**Аналогия**

Допустим, вы пришли в забегаловку, заказали бургер дня, и вам выдали его без вопросов. Это пример «Простой фабрики». Но иногда логика создания состоит из большего количества шагов. К примеру, при заказе бургера дня есть несколько вариантов хлеба, начинки, соусов, дополнительных ингредиентов. В таких ситуациях помогает шаблон «Строитель».

**Вкратце**

Шаблон позволяет создавать разные свойства объекта, избегая загрязнения конструктора (constructor pollution). Это полезно, когда у объекта может быть несколько свойств. Или когда создание объекта состоит из большого количества этапов.

**Википедия**

Шаблон «Строитель» предназначен для поиска решения проблемы антипаттерна Telescoping constructor.

Поясню, что такое антипаттерн Telescoping constructor. Каждый из нас когда-либо сталкивался с подобным конструктором:

**public** **function** **\_\_construct**($size, $cheese = true, $pepperoni = true, $tomato = false, $lettuce = true)

{

}

количество параметров может быстро разрастись, и станет трудно разобраться в их структуре.

**Когда использовать?**

Когда у объекта, может быть, несколько свойств и когда нужно избежать Telescoping constructor. Ключевое отличие от шаблона «Простая фабрика»: он используется в одноэтапном создании, а «Строитель» — в многоэтапном.

**Пример**

Имеются Книги (детективы и классика). Мы хотим создавать каталоги книг, состоящие из определенных книг. Каталогов будет много, нужны будут в разное время. Потому создается класс «Каталог», состоящий из массивов для каждого типа книг и функций вывода на экран о содержимом. А также «Билдер Каталог». В нем описывается «интерфейс» в виде виртуальных функций из «добавить книгу детектив», «добавить книгу классику» и это добавляется в определённые вектора. Наследники такого «Билдер Каталог» являются «Билдер Каталог 2008 года» и любой следующий «Билдер Каталог 2010 года». В нем имеется реализация функций по добавлению разных книг в нужные массивы. Теперь же в main можно вызвать «Билдер Каталог 2008 года» и главную его функцию «Создать каталог», в которой будут прописаны все функции, что надо заюзать для создания такого каталога. На выходе имеет объект «Каталог» состоящий из массивов разных книг. Для идентификации можно добавить поле «Год выпуска». Подобный паттерн позволяет строго следить за конфигурацией каждого из Каталогов!

http://cpp-reference.ru/patterns/creational-patterns/builder/

https://habr.com/ru/company/vk/blog/325492/