

# C++ Advanced

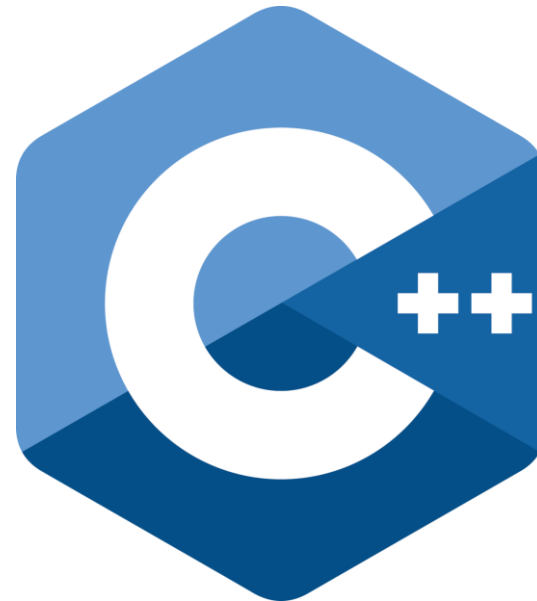
Шаблоны проектирования и C++.

# C++ Advanced

Автор курса



Кирилл Чернега



# C++ Advanced

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

## Шаблоны проектирования и C++.

# C++ Advanced

## Содержание урока

1. Типы шаблонов проектирования:  
Поведенческие, Порождающие, Структурные.
2. Реализация шаблонов в языке C++.
3. Рекомендации по использованию шаблонов в C++.

## Понятие шаблона проектирования

Шаблон проектирования или паттерн (англ. design pattern) — повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

Если привести аналогии, то алгоритм — это кулинарный рецепт с чёткими шагами, а паттерн — инженерный чертёж, на котором нарисовано решение, но не конкретные шаги его реализации.

Не путайте с архитектурными шаблонами (Ярусы, MVC, и т.д.) – они выше по уровню абстракции.

Необходимо знание UML диаграмм

SOLID принцип

[https://sourcemaking.com/design\\_patterns](https://sourcemaking.com/design_patterns)

# C++ Advanced

## Типы шаблонов проектирования (GoF)

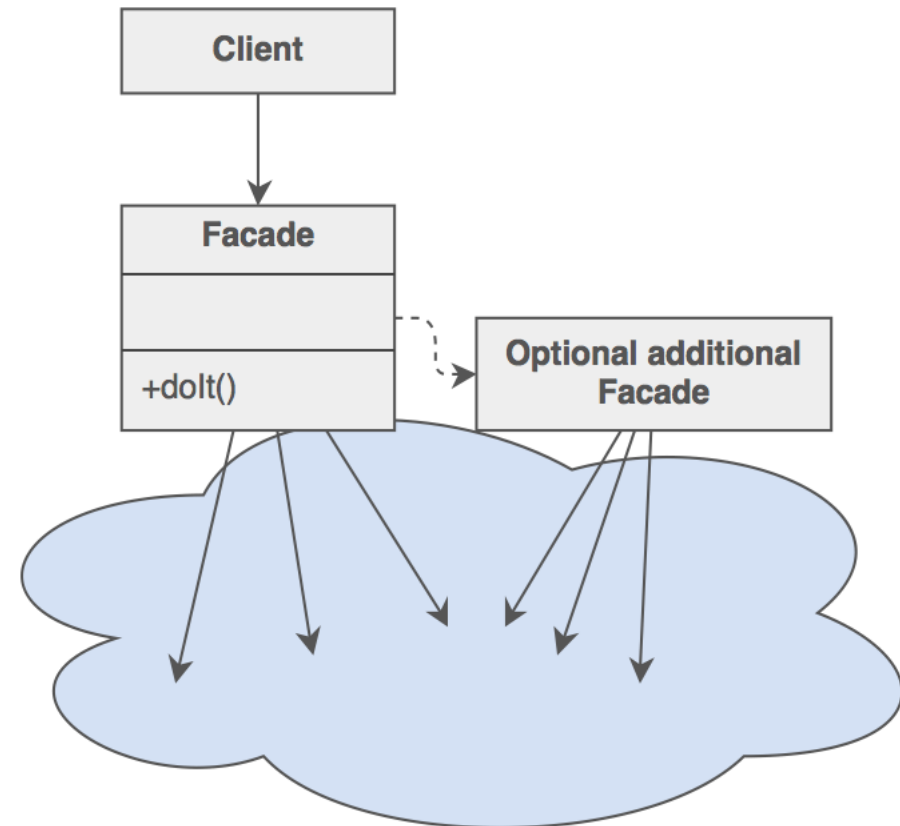
Creational	Structural	Behavioural
Factory Method Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Flyweight Facade Proxy	Interpreter Template Method Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

# C++ Advanced

## Facade (Фасад) - Structural

Предоставляет простой интерфейс к сложной системе классов, библиотеке или фреймворку.

Вашему коду приходится работать с большим количеством объектов некой сложной библиотеки или фреймворка. Вы должны самостоятельно инициализировать эти объекты, следить за правильным порядком зависимостей и так далее.

