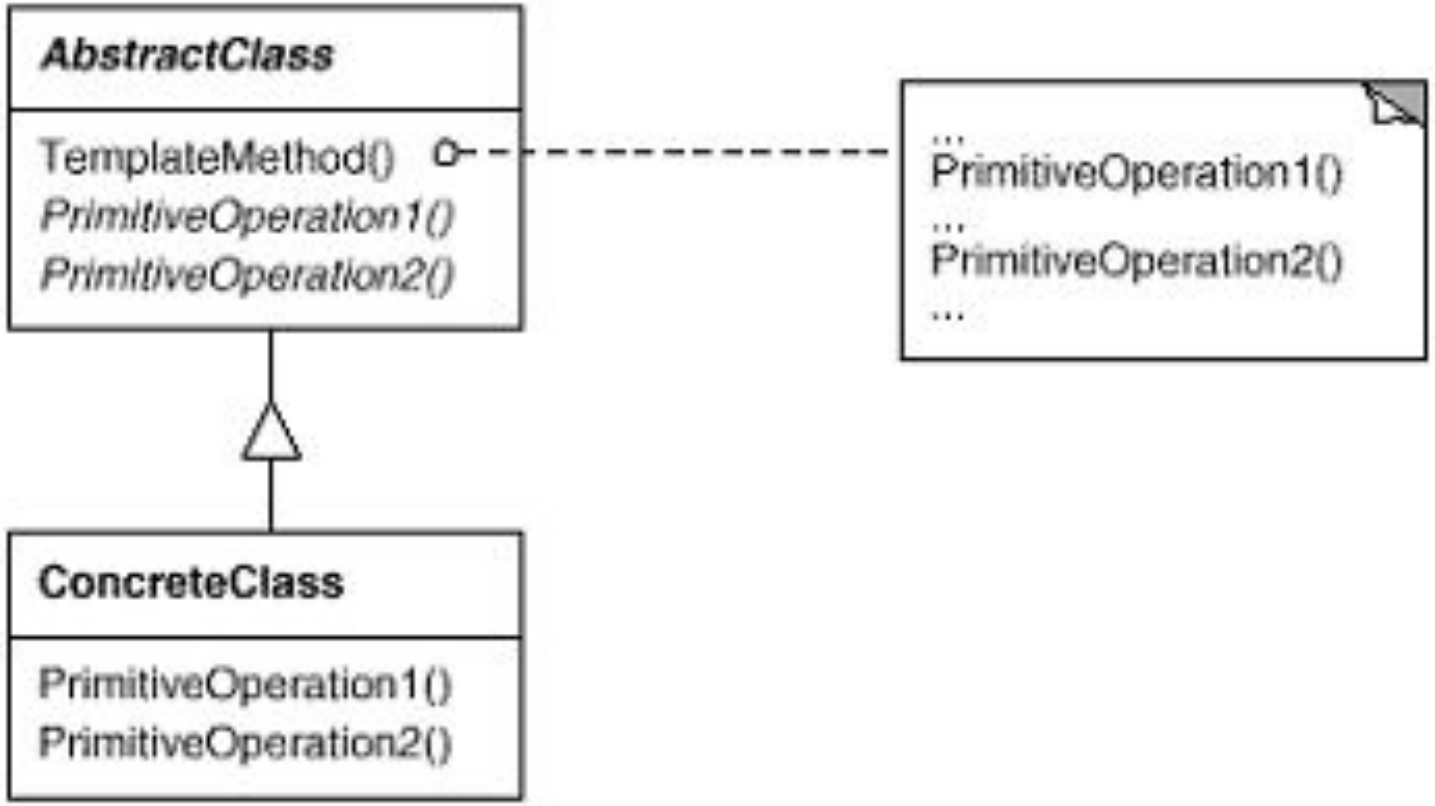
Template Method

Behavioral Pattern

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's structure.

Structure


AbstractClass
TemplateMethod()
PrimitiveOperation1()
PrimitiveOperation2()
...
PrimitiveOperation1()
...
PrimitiveOperation2()
...
ConcreteClass
PrimitiveOperation1()
PrimitiveOperation2()

Description

- The Template Method pattern suggests keeping the outline of the algorithm in a separate method referred to as a template method inside a class, which may be referred to as a template class, leaving out the specific implementations of the variant portions of the algorithm to different subclasses of this class.

