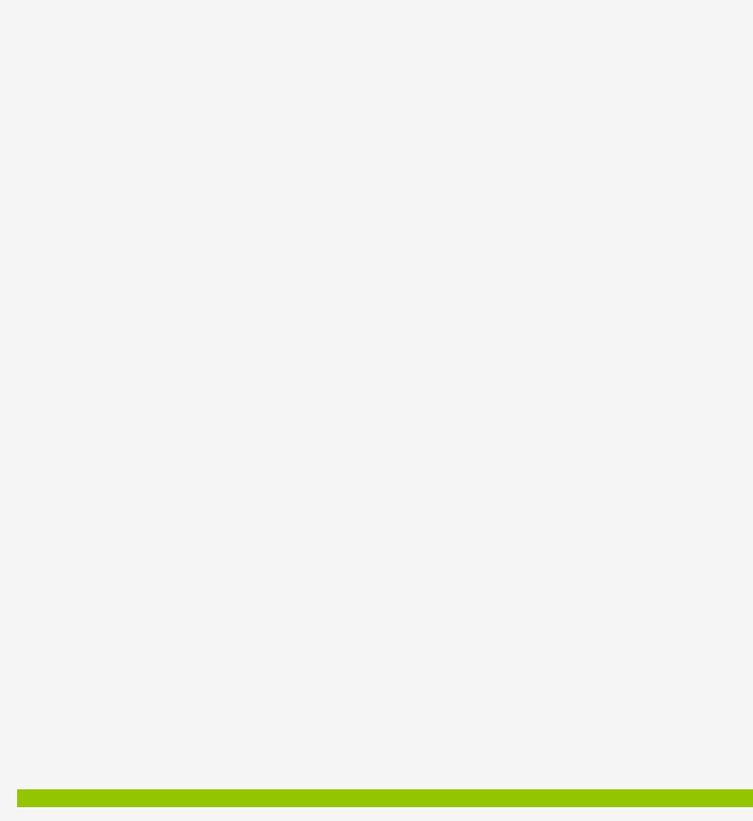


# Паттерн Template Method



# Название и классификация

- Template Method (-шаблонный метод) – поведенческий шаблон проектирования, определяющий основу алгоритма и позволяющий наследникам переопределять некоторые шаги алгоритма, не изменяя его структуру в целом.

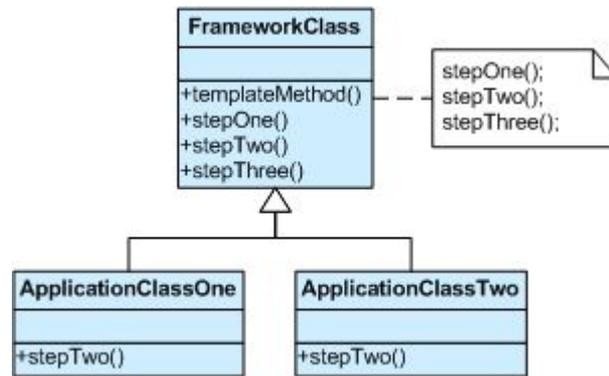
# Назначение

- ▣ Шаблонный метод определяет основу алгоритма и позволяет подклассам переопределить некоторые шаги алгоритма, не изменяя его структуру в целом

# Применимость

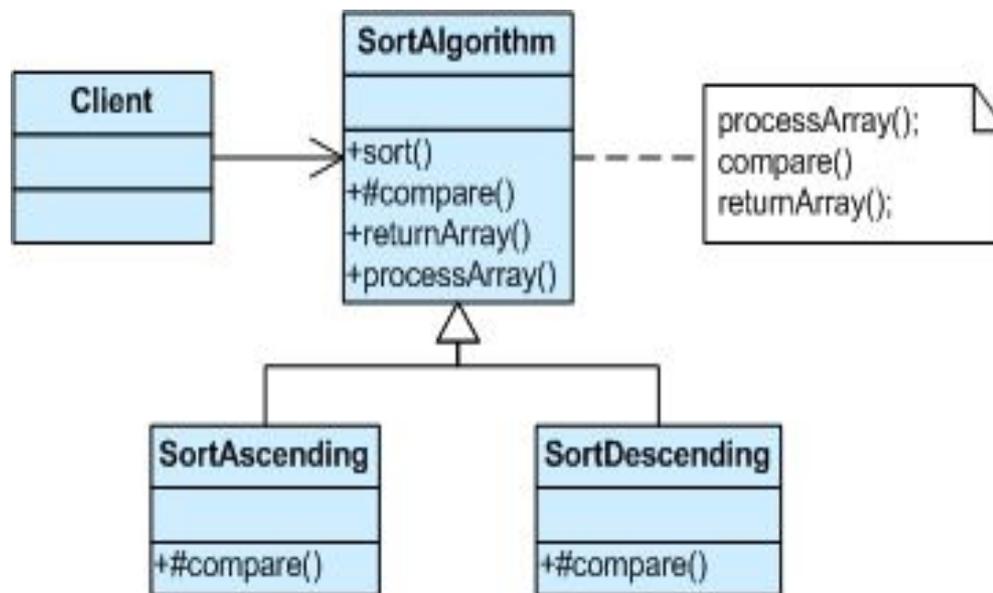
- Однократное использование инвариантной части алгоритма
- Избежание дублирования кода
- Управление расширениями подклассов