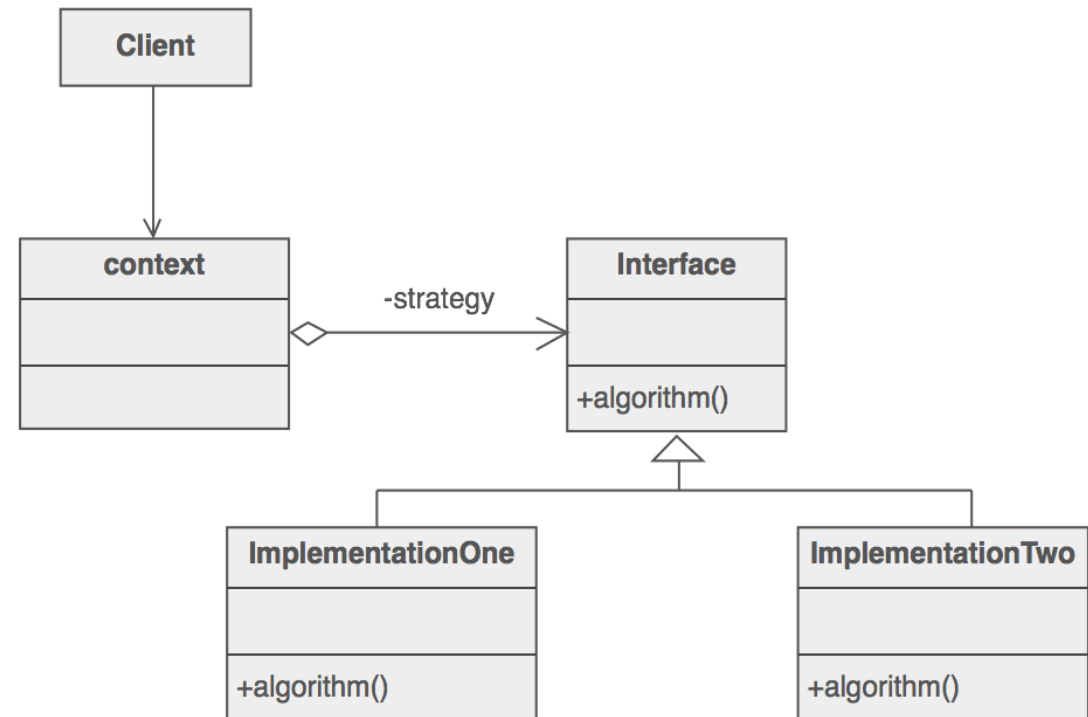




# C++ Advanced

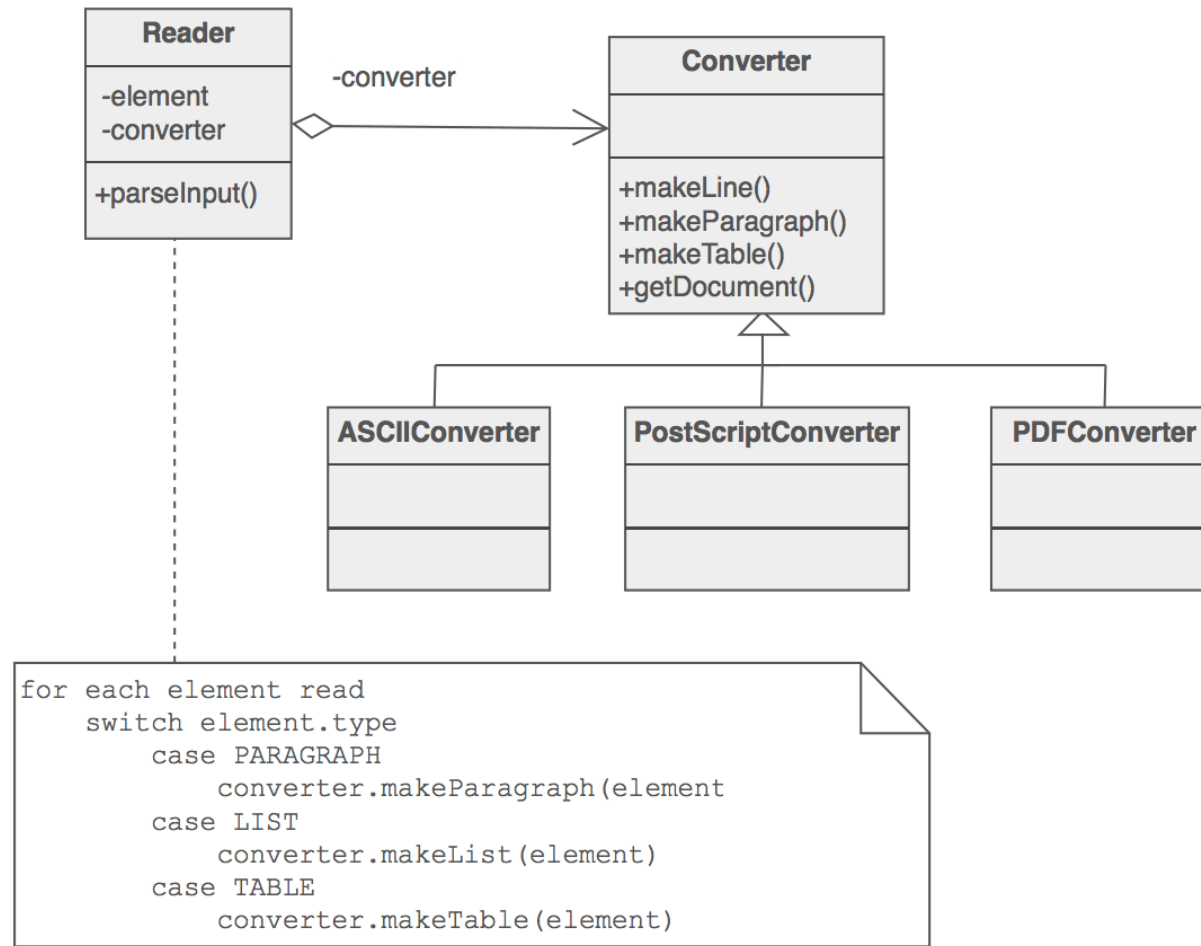
## Strategy ( Стратегия ) - Behavioural

Стратегия — это поведенческий паттерн проектирования, который определяет семейство схожих алгоритмов и помещает каждый из них в собственный класс, после чего алгоритмы можно взаимозаменять прямо во время исполнения программы.



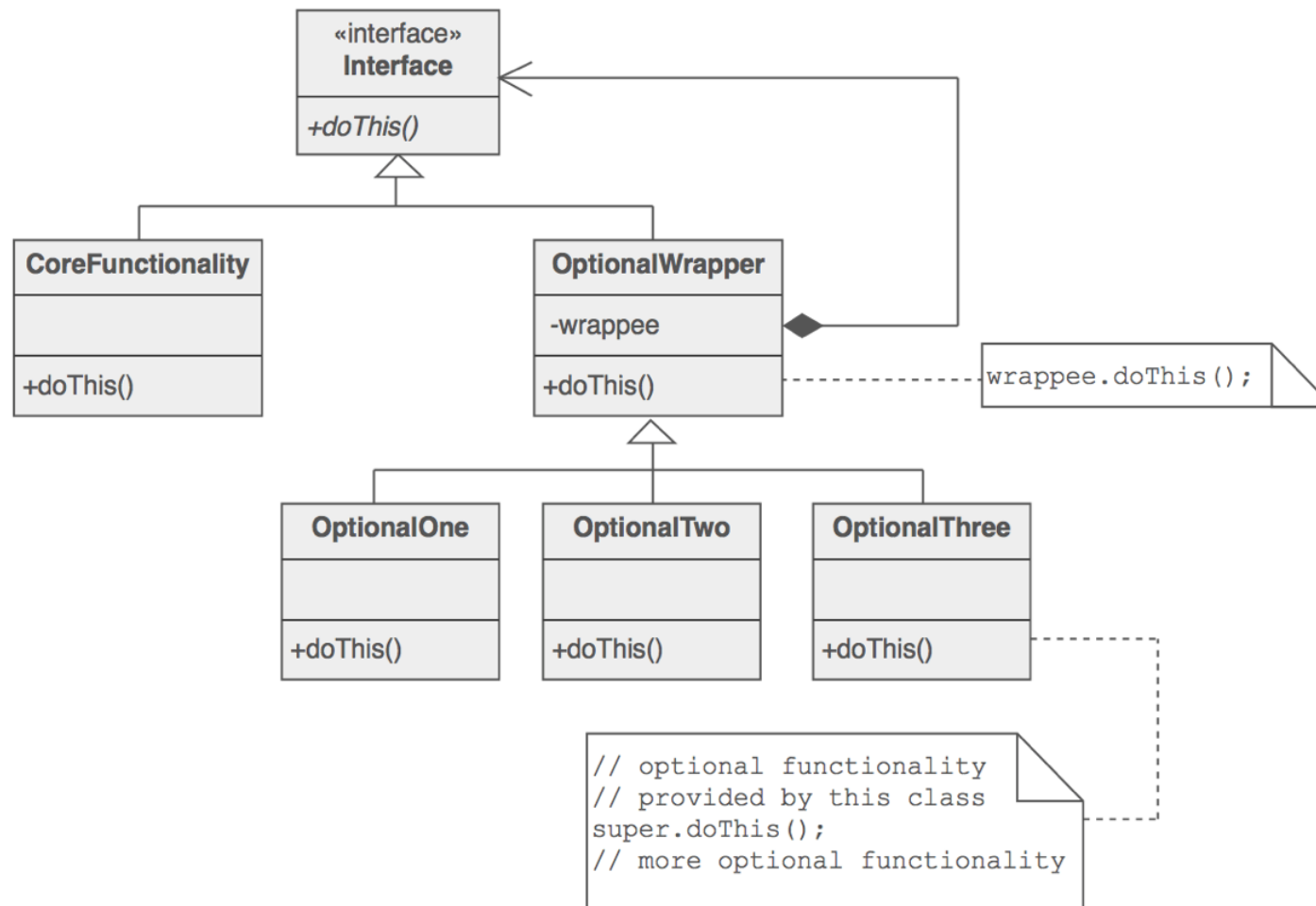
# C++ Advanced

## Builder (Строитель) - Creational



# C++ Advanced

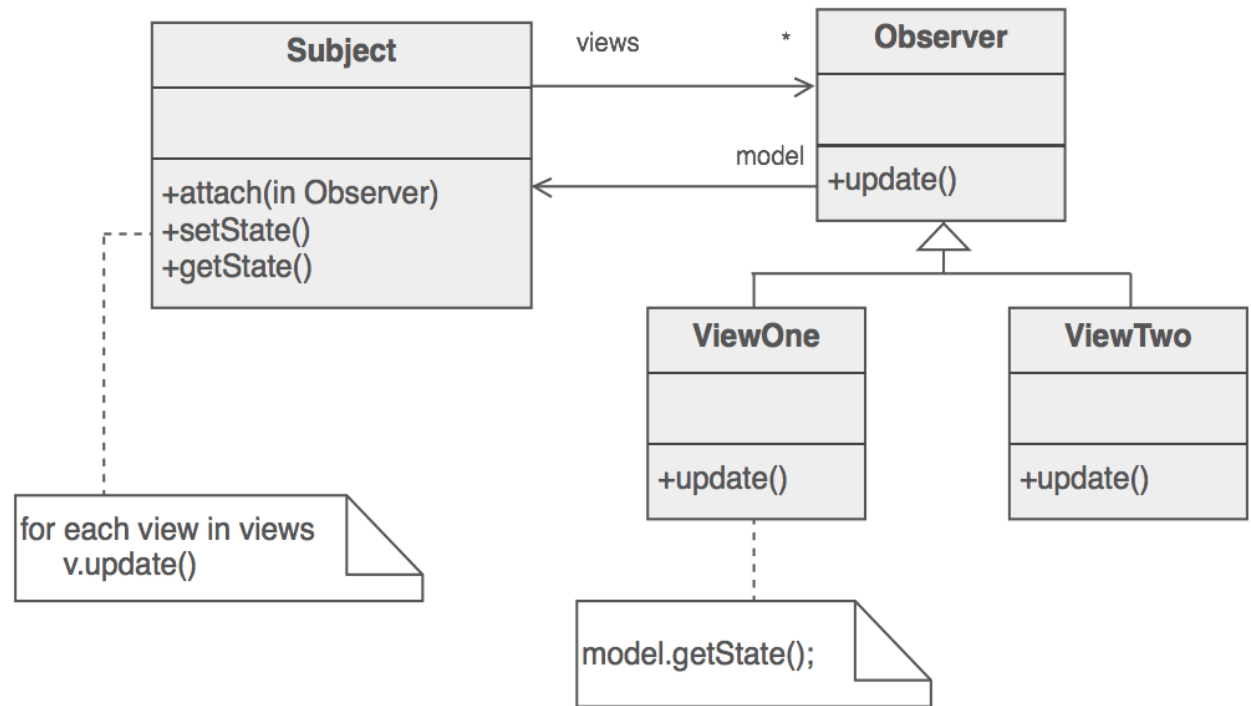
## Decorator (Декоратор) - Structural



# C++ Advanced

## Observer ( Наблюдатель ) - Behavioural

Наблюдатель — это поведенческий паттерн проектирования, который создаёт механизм подписки, позволяющий одним объектам следить и реагировать на события, происходящие в других объектах.

