



Топ контрибуторов

За последние сутки

Нет данных

Получение Q



Знаете ли Вы, что

Лучшие IT работодатели регулярно просматривают рейтинги и профили пользователей в поисках кандидатов. Для корректного отображения ваших данных рекомендуем заполнить ваш профиль и добавить информацию о вас и вашей профессии.



Опросы

Какой у Вас реальный опыт программирования?

меньше 1 года

1 - 2 года

более 3 лет

Ответить

Лента обновлений

Jul 23 01:57

ссылка

Комментарий от GeorgeMayoh:  
Вообще я слышал такую версию: Если класс просто соде...

Jul 22 21:30

ссылка

Комментарий от leonx7:  
Красиво подловил)

Jul 22 21:09

ссылка

Комментарий от leonx7:  
Вопрос хороший, но точно не для теста, где время ограниче...

Jul 21 20:08

ссылка

Добавлен вопрос в тест ASP.NET MVC

Jul 20 20:59

ссылка

Комментарий от brave\_t:  
Опечатка в отличии от

Статистика

Тестов: 153, вопросов: 8596.  
Пройдено: 464636 / 2269562.

Тест "Паттерны проектирования" - пройден

пройти еще раз

Правильных ответов: 18 / 20 (90 %)    требуется: 15

Тема	Результат
Классификация паттернов	3 / 4 (75 %)
Порождающие паттерны	1 / 2 (50 %)
Паттерны поведения	4 / 4 (100 %)
Структурные паттерны	4 / 4 (100 %)
Остальные	6 / 6 (100 %)

Дата прохождения теста: 23.07.2020 21:26

Паттерн проектирования Interpreter (Интерпретатор) также известен как:

Паттерны поведения

- ☐ Ни один из перечисленных
- ☐ Slow (Slower) Executor
- ☒ Little (Small) Language
- ☐ Translator
- ☐ Little (Small) Compiler

833 / 4702

207 / 4702

1577 / 4702

1602 / 4702

436 / 4702

Пояснение: Little language его называют потому, что интерпретировать обычно нужно какой-нибудь маленький придуманный программистом язык.

Комментировать вопрос (всего: 0)

Автор: admin

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

Классификация паттернов

- ☐ Flyweight
- ☐ Builder
- ☒ Memento
- ☐ Proxy

1389 / 4196

475 / 4196

3066 / 4196

1295 / 4196

Комментировать вопрос (всего: 4)

Автор: admin

Для какого из перечисленных паттернов лучше всего подходит метафора "матрешки"?

Структурные паттерны

- ☒ Decorator
- ☐ Abstract Factory
- ☐ Prototype
- ☐ Proxy
- ☐ Singleton

2930 / 4414

469 / 4414

574 / 4414

340 / 4414

81 / 4414

Комментировать вопрос (всего: 4)

Автор: admin

Какие из перечисленных паттернов проектирования ограничивают платформенные зависимости?

- ☐ Flyweight
- ☐ Adapter
- ☒ Abstract Factory
- ☒ Bridge
- ☐ Interpreter

541 / 2465

809 / 2465

927 / 2465

1454 / 2465

640 / 2465

Пояснение: 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.

Комментировать вопрос (всего: 1)

Автор: I\_shevchenko

Какие группы паттернов существуют по версии GoF?

- ☐ Core
- ☒ Structural
- ☐ Organizational
- ☐ Reusable
- ☒ Creational
- ☒ Behavioral

335 / 6679

6100 / 6679

898 / 6679

349 / 6679

5822 / 6679

5933 / 6679

Комментировать вопрос (всего: 1)

Автор: c0nst

Какая разница между паттернами Facade и Adapter?

Структурные паттерны

- ☒ Adapter изменяет интерфейс класса так, чтобы он удовлетворял требованиям пользователя, а Facade предоставляет унифицированный интерфейс к некоторому множеству объектов.
- ☐ Паттерн Adapter убирает зависимость между клиентской реализацией и интерфейсом системы, а Facade вынуждает пользователя реализовывать определенный интерфейс.
- ☐ Adapter оборачивает один класс, тогда как Facade - несколько.
- ☐ Ни один из перечисленных ответов не является верным.
- ☐ Оба паттерна это разновидности одного и того же общего случая.

4236 / 4615

470 / 4615

469 / 4615

183 / 4615

217 / 4615

Комментировать вопрос (всего: 1)

Автор: admin

Паттерн Strategy имеет следующие преимущества:

Паттерны поведения

- ☐ Гарантирует использование единственной стратегии во время выполнения программы.
- ☒ Отделяет алгоритм от класса, в котором он используется.
- ☐ Предоставляет разделение между моделью и представлением.
- ☒ Позволяет переключаться между алгоритмами во время выполнения программы.

1000 / 4530

3124 / 4530

614 / 4530

3466 / 4530

Пояснение: Паттерн Strategy - это паттерн поведения. Он позволяет отделить алгоритм от класса, который его использует. Кроме того, он позволяет переключаться между алгоритмами во время выполнения программы.