# UML-диаграмма классов паттерна Template Method



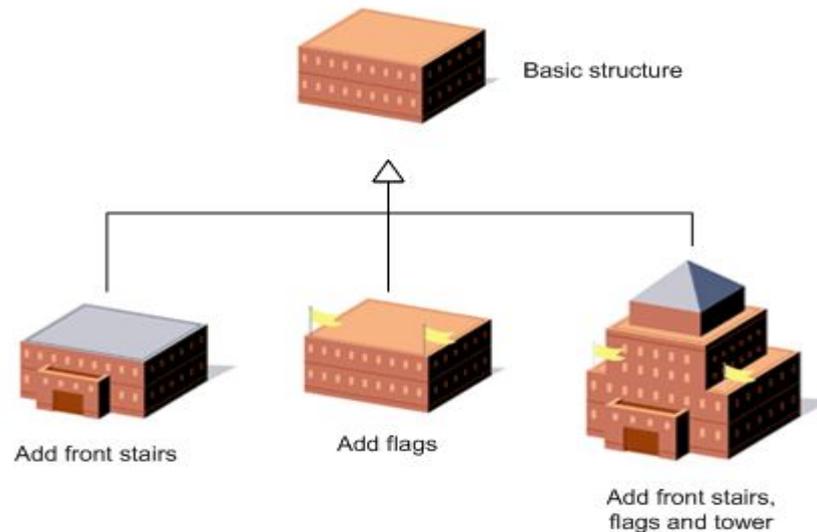
- Реализация метода `templateMethod()` вызывает методы `stepOne()`, `stepTwo()` и `stepThree()`. Метод `stepTwo()` является "замещающим" методом. Он объявлен в базовом классе, а определяется в производных классах. Каркасы приложений широко используют паттерн Template Method. Весь повторно используемый код определяется в базовых классах каркаса, нужное поведение системы клиенты определяют в создаваемых производных классах.

# UML-диаграмма классов паттерна Template Method



# Пример паттерна Template Method

- Паттерн Template Method определяет основу алгоритма и позволяет подклассам изменить некоторые шаги этого алгоритма без изменения его общей структуры. Строители зданий используют шаблонный метод при проектировании новых домов. Здесь могут использоваться уже существующие типовые планы, в которых модифицируются только отдельные части.



# Использование паттерна Template Method

1. Исследуйте алгоритм и решите, какие шаги являются стандартными, а какие должны определяться подклассами.
2. Создайте новый абстрактный базовый класс, в котором будет реализован принцип "не звоните нам, мы сами вам позвоним".
3. Поместите в новый класс основу алгоритма (шаблонный метод) и определения стандартных шагов.
4. Для каждого шага, требующего различные реализации, определите "замещающий" виртуальный метод. Этот метод может иметь реализацию по умолчанию или быть чисто виртуальным.
5. Вызовите "замещающий" метод из шаблонного метода.
6. Создайте подклассы от нового абстрактного базового класса и реализуйте в них "замещающие" методы.

# Результаты

Шаблонные методы вызывают операции следующих видов:

- Конкретные операции
- Конкретные операции из класса `AbstractClass`
- Примитивные операции
- Фабричные методы
- Операции-зацепки

# Реализация

- Использование контроля доступа в C++
- Сокращение числа примитивных операций
- Соглашение об именах

# Родственные паттерны

- Factory Method
- Strategy