



# Design Patterns

by Versus 2020

# Fluent Interface

## Текущий интерфейс

Структурные шаблон





# Суть шаблона

Повысить читаемость кода за счет создания предметно-ориентированного языка (DSL).



# Проблема

В случае частого использования API какой либо службы или построения сложно-понятный языковых конструкций:

- кодовая база усложняется
- дублирование кода



## Решение

Интерфейс взаимодействия с этими объектами в удобно-читаемом формате.

# Структура

© Sql

f from

f where

f fields

m \_\_toString()

m where(condition)

m from(table, alias)

m select(fields)

# Пример кода





# Применимость

Привести к объектной форме (DSL) работу с нестандартными/сложными интерфейсами и сервисами.





# Шаги реализации

1. Реализовать класс, отвечающий за сборку нужной объектной или языковой конструкции.
2. Реализовать методы, отвечающие за построение конструкции. Методы должны возвращать текущий объект.
3. Реализовать метод, преобразующий выставленные свойства в необходимую форму.



# Преимущества

С помощью небольшой надстройки мы получаем удобочитаемый интерфейс разработки.



# Недостатки

Может противоречить к «безотказному» подходу к защите от ошибок, может пагубно повлиять на отладку.

Следует руководствоваться здравым смыслом и не делать цепочки слишком длинными.

ИТОГ

