ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ

Тест не пройден :(

georgiilsaevJRTRyE прошел тест с результатом 6 из 10

РЕЗУЛЬТАТЫ

24	junior	60%
Заработанные очки	Сложность	Правильных ответов

ОТВЕТЫ

#1 Целью какого паттерна является расширение функциональности класса или же ее изменение без использования механизма наследования?

- Proxy
- Decorator ✓
- Flyweight
- Composite

Обсудить вопрос

Объяснение: Структурный паттерн Decorator используется в случаях, когда необходимо без применения механизма наследования расширить функциональность класса или же изменить ее. Другими словами появляется альтернатива наследованию, причем классы не закреплены жестко в иерархии.

Оцените сложность вопроса

Легко ○ ○ ○ ○ ○ ○ ○ ○ ○ Сложно

#2 Назначением какого паттерна проектирования является предоставление удобного интерфейса к громоздкому и сложному API?

- Factory Method

Iterator

Decorator

Strategy

Facade

Обсудить вопрос

Оцените сложность вопроса

Легко Сложно

#3 Паттерн проектирования Template Method позволяет уменьшить количество зависимостей от классов, специфичных для данного приложения.

Да

Нет

Обсудить вопрос

Оцените сложность вопроса

Легко Сложно

#4 Какие из перечисленных паттернов проектирования ограничивают платформенные зависимости?

Flyweight

Adapter

Abstract Factory

Bridge

Interpreter

Обсудить вопрос

Объяснение: Паттерн Flyweight использует разделение для эффективной поддержки множества мелких объектов; Паттерн Adapter преобразует интерфейс одного класса в

интерфейс другого, который ожидают клиенты; Паттерн Interpreter для заданного языка определяет представление его грамматики, а также интерпретатор предложений этого языка; Паттерны Abstract Factory и Bridge отделяют абстракцию от ее реализации так, чтобы то и другое можно было изменять независимо. Это обеспечивает кроссплатформенность.

Оцените сложность вопроса

Легко Сложно

#5 Какие из перечисленных утверждений верны по отношению к паттерну проектирования Composite:

Composite паттерн содержит в себе несколько моделей поведения объекта и предоставляет возможность переключаться между ними.

Паттерн Composite позволяет работать с композитными и простыми объектами одинаковым образом.

Паттерн Composite организовывает объекты в древовидные структуры.

Composite паттерн упрощает работу с группой объектов не связанных типов.

Обсудить вопрос

Оцените сложность вопроса

Легко Сложно

#6 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

#7 Паттерн Фасад похож на паттерн ... тем, что замещает сложную подсистему и может сам её инициализировать. Но в отличие от паттерна Фасад, паттерн ... имеет тот же интерфейс, что его служебный объект, благодаря чему их можно взаимозаменять.

- адаптер
- декоратор
- заместитель

Обсудить вопрос

Объяснение: Заместитель (англ. Proxy) - представляет из себя структурный паттерн проектирования, позволяющий подставлять вместо реальных объектов специальные объекты-заменители. Эти объекты перехватывают вызовы к оригинальному объекту, позволяя сделать что-то до или после передачи вызова оригиналу.

Фасад (англ. Facade) похож на Заместитель тем, что замещает сложную подсистему и может сам её инициализировать. Но в отличие от Фасада, Заместитель имеет тот же интерфейс, что его служебный объект, благодаря чему их можно взаимозаменять.

Оцените сложность вопроса

Легко Сложно

#8 Какие из перечисленных паттернов проектирования относятся к порождающим?

- Prototype
- Command
- Strategy
- Factory
- Singleton

Обсудить вопрос

Объяснение: К порождающим шаблонам относятся: Abstract Factory (Абстрактная фабрика), Factory (Фабрика), Builder (Строитель), Singleton (Одиночка), Prototype (Прототип), Factory Method (Фабричный метод).Strategy, Command относятся к шаблонам поведения.

Оцените сложность вопроса

Легко Сложно

#9 Какие из перечисленных паттернов относятся к паттернам поведения (behavioral) в соответствии с GoF?

Flyweight

Proxy

Builder

Memento

Обсудить вопрос

Оцените сложность вопроса

Легко Сложно

#10 Паттерн проектирования ... обеспечивает существование одного экземпляра некоторого класса и предоставляет единую точку доступа к нему.

Adapter

Bridge

Singleton

Proxy

Decorator

Обсудить вопрос

Объяснение: Паттерн Одиночка (англ. Singleton) — это порождающий паттерн проектирования, гарантирующий, что у класса есть только один экземпляр, и предоставляющий к нему глобальную точку доступа. Данный паттерн решает сразу две проблемы, нарушая принцип единственной ответственности класса:

- Гарантирует наличие единственного экземпляра класса. Чаще всего это полезно для доступа к какому-то общему ресурсу, например, базе данных.
- Предоставляет глобальную точку доступа. Это не просто глобальная переменная, через которую можно достучаться к определённому объекту. Глобальные переменные не защищены от записи, поэтому любой код может подменять их значения без вашего ведома.

Оцените сложность вопроса 👇

Легко Сложно

support@mg.proghub.ru