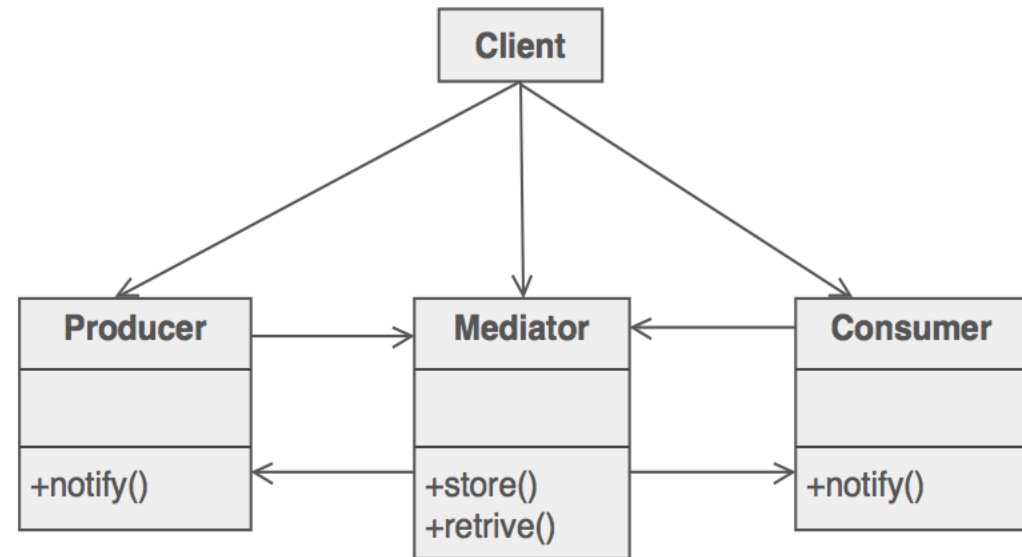


# C++ Advanced

## Mediator (Медиатор) - Behavioural

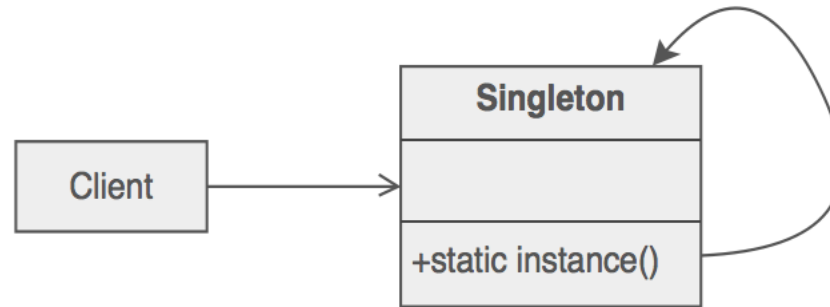
Посредник — это поведенческий паттерн проектирования, который позволяет уменьшить связанность множества классов между собой, благодаря перемещению этих связей в один класс-посредник.

Связь многие-ко-многим.



