GeekBrains

Урок 5

Структурные паттерны

Обзор структурных паттернов. Adapter, Bridge, Composite, Decorator, Facade, Proxy

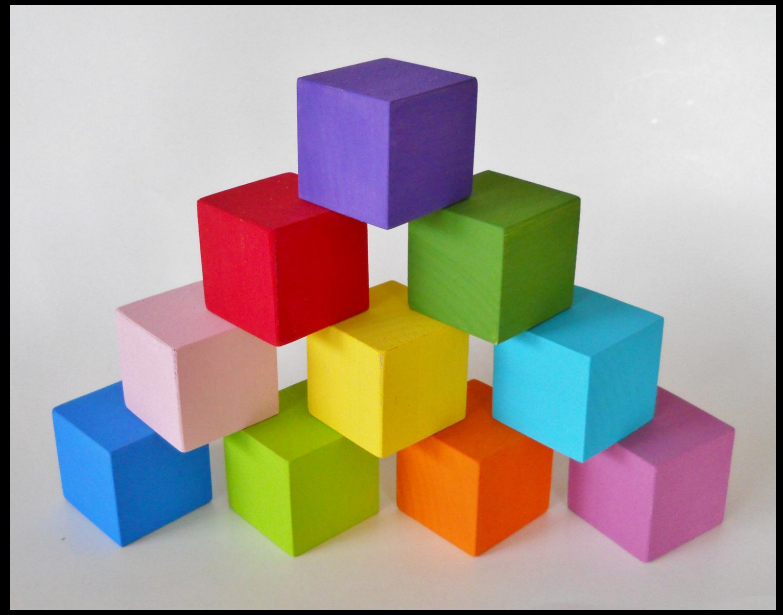
План урока

1. Структурные паттерны
2. Практика

Структурные паттерны

Структурные паттерны

Определяют, как из классов и объектов образуются более крупные структуры.



Структурные паттерны

1. Адаптер
2. Мост
3. Компоновщик
4. Декоратор
5. Фасад
6. Заместитель

Адаптер

Преобразует интерфейс одного класса в интерфейс другого, который ожидают клиенты.

Мост

Отделяет абстракцию от её реализации так, чтобы их можно было изменять независимо.

Компоновщик

Компонует объекты в древовидные иерархические структуры для представления иерархий «часть — целое»

Позволяет клиентам единообразно трактовать индивидуальные и составные объекты

Декоратор

Динамически добавляет объекту новые обязанности.

Фасад

Определяет интерфейс более высокого уровня, который упрощает использование подсистемы.

Заместитель

Позволяет сослаться на объект более изощрённо, чем это возможно с простым указателем.

Практическое задание

В этой самостоятельной работе тренируем умения:

Выбирать подходящий структурный шаблон

Применять структурные шаблоны в своём коде

Зачем:

Для использования структурных шаблонов в своём коде

Последователь ность действий:

Можно сделать всё или одно на выбор, применив при этом один из структурны

1. Создать декоратор для получения связки url-view в приложении, чтобы можно было добавлять url, как во фреймворке Flask @app(‘/some_url/’).

Последователь ность действий:

2. Добавить декоратор @debug для view; если мы указываем этот декоратор над view, то в терминал выводятся название функции и время её выполнения.

3. Добавить подкатегории. Т. е., категория курса может входить в другую категорию, а может не входить, и вложенность будет любая. Например:, «Программирование -> Веб -> Python -> Django». После на страницу списка категорий добавить вывод количества курсов в каждой из категорий. Например, Программирование — 10, Веб — 5, Python — 3, ...

Последователь ность действий:

4. Добавить два новых вида WSGI-application. Первый — логирующий (такой же, как основной, только для каждого запроса выводит информацию (тип запроса и параметры) в консоль. Второй — фейковый (на все запросы пользователя отвечает: 200 OK, Hello from Fake).
5. По желанию можно добавить любой другой полезный функционал

Спасибо!

Каждый день вы становитесь лучше :)