Template ID

Student:

Izabela Kuźniar

Teacher:

Andrea Corradini

Course:

Software Design Patterns

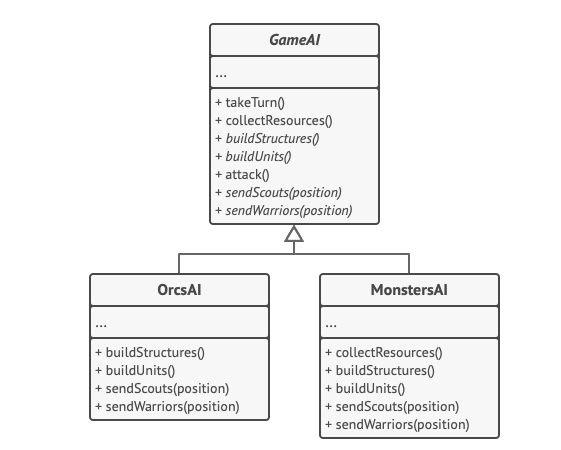
# Name and category

Template is a behavioral pattern.

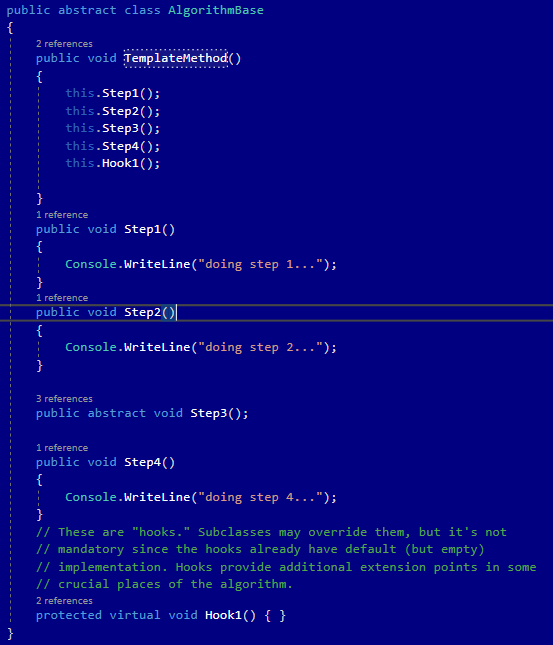
# Intent:

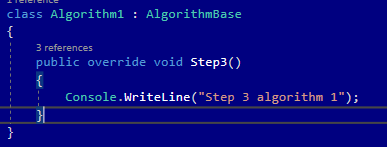
Template defines a skeleton of an algorithm in superclass but it let’s override some steps in subclasses.

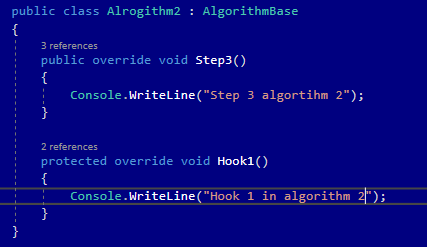
# Structure as a UML class diagram

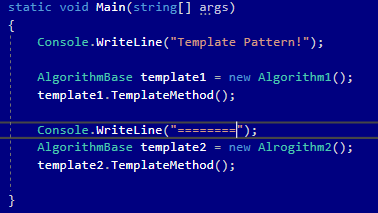


# Implementation:









# Consequences:

Benefits:

* Ability to let the client override specific steps for the big algorithm, this makes them less affected by changes that happen to other parts of the algorithm.
* Possibility to pull duplicate code into a superclass.

Drawbacks:

* Some clients could be limited by the skeleton that was provided.
* You might violate the Liskov Substitution Principle by suppressing a default step implementation via a subclass.
* Template methods tend to be harder to maintain the more steps they have.

# Known uses

* In Java JDK: java.util.Collections and java.util.Arrays provide a sorting algorithm as a templet method

# Related patterns

1. Factory Method is a specialization of Template Method. At the same time, a Factory Method may serve as a step in a large Template Method.
2. Template Method is based on inheritance: it let’s change parts of the algorithm in the subclasses. Strategy is based on composition: it let’s change object’s behavior by supplying it with different strategies that correspond to the behavior. Template Method works at the class level, so it’s static. Strategy works on the object level, letting you switch behaviors at runtime.