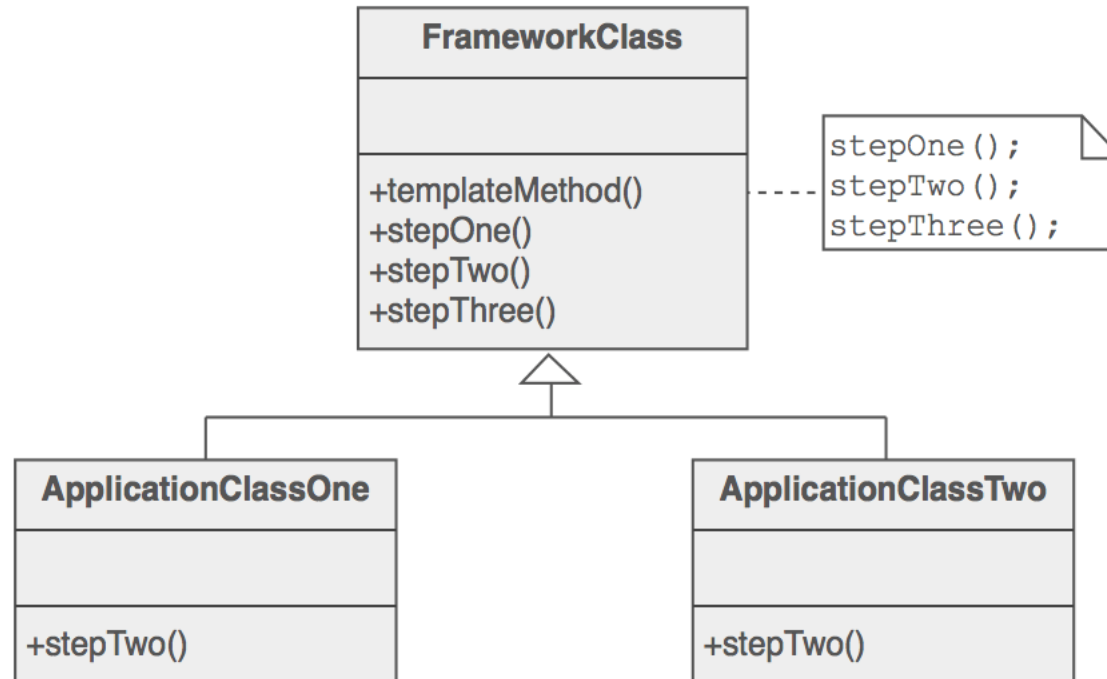
# C++ Advanced

## Singleton (Одиночка) - Creational



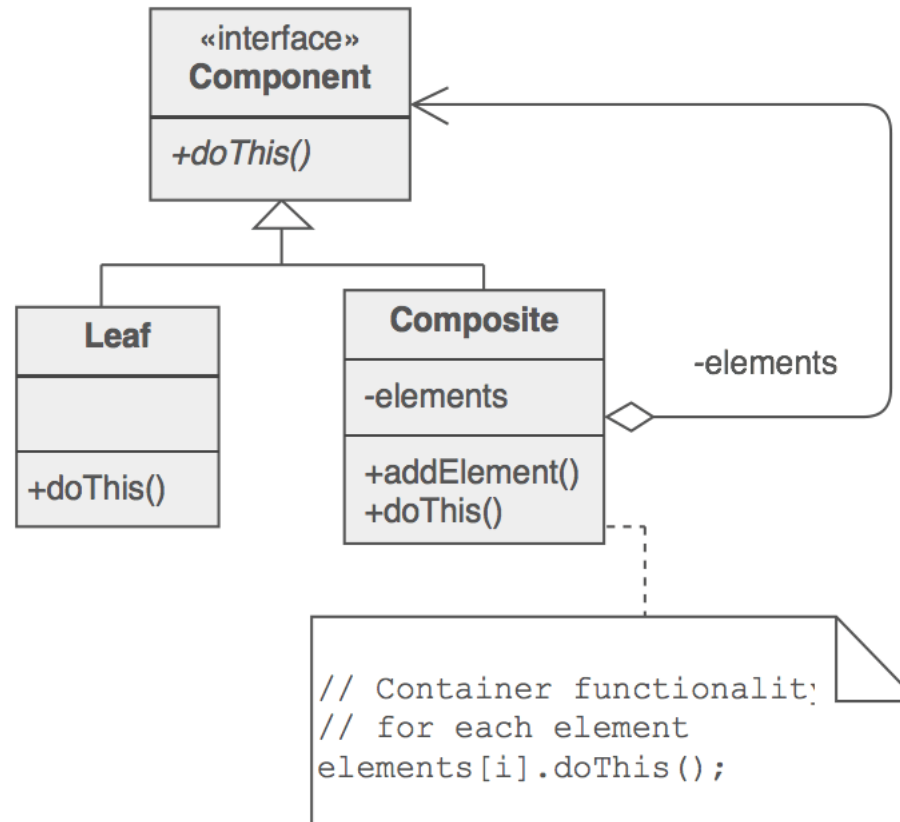
# C++ Advanced

## Template Method (Шаблонный метод) - Behavioural



# C++ Advanced

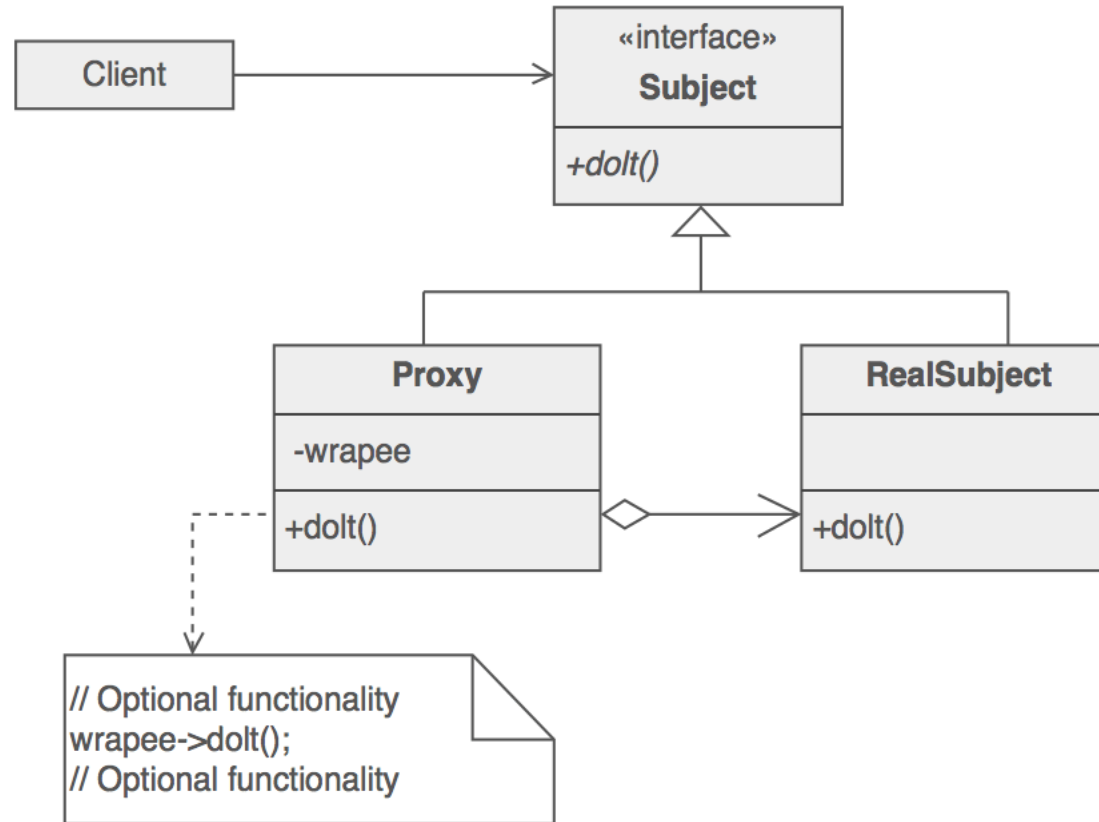
## Composite (Компоновщик) - Structural





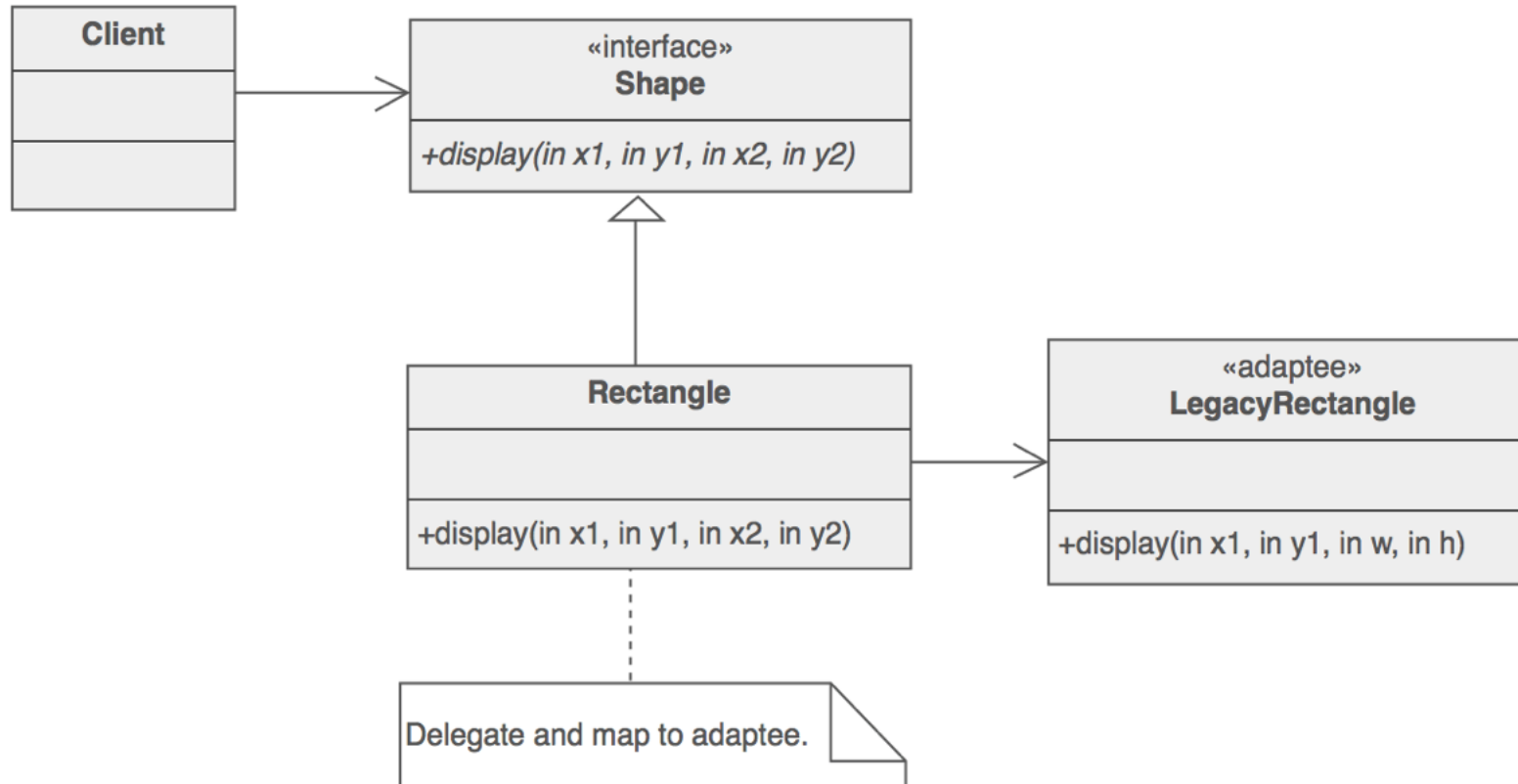