Description

- The Template class does not necessarily have to leave the implementation to subclasses in its entirety.
- Instead, as part of providing the outline of the algorithm, the Template class can also provide some amount of implementation that can be considered as invariant across different implementations.
- It can even provide default implementation for the variant parts, if appropriate.
- Only specific details will be implemented inside different subclasses.
- This type of implementation eliminates the need for duplicate code, which means a minimum amount of code to be written.

Template Class defined in 2 ways

1. Abstract Class

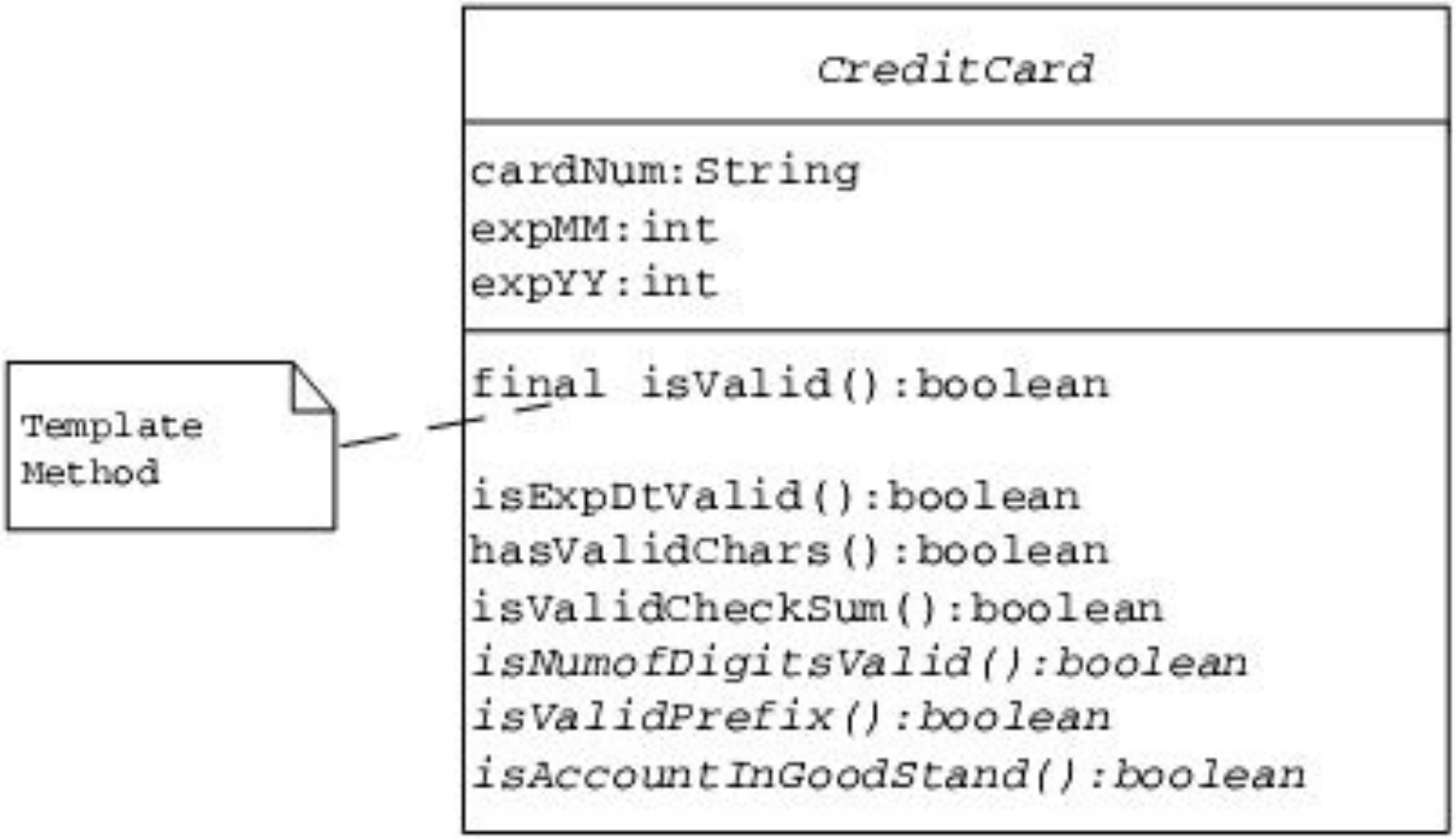
The set of variant steps can be designed as abstract methods. Specific implementations can be provided for these abstract methods inside a set of concrete subclasses of the abstract Template class.

