

A decorative geometric pattern in the bottom-left corner, featuring interlocking shapes in shades of orange, purple, green, and brown, resembling a traditional Islamic or Persian tile design. The pattern is partially obscured by a diagonal blue and grey band that runs from the top-left towards the bottom-right.

Рефакторинг (ОСНОВЫ)

Общие понятия и определения

"Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure."

– Martin Fowler

Рефакторинг представляет собой процесс такого изменения программной системы, при котором не меняется внешнее поведение кода, но улучшается его внутренняя структура.

Это способ
систематического
приведения кода в
порядок, при котором
шансы появления новых
ошибок минимальны.

*В сущности,
при проведении
рефакторинга кода
вы улучшаете его
дизайн уже после того,
как он написан.*



Цель рефакторинга

сделать код программы
более легким для
понимания,
без этого
рефакторинг нельзя
считать успешным.




Рефакторинг VS Оптимизация

И рефакторинг и оптимизация обычно не изменяют поведение программы

Оптимизация часто затрудняет понимание кода.
Рефакторинг – упрощает.

Оптимизация позволяет «ускорить» работу программы.
Рефакторинг может «замедлять» работу программы.



Рефакторинг VS Расширение функциональности программного обеспечения

при рефакторинге
не изменяется поведение программы.

его зачастую применяют **до и/или после**
расширения функциональности систем


Преимущества применения рефакторинга

Улучшает композицию программного кода

Облегчает читабельность программного кода

Помогает обнаружить ошибки

Помогает писать код быстрее



**Рефакторинг
при разработке кода
нужно применять
ПОСТОЯННО**

Причины применения

необходимость добавить новую функцию, которая недостаточно укладывается в принятое архитектурное решение;

необходимость исправить ошибку, причины возникновения которой сразу не ясны;

преодоление трудностей в командной разработке, которые обусловлены сложной логикой программы.

Признаки кода, нуждающегося в рефакторинге

(*Code smell*, also known as **bad smell**)

Code Smells - не сама проблема, а её индикатор

Существуют видимые проблемы в коде, требующие рефакторинга:

дублирование кода;

длинный метод;

большой класс;

длинный список параметров;

«завистливые» функции — метод, который чрезмерно обращается к данным другого объекта;

«стрельба дробью»;

несгруппированные данные;