Комментировать вопрос (всего: 9)

Автор: admin

К какой группе относится паттерн проектирования "хранитель" ("Memento") (в соответствии с GoF)?		
<input type="radio"/>	к структурной	594 / 3263
<input checked="" type="radio"/>	к поведенческой	2354 / 3263
<input type="radio"/>	к порождающей	154 / 3263
<input type="radio"/>	ни к одной из перечисленных	136 / 3263
<a href="#">Комментировать вопрос</a> (всего: 2)		Автор: <a href="#">roman_v1</a>
Какой шаблон подменяет собой сложный объект и контролирует доступ к нему?		Структурные паттерны
<input type="radio"/>	Decorator	404 / 4401
<input type="radio"/>	Bridge	344 / 4401
<input type="radio"/>	Adapter	653 / 4401
<input checked="" type="radio"/>	Proxy	2975 / 4401
<a href="#">Комментировать вопрос</a> (всего: 1)		Автор: <a href="#">TestTol</a>
К какой группе относится паттерн проектирования "Decorator" (в соответствии с GoF)?		Классификация паттернов
<input type="radio"/>	Паттерны создания объектов (creational)	312 / 4224
<input type="radio"/>	Паттерн не относится ни к одной из перечисленных групп	149 / 4224
<input type="radio"/>	Паттерны поведения (behavioral)	720 / 4224
<input checked="" type="radio"/>	Структурные паттерны (structural)	3011 / 4224
<a href="#">Комментировать вопрос</a> (всего: 0)		Автор: <a href="#">admin</a>
Когда следует использовать паттерн "приспособленец"?		Классификация паттернов
<input checked="" type="checkbox"/>	когда большинство состояний объектов могут быть сохранены на диске или рассчитаны во время исполнения	715 / 2169
<input checked="" type="checkbox"/>	когда нужно сократить затраты при работе с большим количеством мелких объектов.	1264 / 2169
<input type="checkbox"/>	когда объект может иметь несколько представлений	739 / 2169
<input type="checkbox"/>	когда нужно изменить реализацию без изменения абстракции	618 / 2169
<a href="#">Комментировать вопрос</a> (всего: 5)		Автор: <a href="#">levilaulada</a>
При малом количестве объектов лучше использовать паттерн Facade вместо Mediator.		Классификация паттернов
<input type="radio"/>	Да	1612 / 4536
<input checked="" type="radio"/>	Нет	2891 / 4536
Пояснение: Эти паттерны относятся к разным категориям (Facade - структурный, Mediator - поведенческий) и между собой абсолютно не связаны.		
<a href="#">Комментировать вопрос</a> (всего: 3)		Автор: <a href="#">admin</a>
Какие из перечисленных утверждений верны по отношению к паттерну проектирования Composite:		Структурные паттерны
<input type="checkbox"/>	Composite паттерн содержит в себе несколько моделей поведения объекта и предоставляет возможность переключаться между ними.	755 / 4469
<input checked="" type="checkbox"/>	Паттерн Composite позволяет работать с композитными и простыми объектами одинаковым образом.	2919 / 4469
<input checked="" type="checkbox"/>	Паттерн Composite организует объекты в древовидные структуры.	2862 / 4469
<input type="checkbox"/>	Composite паттерн упрощает работу с группой объектов не связанных типов.	1317 / 4469
<a href="#">Комментировать вопрос</a> (всего: 2)		Автор: <a href="#">admin</a>
Паттерн проектирования Factory Method обеспечивает тесное связывание между созданным объектом и использующим его кодом.		Порождающие паттерны
<input type="radio"/>	Да	1561 / 7255
<input checked="" type="radio"/>	Нет	5662 / 7255
Пояснение: Наоборот, он обеспечивает абстрагирование от конкретного класса-наследника.		
<a href="#">Комментировать вопрос</a> (всего: 1)		Автор: <a href="#">admin</a>
Какие основные задачи решает применение паттернов проектирования?		
<input checked="" type="checkbox"/>	Облегчается дискуссия об абстрактных структурах данных между разработчиками.	4401 / 6745
<input checked="" type="checkbox"/>	Описывается решение целого класса абстрактных проблем.	5389 / 6745
<input checked="" type="checkbox"/>	Показываются отношения и взаимодействия между классами или объектами.	4459 / 6745
<input checked="" type="checkbox"/>	Производится унификация терминологии, названий модулей и элементов проекта.	4166 / 6745
Пояснение: 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.		
<a href="#">Комментировать вопрос</a> (всего: 5)		Автор: <a href="#">brat</a>
Что из перечисленного верно о паттерне Memento:		Паттерны поведения
<input type="checkbox"/>	Позволяет динамически добавлять новый функционал к существующим объектам	542 / 4193
<input type="checkbox"/>	Это структурный паттерн проектирования	1080 / 4193
<input type="checkbox"/>	Его использование приводит к нарушению инкапсуляции	551 / 4193
<input checked="" type="checkbox"/>	Позволяет сохранить внутреннее состояние объекта	3390 / 4193
<input type="checkbox"/>	Это порождающий паттерн проектирования	399 / 4193
Пояснение: Memento относится к паттернам поведения (behavioral). Он позволяет сохранить (например, для последующего восстановления) внутреннее состояние объекта без нарушения инкапсуляции.		
<a href="#">Комментировать вопрос</a> (всего: 0)		Автор: <a href="#">c0nst</a>
Паттерн проектирования ... обеспечивает существование одного экземпляра некоторого класса и предоставляет единую точку доступа к нему.		Порождающие паттерны
<input checked="" type="radio"/>	Singleton	6368 / 6925
<input type="radio"/>	Adapter	140 / 6925
<input type="radio"/>	Proxy	157 / 6925
<input type="radio"/>	Bridge	166 / 6925
<input type="radio"/>	Decorator	74 / 6925



