

ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ



Тест не пройден :(

georgiilsaev прошел тест с результатом 4 из 10

РЕЗУЛЬТАТЫ

16

Заработанные очки

junior

Сложность

40%

Правильных ответов

ОТВЕТЫ

#1 К какой группе относится паттерн проектирования "Decorator" (в соответствии с GoF)?

Структурные паттерны (structural)



Паттерны поведения (behavioral)

Паттерны создания объектов (creational)

Паттерн не относится ни к одной из перечисленных групп

[Обсудить вопрос](#)

Оцените сложность вопроса 

Легко



Сложно



#2 Выберите все верные утверждение про отношения паттернов Фасад (Facade) и Адаптер (Adapter)

фасад задает новый интерфейс, а адаптер повторно использует старый

фасад можно сделать адаптером, так как обычно нужен только один объект-фасад

фасад оборачивает подсистему, а адаптер оборачивает только один класс



адаптер позволяет двум существующим интерфейсам работать сообща, вместо задания нового как адаптер

Объяснение: Паттерн Фасад — это структурный паттерн проектирования, предоставляющий простой интерфейс к сложной системе классов, библиотеке или фреймворку, а Адаптер позволяет объектам с несовместимыми интерфейсами работать вместе.

Фасад задаёт новый интерфейс, тогда как Адаптер повторно использует старый. Адаптер оборачивает только один класс, а Фасад оборачивает целую подсистему. Более того, Адаптер позволяет двум существующим интерфейсам работать сообща, вместо того, чтобы задать полностью новый.

Оцените сложность вопроса

Легко

Сложно

#3 Какой шаблон проектирования можно определить следующей фразой: " .. - поведенческий шаблон проектирования, обеспечивающий взаимодействие множества объектов, формируя при этом слабую связанность и избавляя объекты от необходимости явно ссылаться друг на друга "?

Mediator

Low Coupling

High Cohesion

Information Expert

Observer

[Обсудить вопрос](#)

Оцените сложность вопроса

Легко

Сложно

#4 Какие из перечисленных утверждений верны по отношению к паттерну проектирования Адаптер (Adapter):

Паттерн преобразует интерфейс одного класса в интерфейс другого, который ожидают клиенты.


Паттерн определяет представителя другого объекта, не изменяя его интерфейс.

Структура паттерна Адаптер (Adapter) аналогична структуре паттерна Мост (Bridge).

Адаптер (Adapter) — это настоящий sibling реализации Замоскителя (Proxy)

Объяснение: 1. Структура паттерна мост аналогична структуре адаптера, но у моста иное назначение. Он отделяет интерфейс от реализации, чтобы то и другое можно было изменять независимо. Адаптер же призван изменить интерфейс существующего объекта. GoF

2. Замеситель (Proxy) определяет представителя или суррогат другого объекта, но не изменяет его интерфейс. GoF

Оцените сложность вопроса 

Легко

Сложно

#5 Поведенческий паттерн проектирования, который позволяет передавать запросы последовательно по цепочке обработчиков. Каждый последующий обработчик решает, может ли он обработать запрос сам и стоит ли передавать запрос дальше по цепи. Какой это паттерн?


компоновщик (composite)

цепочка обязанностей (chain of responsibility)



наблюдатель (observer)

Объяснение: Цепочка обязанностей (англ. CoR, Chain of Command, Chain of Responsibility) - паттерн который представляет из себя поведенческий паттерн проектирования, позволяющий передавать запросы последовательно по цепочке обработчиков. Каждый последующий обработчик решает, может ли он обработать запрос сам и стоит ли передавать запрос дальше по цепи.

Оцените сложность вопроса 

Легко

Сложно

#6 Какие основные задачи решает применение паттернов проектирования?

Производится унификация терминологии, названий модулей и элементов проекта.

Показываются отношения и взаимодействия между классами или объектами.

Облегчается дискуссия об абстрактных структурах данных между разработчиками. 

Описывается решение целого класса абстрактных проблем. 

Объяснение: Паттерны проектирования — это многократно применяемая архитектурная конструкция, предоставляющая решение общей проблемы проектирования в рамках конкретного контекста и описывающая значимость этого решения. Паттерн не является законченным образцом проекта, который может быть прямо преобразован в код (что отличает его от близкого понятия идиомы), скорее это описание или образец для того, как решить задачу, таким образом, чтобы это можно было использовать в различных ситуациях. Объектно-ориентированные паттерны зачастую показывают отношения и взаимодействия между классами или объектами, без определения того, какие конечные классы или объекты приложения будут использоваться.

Главная польза каждого отдельного паттерна состоит в том, что он описывает решение целого класса абстрактных проблем. Также тот факт, что каждый паттерн имеет свое имя, облегчает дискуссию об абстрактных структурах данных (ADT)

между разработчиками, так как они могут ссылаться на известные шаблоны. Таким образом, за счёт паттернов производится унификация терминологии, названий модулей и элементов проекта.

Оцените сложность вопроса

Легко

Сложно

#7 Какая разница между паттернами Facade и Adapter?

Adapter оборачивает один класс, тогда как Facade - несколько.

Adapter изменяет интерфейс класса так, чтобы он удовлетворял требованиям пользователя, а Facade предоставляет унифицированный интерфейс к некоторому множеству классов.

Паттерн Adapter убирает зависимость между клиентской реализацией и интерфейсом системы, а Facade вынуждает пользователя реализовывать определенный интерфейс.

Оба паттерна это разновидности одного и того же общего случая.

Ни один из перечисленных ответов не является верным.

[Обсудить вопрос](#)

Оцените сложность вопроса

Легко

Сложно

#8 Паттерн проектирования Factory Method обеспечивает тесное связывание между созданным объектом и использующим его кодом.

нет

[Обсудить вопрос](#)

Объяснение: Фабричный метод (Factory Method) представляет из себя порождающий паттерн проектирования, который определяет общий интерфейс для создания объектов в суперклассе, позволяя подклассам изменять тип создаваемых объектов.

Паттерн обеспечивает абстрагирование от конкретного класса-наследника и помогает создавать элементы интерфейса, не привязывая основной код программы к конкретным классам элементов.

Оцените сложность вопроса

Легко

Сложно

#9 Паттерны проектирования State и Strategy имеют одинаковую структуру классов.

Да

Нет

[Обсудить вопрос](#)

Оцените сложность вопроса

Легко

Сложно

#10 Какой из данных паттернов проектирования является структурным паттерном?

Singleton

Facade

Command

Visitor

[Обсудить вопрос](#)

Оцените сложность вопроса

Легко

Сложно

Наш сайт использует куки чтобы обеспечить качественную работу функционала сайта. Продолжая использовать сайт ProgHub, вы соглашаетесь с нашей политикой конфиденциальности.

ОК



support@mg.proghub.ru