

Министерство образования и науки Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

ОТЧЁТ  
«Лабораторная работа №1»  
по дисциплине  
«Объектно-ориентированный анализ и проектирование»

Выполнил:  
студент группы 932220  
Муравьев Пётр Евгеньевич

Томск – 2024

**Диаграмма без паттерна:** [клик](#)

**Диаграмма с паттерном:** [клик](#)

## Документация

### Реализация без использования паттерна

От абстрактного класса VolkswagenPolo наследуются 5 классов VP\_Origin, VP\_Respect, VP\_Status, VP\_FootballEdition и VP\_Exclusive, которые являются различными конфигурациями автомобиля Volkswagen Polo.

В данной реализации пользователь может выбрать только одну из 5 этих конфигураций. При этом, для того, чтобы создать Volkswagen Polo какой-либо другой конфигурации, необходимо обращаться к коду и прописывать 6-й класс и прописывать всё его множество полей.

Долго и неудобно, попробуем реализовать с паттерном билдер.

### Реализация с использованием паттерна

Класс VolkswagenPolo содержащий 5 полей - элементов конфигурации автомобиля Volkswagen Polo и метод GetInfo() для получения информации об объекте класса.

Интерфейс IVolkswagenPoloBuilder описывает методы, которые должен реализовать строитель VolkswagenPoloBuilder.

Класс VolkswagenPoloBuilder реализует интерфейс IVolkswagenPoloBuilder, храня объект volkswagenPolo класса VolkswagenPolo и возвращая его в методе GetResult().

Класс Director хранит ссылочную переменную builder типа IVolkswagenPoloBuilder, для того чтобы он мог работать с объектами всех классов которые реализуют интерфейс IVolkswagenPoloBuilder. А его метод Construct() строит объект по параметрам которые выбрал

пользователь.

Таким образом, в данной реализации есть возможность создавать уникальную конфигурацию автомобиля Volkswagen Polo.

**Вывод:**

Паттерн билдер в моём случае очень хорошо подошел и довольно изящно решил проблему создания уникальных конфигураций Volkswagen Polo.