

Пермский филиал федерального государственного автономного  
образовательного учреждения высшего образования  
«Национальный исследовательский университет  
«Высшая школа экономики»

*Факультет экономики, менеджмента и бизнес-информатики*

Рязанов Иван Дмитриевич

**ОРГАНИЗАЦИЯ ПАТТЕРНОВ ПРОЕКТИРОВАНИЯ.  
ПОРОЖДАЮЩИЕ ПАТТЕРН «СТРОИТЕЛЬ»**

*Лабораторная работа*

студента образовательной программы «Программная инженерия»  
по направлению подготовки 09.03.04 Программная инженерия

Руководитель  
к.т.н., доцент кафедры Информационных технологий в бизнесе НИУ ВШЭ-Пермь

---

А.В. Кычкин

Пермь, 2020 год

# Оглавление

Глава 1. Паттерн «Строитель» .....	3
Глава 2. Проектирование и реализация .....	7
2.1 Проектирование . . . . .	7
2.2 Реализация . . . . .	8

# Глава 1. Паттерн «Строитель»

**Название и классификация паттерна.** Строитель - паттерн, порождающий объекты.

**Назначение.** Отделяет конструирование сложного объекта от его представления, так что в результате одного и того же процесса конструирования могут получаться разные представления. В паттерне строитель абстрагированы все эти отношения. В нем любой класс конвертора называется строителем, а загрузчик – распорядителем.

**Применимость.** Использование паттерна Строитель целесообразно если:

- процесс создания нового объекта не должен зависеть от того, из каких частей этот объект состоит и как эти части связаны между собой;
- необходимо обеспечить получение различных вариаций объекта в процессе его создания.

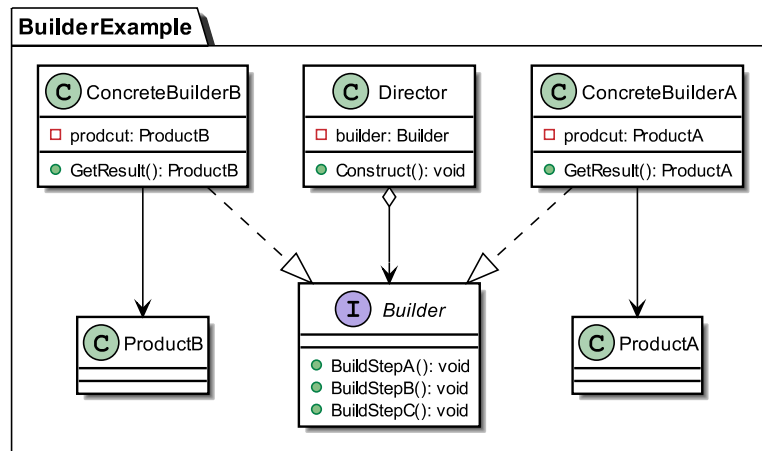


Рис. 1.1. Диаграмма классов паттерна «Строитель»

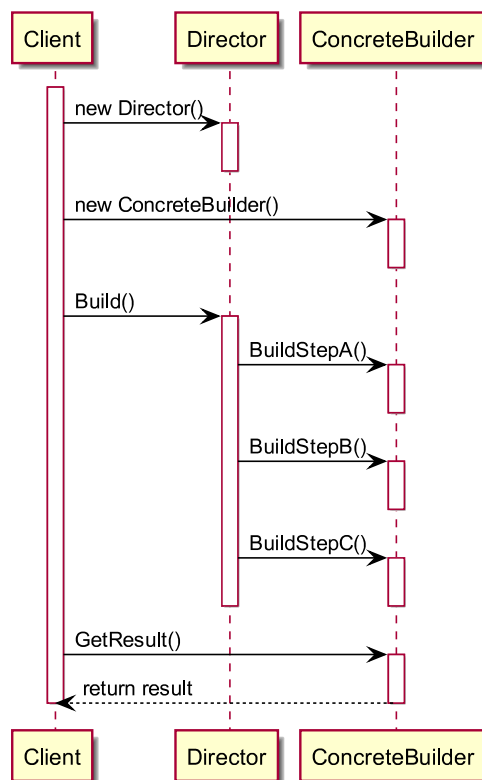


Рис. 1.2. Диаграмма последовательности паттерна «Строитель»

## **Участники**

- Builder - абстрактный строитель: объявляет интерфейс для операций, создающих части продукта;
- ConcreteBuilder (ConcreteBuilderA, ConcreteBuilderB) - конкретный строитель: реализует операции, создающие конкретные части продукта;
- Product (ProductA, ProductB) - продукт: определяет объект-продукт, создаваемый соответствующим строителем;
- Director - распорядитель (директор): пошагово конструирует продукт, используя методы строителя;

## **Отношения**

- клиент создает объект-распорядитель Director и конфигурирует его нужным объектом-строителем Builder;
- распорядитель уведомляет строителя о том, что нужно построить очередную часть продукта;
- строитель обрабатывает запросы распорядителя и добавляет новые части к продукту;
- клиент забирает продукт у строителя.

