



# Паттерны проектирования

# Основные понятия

## В программировании

**Паттерны проектирования** – это наиболее распространенные подходы к решению типовых задач при разработке программного обеспечения

### Примеры паттернов в программировании

Абстрактная фабрика

Адаптер

Мост

Наблюдатель

Mvvm/ ecs

И т.д

## В жизни

**Паттерны** – это модели поведения, доведенные до автоматизма, часто повторяющиеся при сходных внешних условиях

### Примеры паттернов в реальной жизни

Алгоритм производства товара

Принцип по которому определяются точки маршрута

Адаптация к погодным условиям

Передача определенного спектра полномочий заместителю

И т.д.

# Классификация паттернов

## Порождающие

Описывают создание объектов без внесения в ПО лишних зависимостей

Абстрактная фабрика

Фабричный метод

Строитель

Одиночка

И т.д.

## Структурные

Определяют различные способы построения связей между объектами ПО

Адаптер

Мост

Декоратор

Фасад

И т.д.

## Поведенческие

Определяют эффективные способы коммуникации между объектами

Хранитель

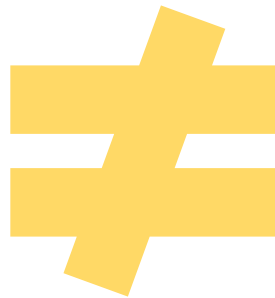
Наблюдатель

Состояние

Стратегия

И т.д.

Паттерн



Алгоритм

Улучшение читаемости кода

Улучшение повторного использования кода

Соккрытие сложности

Облегчение тестирования

## Недостатки использования паттернов проектирования

Затратность внедрения

Усложнение кода

Ограничение свободы действий

Сложность понимания

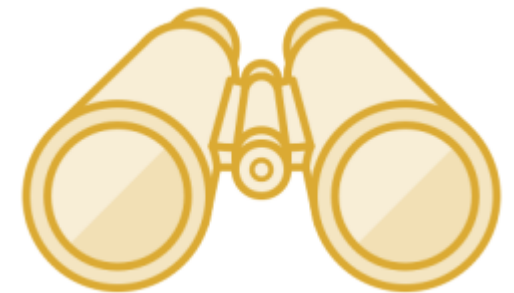


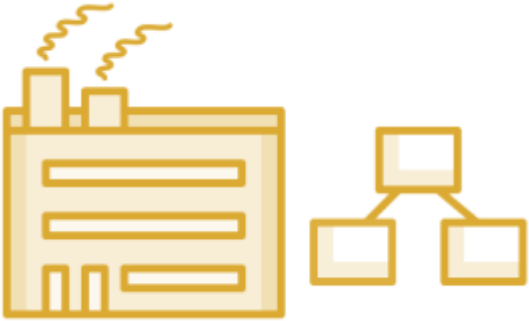
## Одиночка (singleton)

Гарантирует наличие только одного экземпляра класса

## Наблюдатель (observer)

Реализует оповещение «подписчиков» об изменениях одного объекта



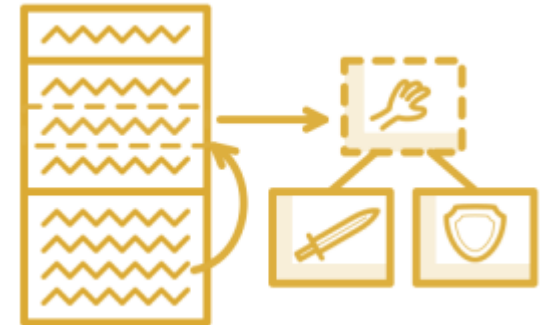


## Фабричный метод (Factory)

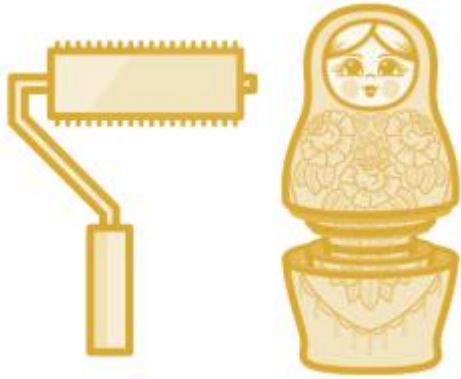
Используется при необходимости генерации объектов, тип которых будет меняться динамически

## Стратегия (Strategy)

Реализует смену алгоритма работы объекта во время выполнения







## Декоратор (Decorator)

Во время выполнения добавляет объектам новую функциональность

## Адаптер (Adapter)

Необходим для совмещения интерфейсов двух классов к общему интерфейсу