# C++ Advanced

## Proxy (Заместитель) - Structural



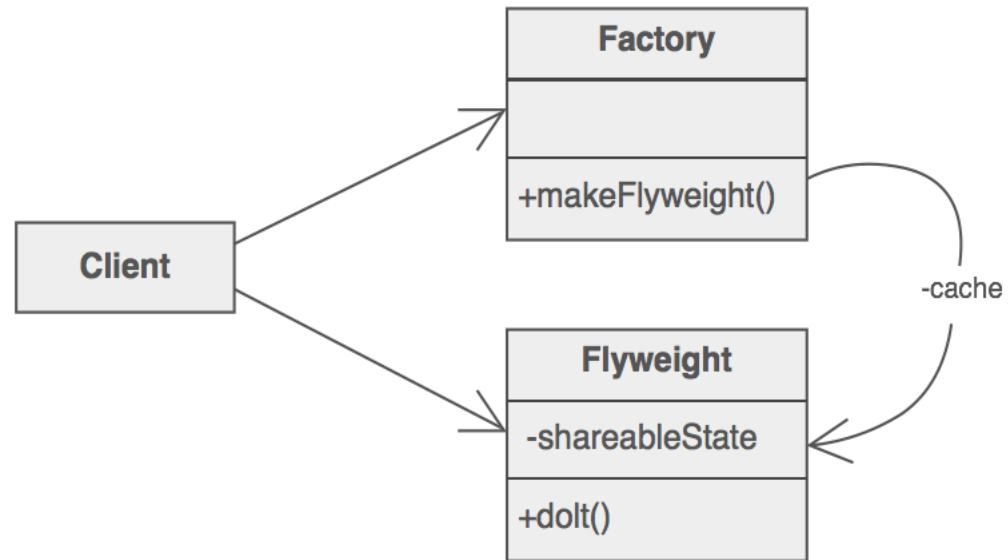
# C++ Advanced

## Adapter (Адаптер) - Structural



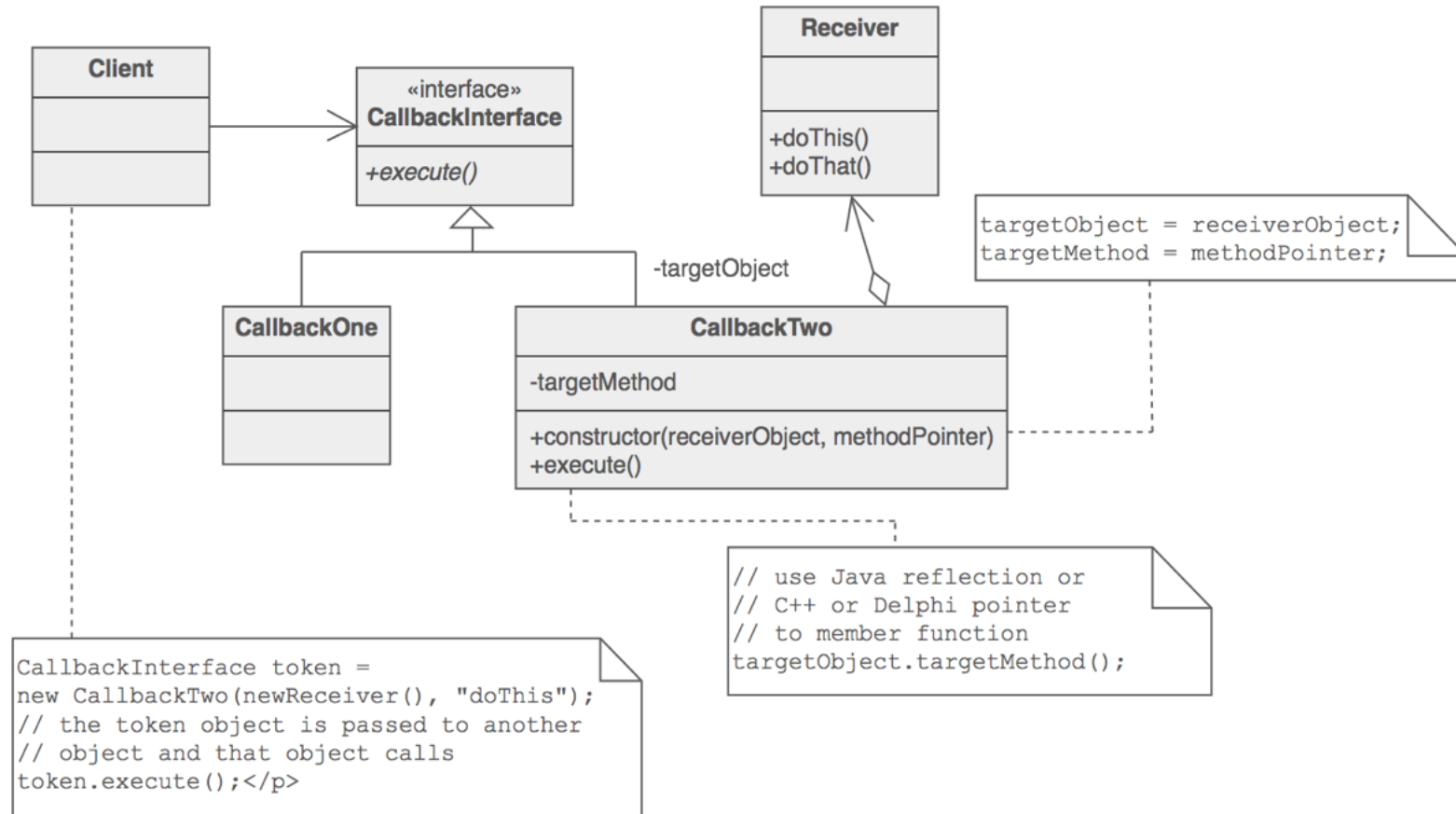
# C++ Advanced

## Flyweight ( Легковес ) - Structural



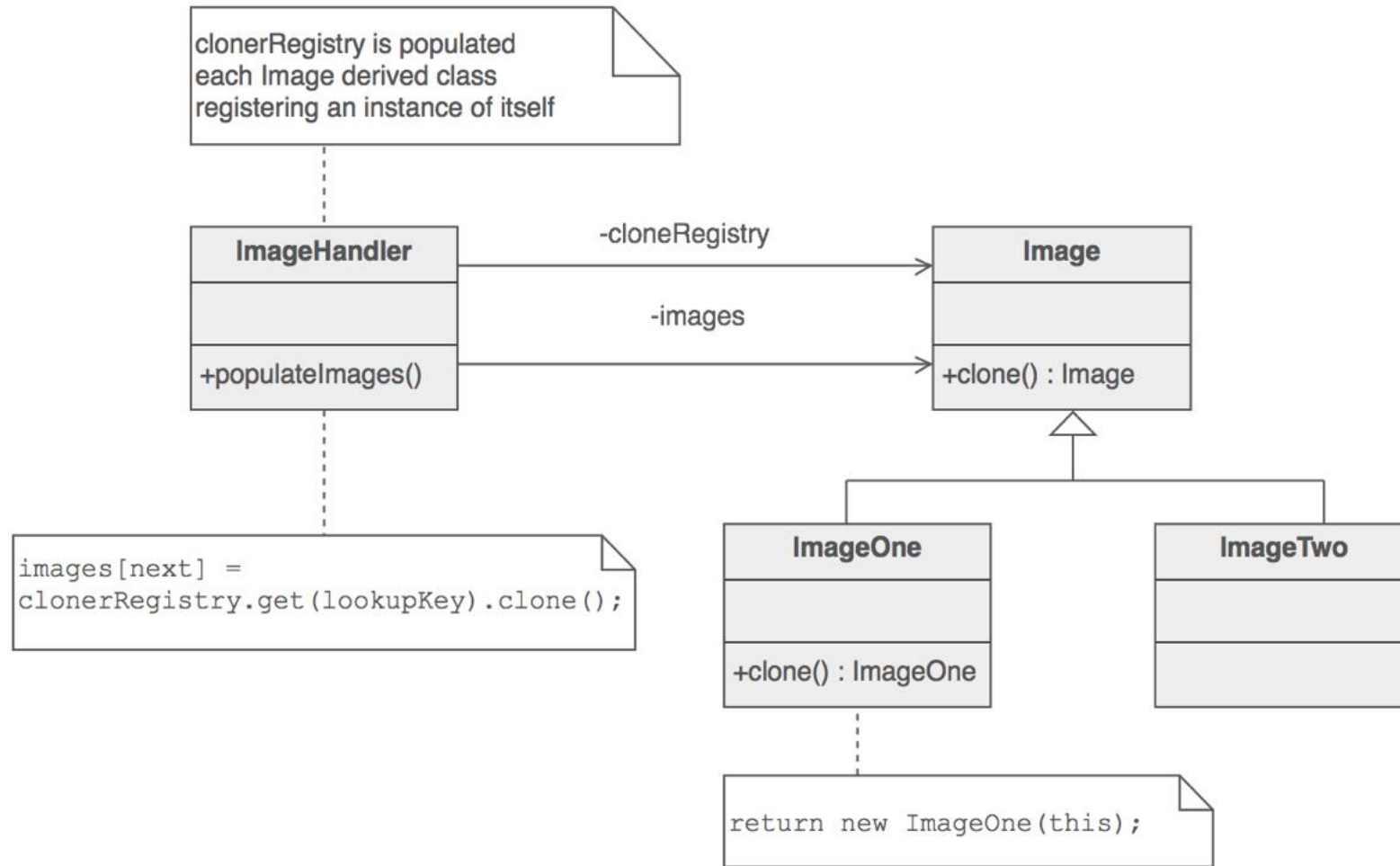
# C++ Advanced

## Command ( Команда ) - Behavioural



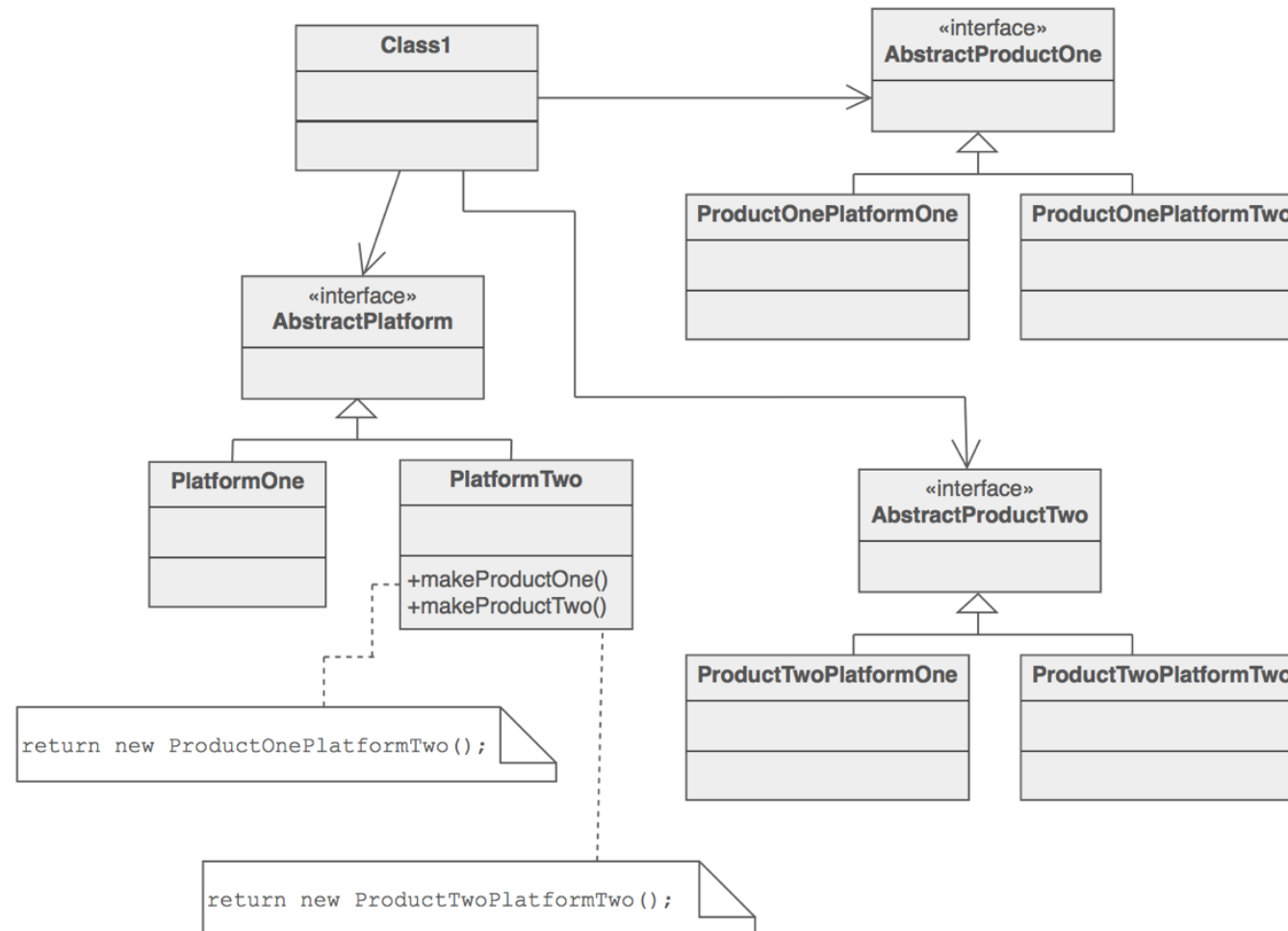