## **Плюсы и минусы.**

Плюсы:

- возможность контролировать процесс создания сложного продукта;
- возможность получения разных представлений некоторых данных.

Минусы:

- ConcreteBuilder и создаваемый им продукт жестко связаны между собой, поэтому при внесении изменений в класс продукта скорее всего придется соответствующим образом изменять и класс ConcreteBuilder.

## **Области применения**

1. Создание XML-документов или HTML-документов с помощью классов-строителей с методами для создания элементов документа (тегов).
2. Класс `StringBuilder` в C#.

## Глава 2. Проектирование и реализация

### 2.1. Проектирование

Для реализации был выбран 1 вариант:

«Создание отчета, содержащего различные графические компоненты (текст, диаграммы, графики), которые могут идти в произвольном порядке.»

Перед началом работы построим диаграмму классов.

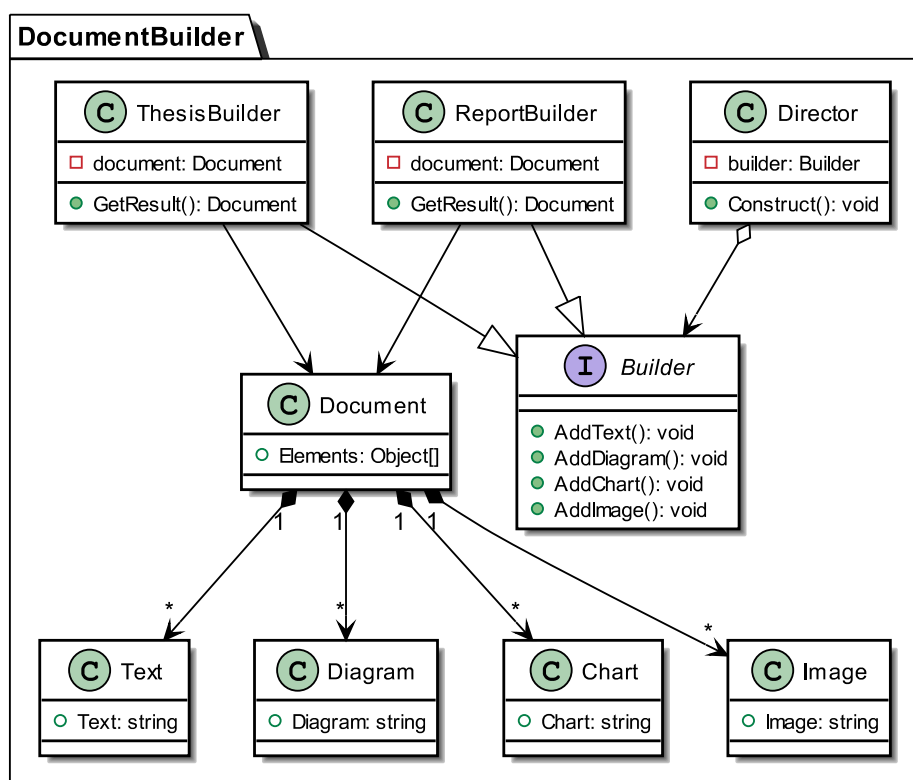


Рис. 2.1. Диаграмма классов

#### Участники.

- Builder - абстрактный строитель: объявляет интерфейс для операций, создающих части продукта;
- ThesisBuilder, ReportBuilder) - конкретные строители: реализуют операции, создающие разные типы документов (Курсовая или отчет);
- Document - продукт: документ, создаваемый соответствующим строителем;

- Director - распорядитель (директор): пошагово конструирует продукт, используя методы строителя;
- Text - текст;
- Diagram - диаграмма;
- Chart - график;
- Image - изображение;

## 2.2. Реализация

Реализация паттерна «Строитель» находится в git-репозитории по ссылке: [github.com](https://github.com)