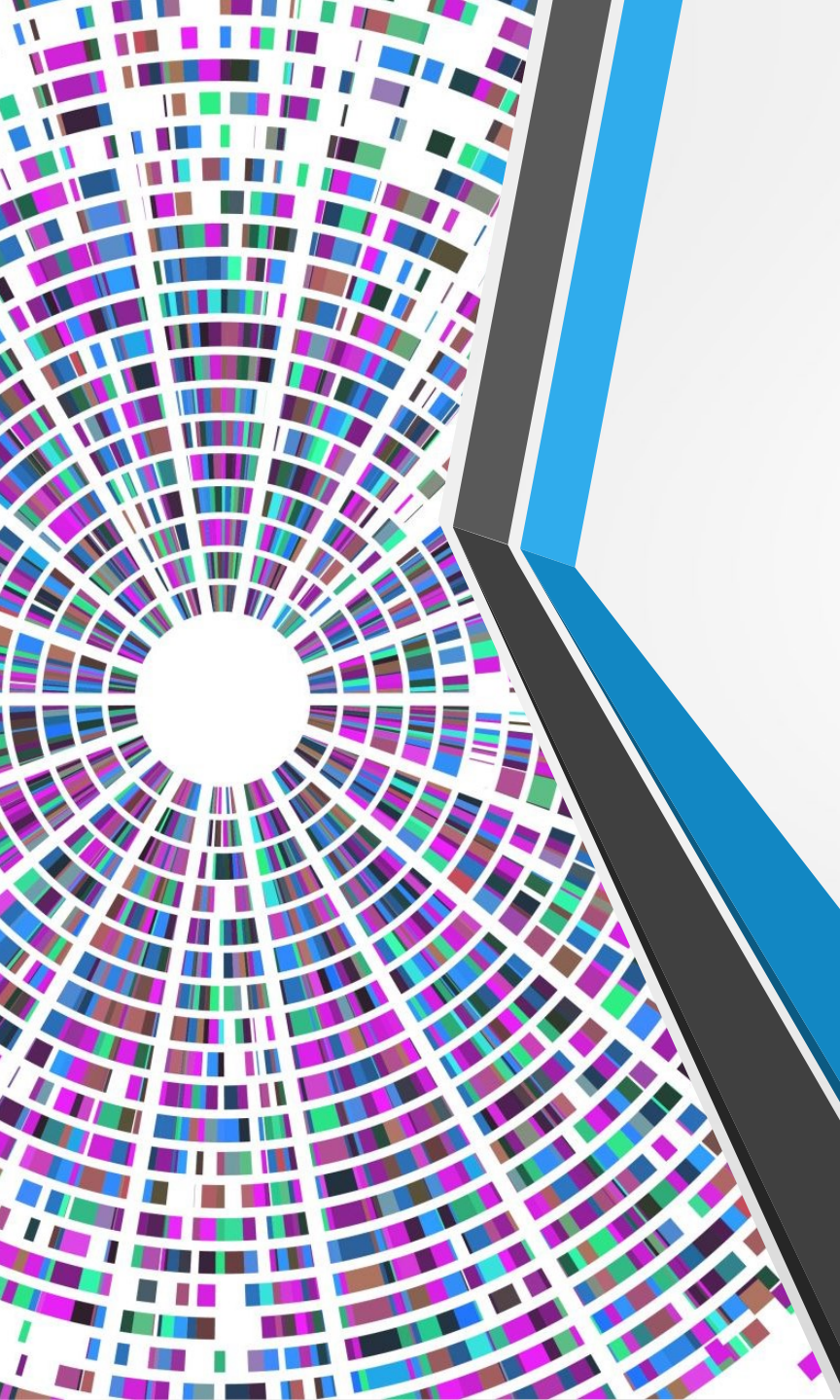
...