# C++ Advanced

## Prototype ( Прототип ) - Creational



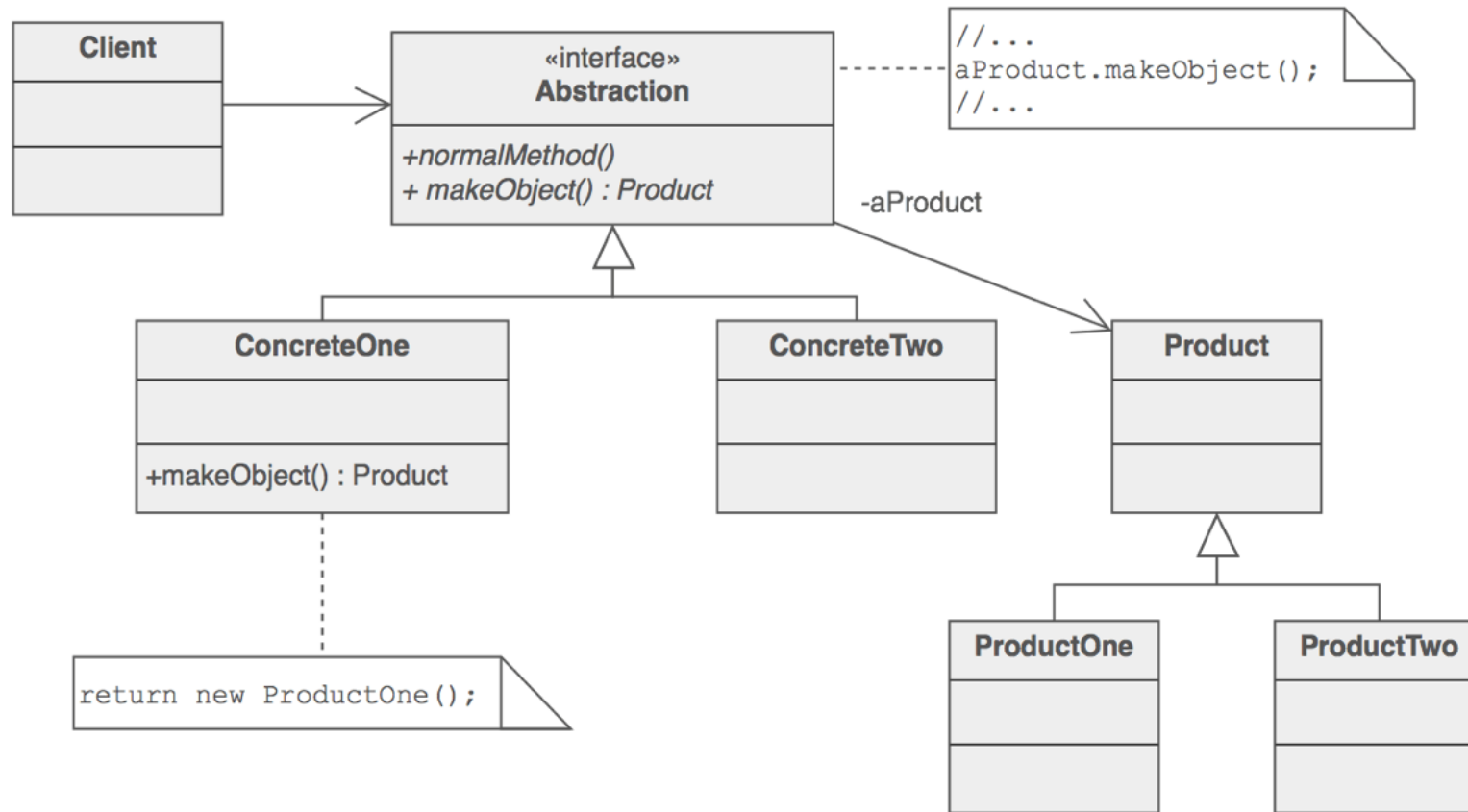
# C++ Advanced

## Abstract Factory ( Абстрактная фабрика ) - Creational



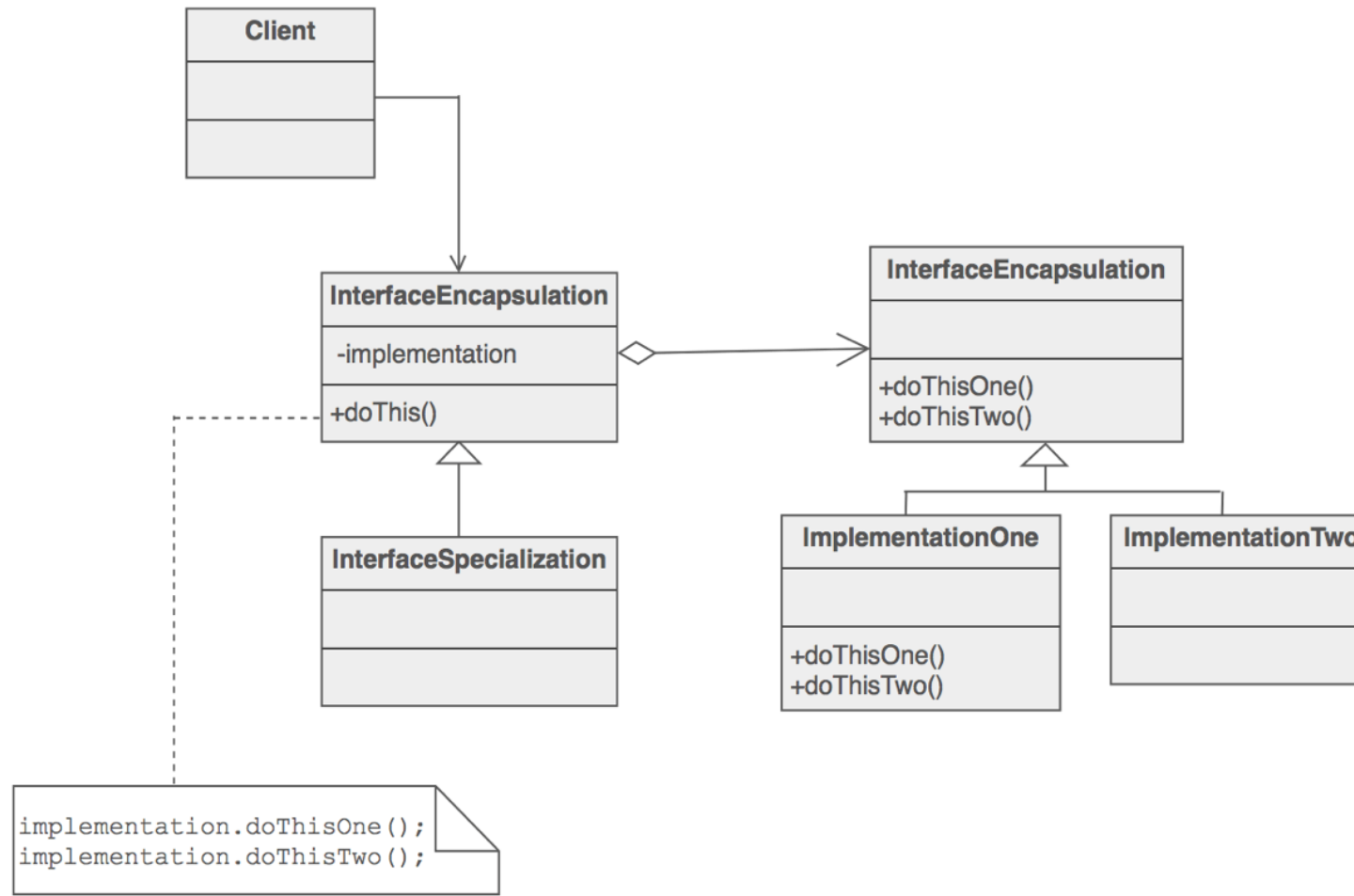
# C++ Advanced

## Factory Method ( Фабричный метод) - Creational



# C++ Advanced

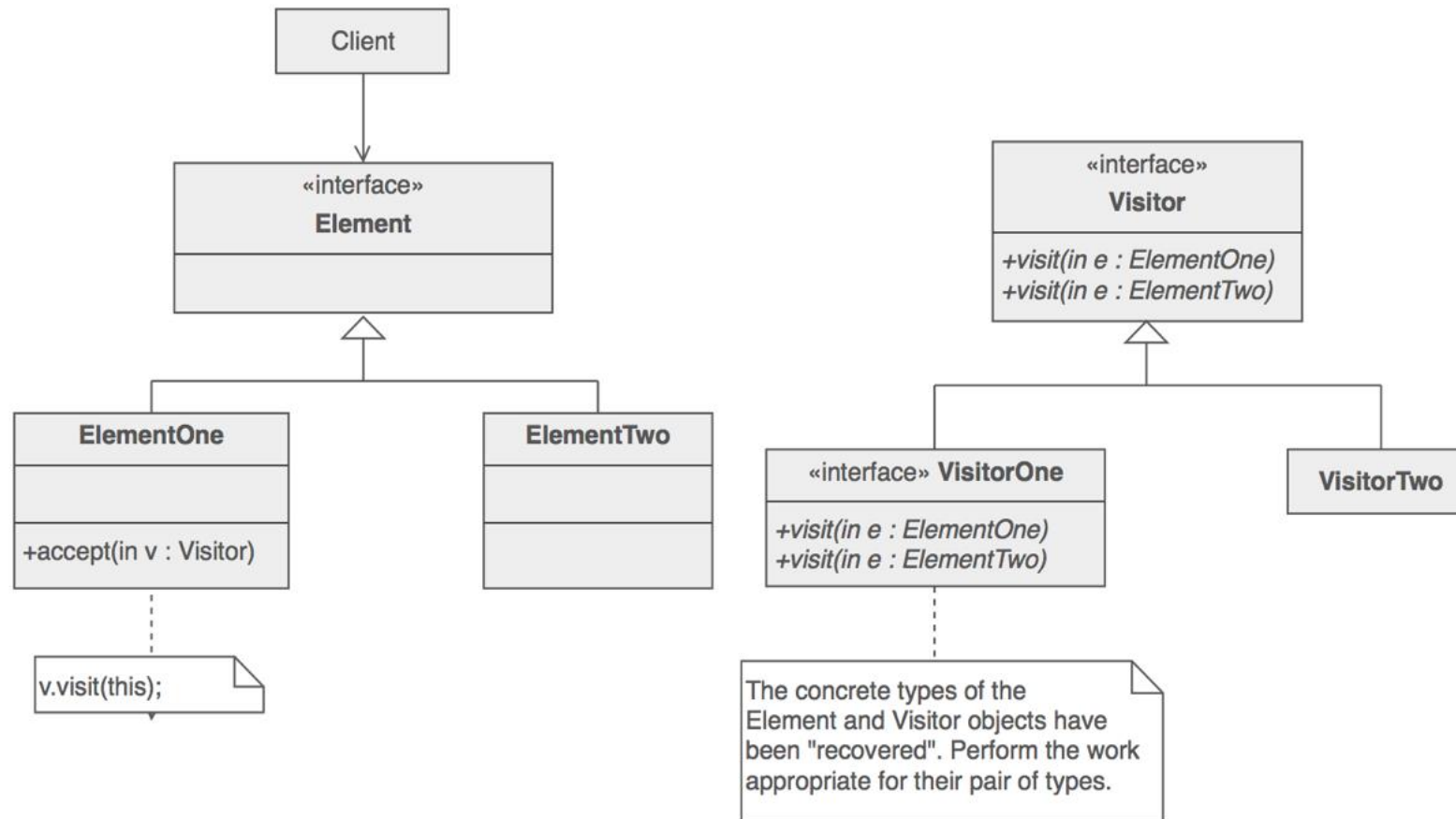
## Bridge( Мост ) - Structural





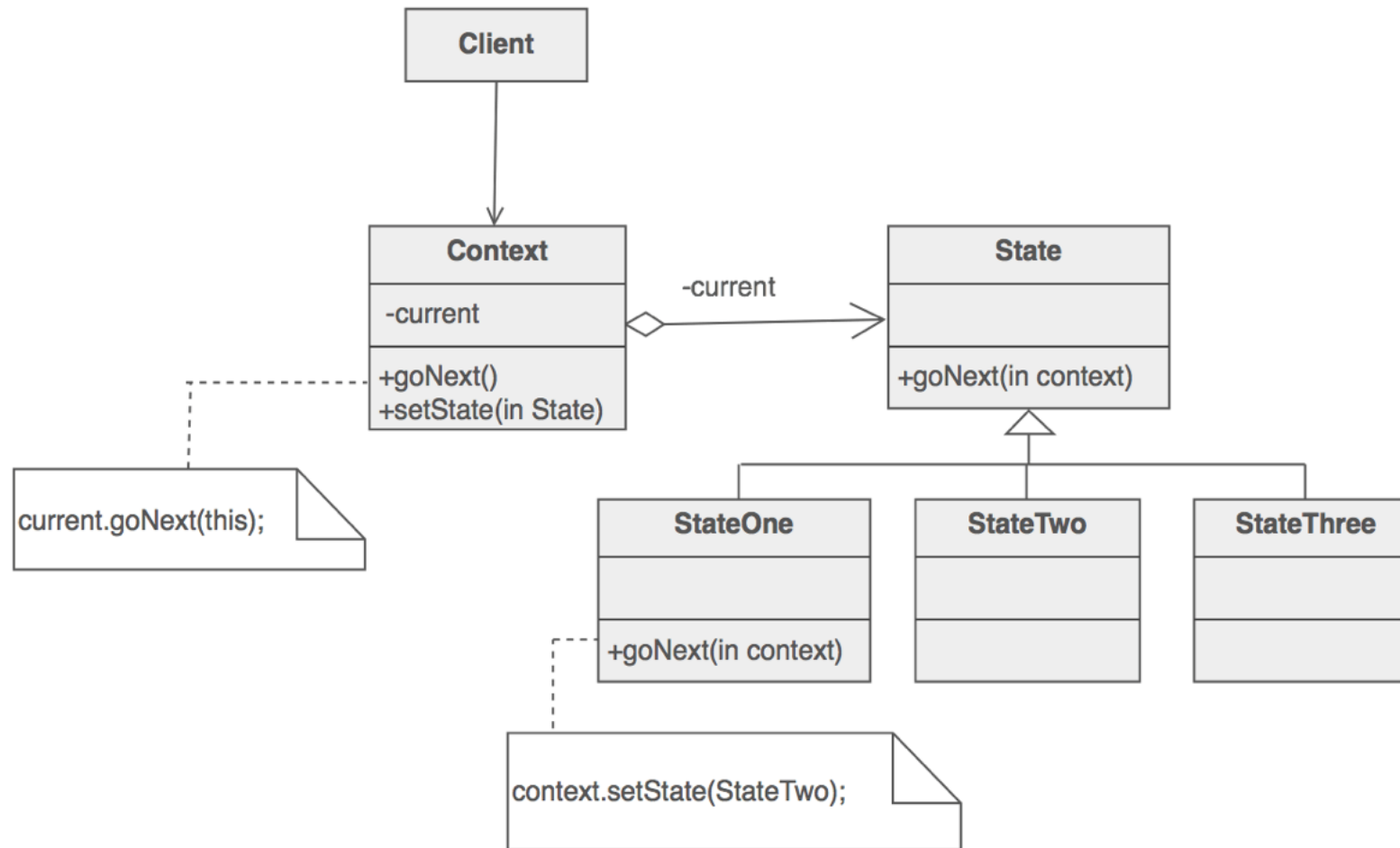