Template Class defined in 2 ways

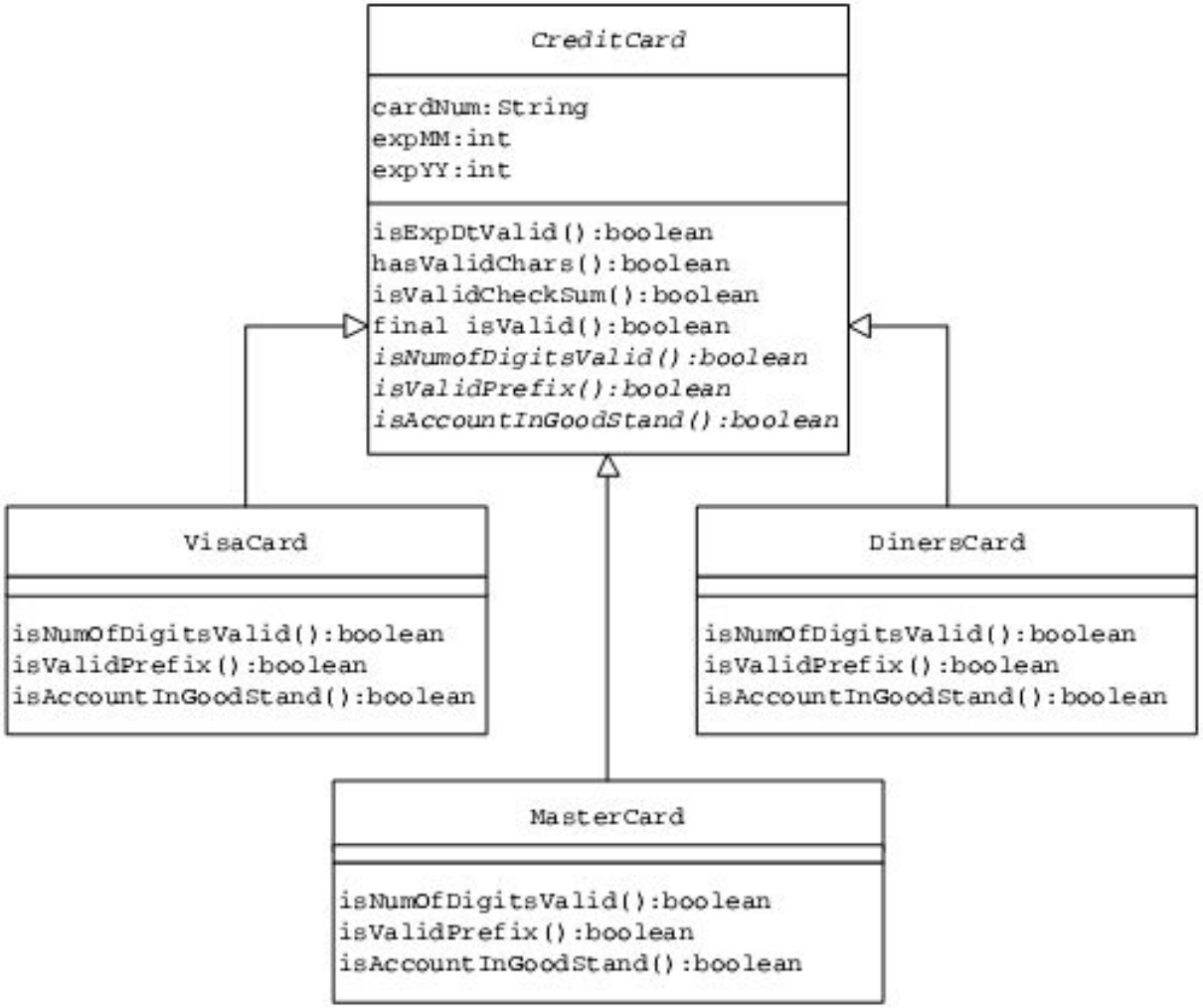
2. Concrete Class

The set of variant steps can be designed as non-abstract methods with the default implementation. Subclasses of the Template class can override these methods to provide specific implementations without altering the outline of the algorithm.

Example - to check the validity of a given credit card



Class Hierarchy



Try…

- Identify how the Template Method pattern is used when you design an applet with custom code in any of the applet life-cycle methods (init, start, paint, stop and destroy).