**Проблемы,
к которым
приводит
дублирование
кода:**

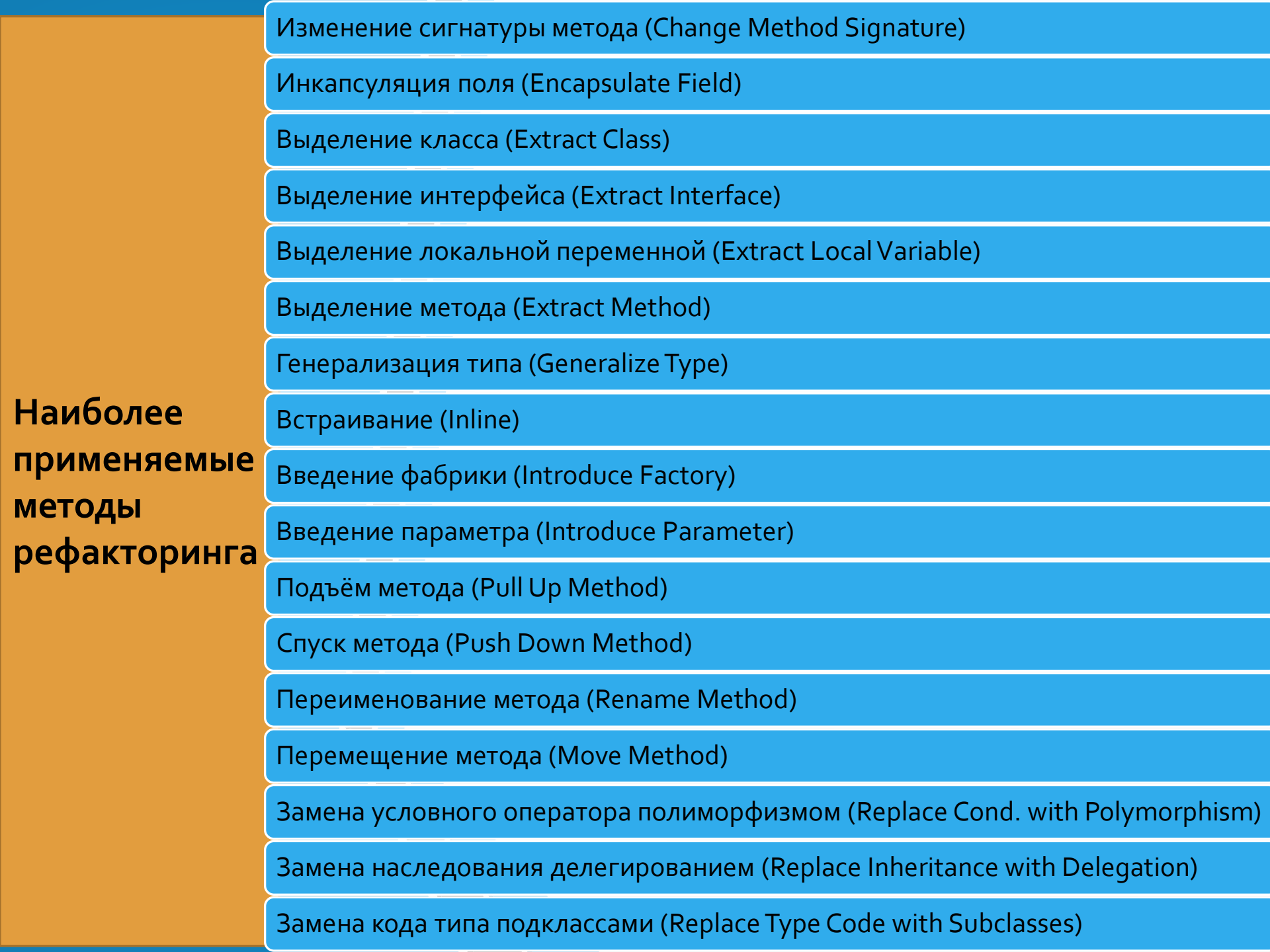
Большое количество
кода затрудняет его
понимание.


Скрытое значение.

Аномалии обновления.