# C++ Advanced

## Visitor( Посетитель ) - Behavioural



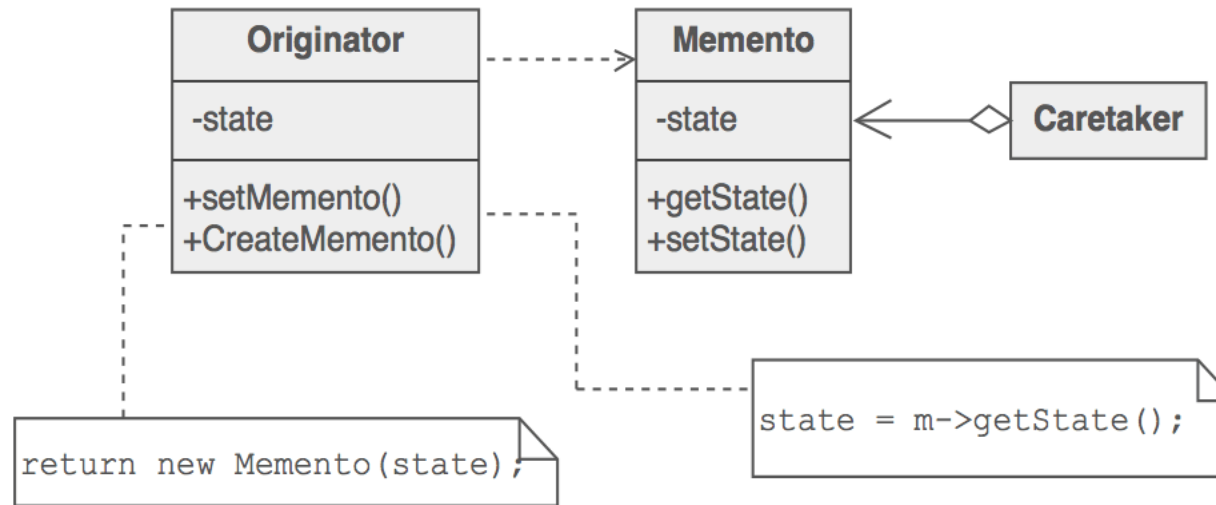
# C++ Advanced

## State ( Состояние ) - Behavioural



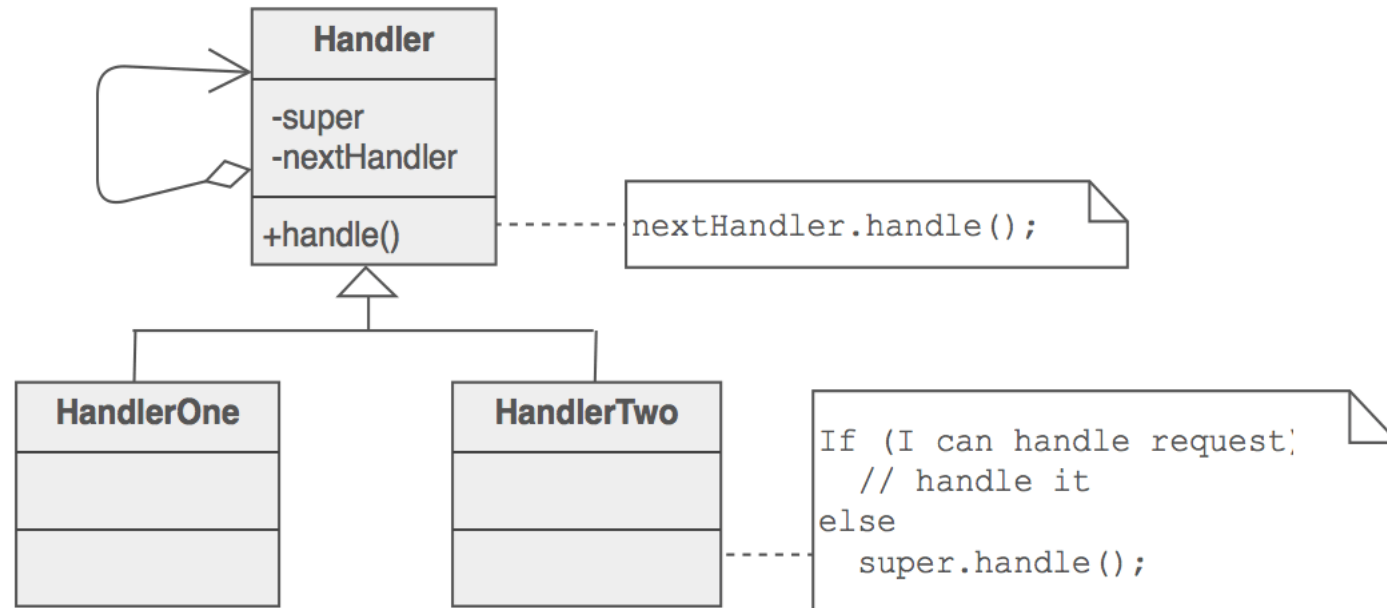
# C++ Advanced

## Memento ( Снимок ) - Behavioural



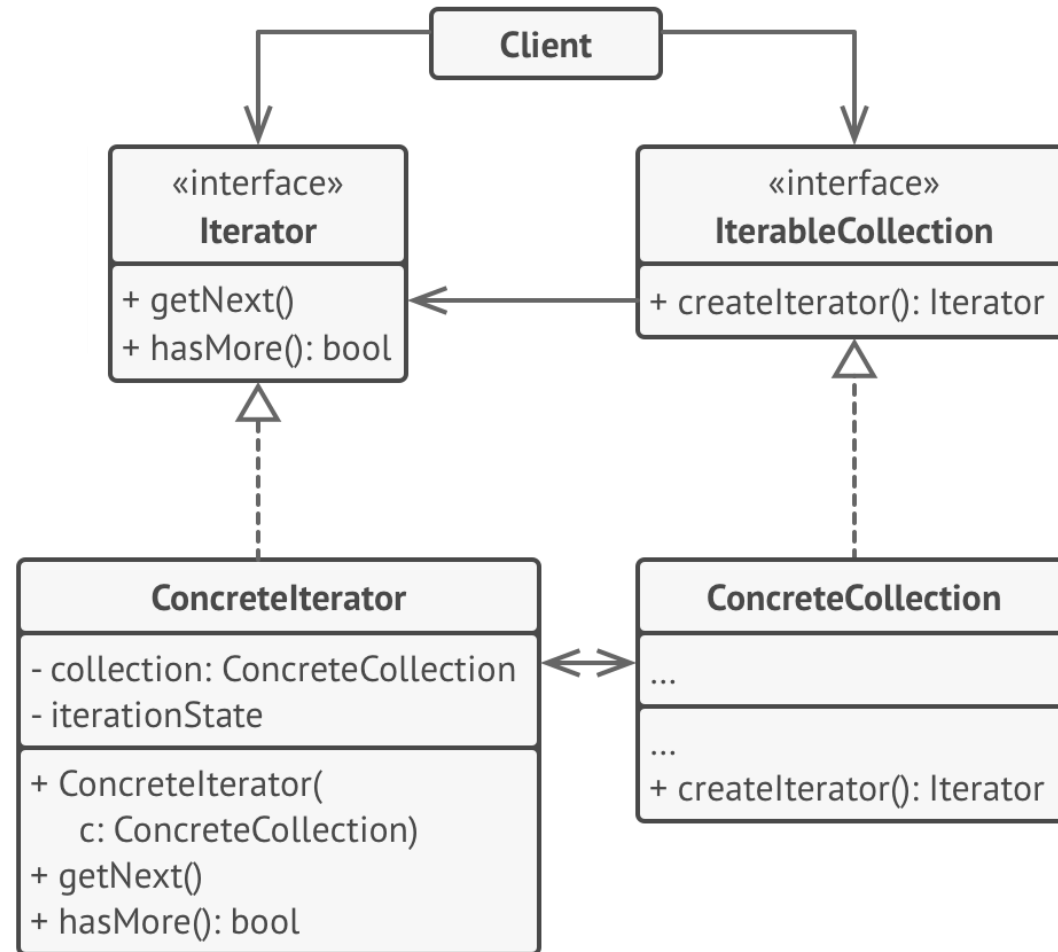
# C++ Advanced

## Chain of Responsibility ( Цепочка обязанностей ) - Behavioural



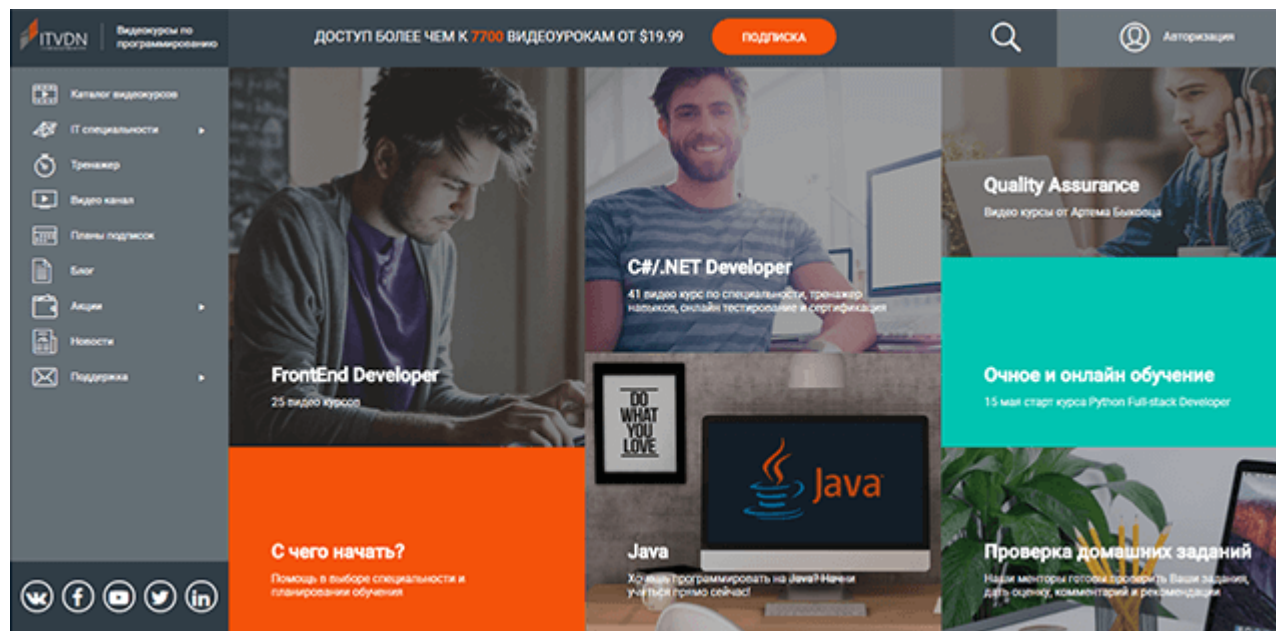
