







Введение в паттерны. Структурные

Что будет на уроке сегодня



Очередная группа паттернов!



Разбор группы паттернов «Структурные»



Как их применять



Adapter, Facade, Decorator - такие похожие, но разные



Как их называть



Чем паттерны отличаются



Повторим! Паттерны — это общепринятый подход реализации архитектуры программ

- 宁 Можно не знать паттернов и достичь цели
- 💡 Паттерны позволяют писать понятный код для других разработчиков
- 宁 Нельзя писать ради паттернов. Их надо использовать только там, где они уместны
- 💡 Паттерны не всегда уменьшают количество кода, а даже наоборот



Очередная группа паттернов - номер 2

1

Поведенческие

- 1. Chain of responsibility
- 2. Strategy
- 3. Visitor

2

Структурные (МЫ ТУТ)

- Adapter
- 2. Facade
- 3. Decorator

3

Порождающие

- 1. Factory Method
- 2. Abstract Factory
- 3. Builder
- 4. Singleton



Что такое структурные паттерны?

→ **Структурные** — удобное построение связей между частями кода и объектов



Структурные паттерны

Хаос присущ даже разработке. При использовании данных паттернов хаоса становится меньше. Хороший программист стремится контролировать хаос.



Decorator

- → Популярный паттерн
- → Декорируем объект и обрастаем новым функционалом
- → Прост в использовании



Facade

- → Популярный паттерн
- → Агрегируем и делаем сложные вещи простым через один фасад
- → Прост в использовании



Adapter

- → Популярный паттерн
- → Новый код нужно заставить со старым? Адаптер идеальное решение!
- → Прост в использовании



Практическое задание

1. Придумайте небольшое приложение консольного типа, который берет различные Json структуры (предположительно из разных веб сервисов), которые олицетворяют товар в магазинах. Структуры не похожи друг на друга, но вам нужно их учесть, сделать универсально. Структуры на ваше усмотрение.





Спасибо /// за внимание /