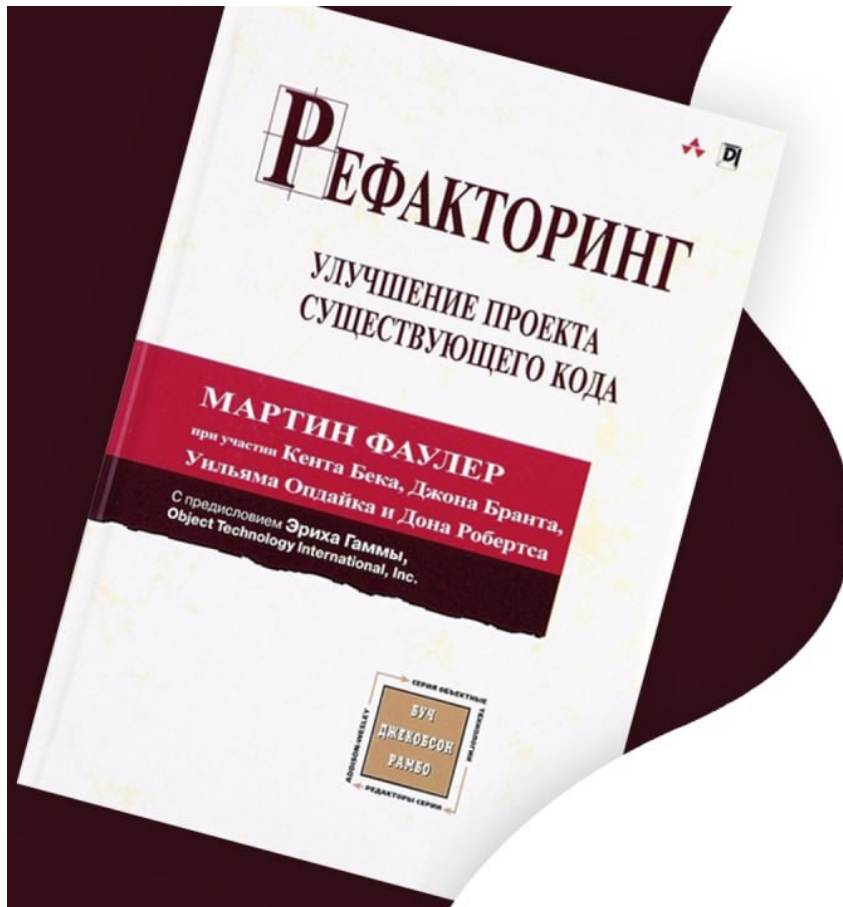


Методология рефакторинга

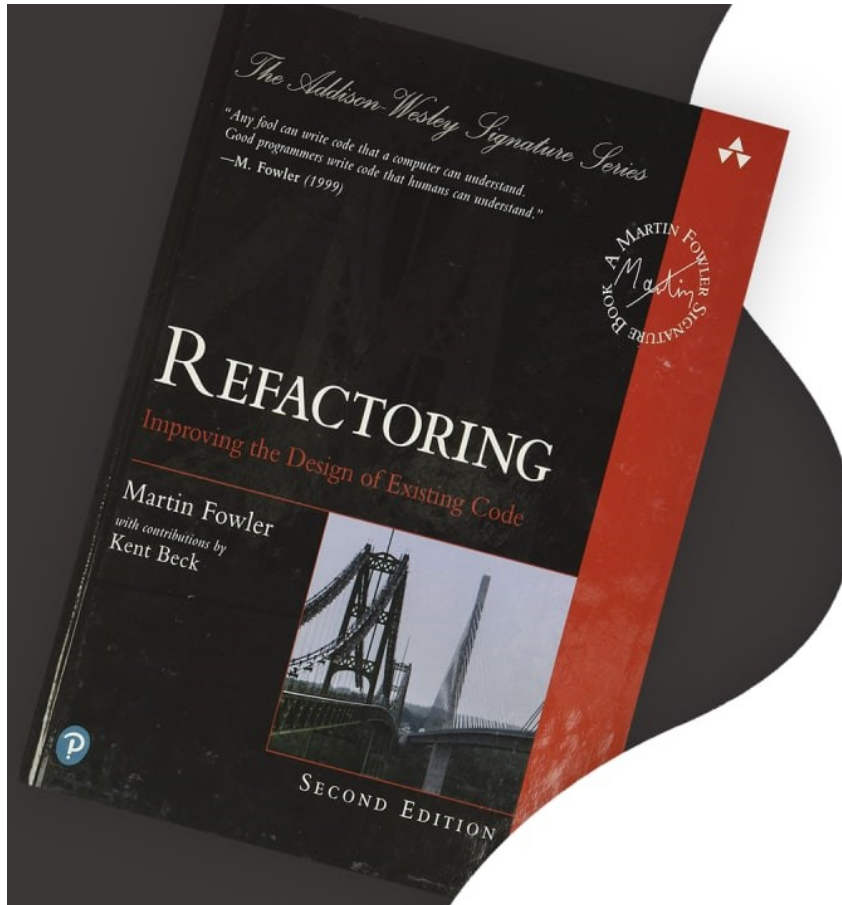




Список литературы



Рефакторинг.
Улучшение проекта
существующего кода
Фаулер Мартинх и другие



Refactoring

Martin Fowler

2018



Рефакторинг кода на JavaScript

Фаулер Мартин



Чистый код

Роберт Мартин