# C++ Advanced

## Iterator ( Итератор) - Behavioural



# Смотрите наши уроки в видео формате

ITVDN.com



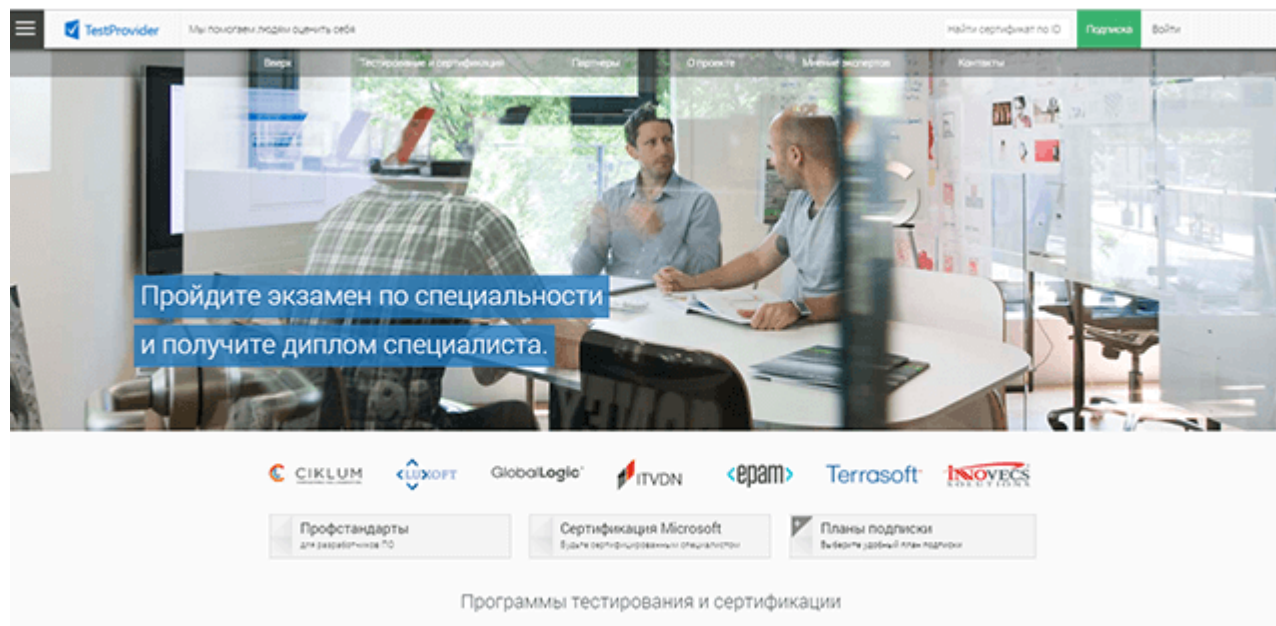
Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



# Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



## Q&A



# Информационный видеосервис для разработчиков программного обеспечения

