



Топ контрибуторов

За последние сутки

Нет данных

Получение Q



Знаете ли Вы, что

После прохождения теста можно комментировать вопросы теста, а Ваши комментарии увидят модераторы теста и пользователи, которым когда-либо эти вопросы попадались.



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 отличииЕ от

Статистика

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

Тест "Паттерны проектирования" - пройден

пройти еще раз

Правильных ответов: 17 / 20 (85 %) требуется: 15

Тема	Результат
Классификация паттернов	4 / 4 (100 %)
Порождающие паттерны	2 / 2 (100 %)
Паттерны поведения	4 / 4 (100 %)
Структурные паттерны	3 / 4 (75 %)
Остальные	4 / 6 (66 %)

Дата прохождения теста: 21.07.2020 15:35

✔ Паттерн проектирования Factory Method обеспечивает тесное связывание между созданным объектом и использующим его кодом.	Порождающие паттерны
<div>Да</div>	1561 / 7255
<div>✔ Нет</div>	5662 / 7255

Пояснение: Наоборот, он обеспечивает абстрагирование от конкретного класса-наследника.

Комментировать вопрос (всего: 1)

Автор: admin

✖ Какая разница между паттернами Facade и Adapter?	Структурные паттерны
<div>Оба паттерна это разновидности одного и того же общего случая.</div>	217 / 4615
<div>✔ Adapter изменяет интерфейс класса так, чтобы он удовлетворял требованиям пользователя, а Facade предоставляет унифицированный интерфейс к некоторому множеству объектов.</div>	6236 / 4615
<div>Ни один из перечисленных ответов не является верным.</div>	183 / 4615
<div>Паттерн Adapter убирает зависимость между клиентской реализацией и интерфейсом системы, а Facade вынуждает пользователя реализовывать определенный интерфейс.</div>	470 / 4615
<div>Adapter оборачивает один класс, тогда как Facade - несколько.</div>	469 / 4615

Комментировать вопрос (всего: 1)

Автор: admin

✔ Сколько существует разных типов (и соответственно принципиально разных способов реализации) паттерна Adapter?	Структурные паттерны
<div>1</div>	767 / 4317
<div>✔ 2</div>	2687 / 4317
<div>3</div>	634 / 4317
<div>4</div>	184 / 4317

Пояснение: Существует ClassAdapter, где используется множественное наследование, и ObjectAdapter, где используется композиция.

Комментировать вопрос (всего: 2)

Автор: c0nst

✔ Паттерн проектирования Template Method позволяет уменьшить количество зависимостей от классов, специфичных для данного приложения.	Паттерны поведения
<div>✔ Да</div>	2851 / 4494
<div>Нет</div>	1609 / 4494

Комментировать вопрос (всего: 2)

Автор: admin

✔ Какой паттерн проектирования используется для создания семейств зависимых между собой объектов?	Порождающие паттерны
<div>Builder</div>	673 / 7196
<div>Factory Method</div>	688 / 7196
<div>Dependency Injection</div>	1043 / 7196
<div>Prototype</div>	476 / 7196
<div>✔ Abstract Factory</div>	4279 / 7196

Комментировать вопрос (всего: 1)

Автор: admin

✔ Отметьте верные утверждения	
<div>✔ Абстрактная фабрика основана на композиции, ее задача - создание семейств взаимосвязанных объектов.</div>	1681 / 2537
<div>✔ Фабричный метод основан на наследовании, его задача - перемещение создания экземпляров в subclasses.</div>	1671 / 2537
<div>Фабричный метод основан на композиции, его задача - создание семейств взаимосвязанных объектов.</div>	632 / 2537
<div>Фабричная фабрика основана на наследовании, ее задача - перемещение создания экземпляров в subclasses.</div>	718 / 2537

Пояснение: Абстрактная фабрика основана на композиции: создание объектов реализуется в методе, доступ к которому осуществляется через интерфейс. Фабричный метод основан на наследовании: создание объектов делегируется subclasses, реализующим фабричный метод для создания объектов.

Комментировать вопрос (всего: 3)

Автор: GalinaU

✔ Какой шаблон подменяет собой сложный объект и контролирует доступ к нему?	Структурные паттерны
<div>Decorator</div>	404 / 4401
<div>✔ Proxy</div>	2975 / 4401
<div>Adapter</div>	653 / 4401
<div>Bridge</div>	344 / 4401

Комментировать вопрос (всего: 1)

Автор: TestTol

✔ Целью какого паттерна является расширение функциональности класса или же ее изменение без использования механизма наследования?	Структурные паттерны
<div>✔ Decorator</div>	3044 / 4271
<div>Flyweight</div>	232 / 4271
<div>Composite</div>	648 / 4271
<div>Proxy</div>	325 / 4271

Пояснение: 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.

Комментировать вопрос (всего: 2)

Автор: nobleclone

✅	Какие основные задачи решает применение паттернов проектирования?	
🗒️	✅ Производится унификация терминологии, названий модулей и элементов проекта.	4166 / 6745
🗒️	✅ Описывается решение целого класса абстрактных проблем.	5389 / 6745
🗒️	✅ Облегчается дискуссия об абстрактных структурах данных между разработчиками.	4401 / 6745
🗒️	✅ Показываются отношения и взаимодействия между классами или объектами.	4459 / 6745
<i>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.</i>		
Комментировать вопрос (всего: 5)		Автор: brat

❌	Какое назначение паттерна Factory Method?	
🗒️	✅ Определяет интерфейс для создания объекта, но оставляет подклассам решение о том, какой класс инстанцировать. Этот паттерн позволяет классу делегировать инстанцирование подклассам.	2172 / 3248
🗒️	❌ Отделяет конструирование сложного объекта от его представления, так что в результате одного и того же процесса конструирования могут получаться разные представления	358 / 3248
🗒️	❌ Предоставляет интерфейс для создания семейств взаимосвязанных или взаимозависимых объектов, не специфицируя их конкретных подклассов.	525 / 3248
🗒️	❌ Задаёт виды создаваемых объектов с помощью экземпляра-прототипа и создаёт новые объекты путем копирования этого прототипа.	172 / 3248
Комментировать вопрос (всего: 0)		Автор: ruslan5t

✅	Какой из данных паттернов проектирования является структурным паттерном?	Классификация паттернов
🗒️	❌ Singleton	439 / 4511
🗒️	✅ Facade	3382 / 4511
🗒️	❌ Command	357 / 4511
🗒️	❌ Visitor	313 / 4511
Комментировать вопрос (всего: 0)		Автор: admin

✅	Верно ли высказывание: Различие между паттернами посредник (Mediator) и наблюдатель (Observer) в том, что наблюдатель распределяет обмен информацией между объектами, а посредник наоборот, инкапсулирует взаимодействие между другими объектами.	Паттерны поведения
🗒️	✅ да	3284 / 4270
🗒️	❌ нет	959 / 4270
<i>Пояснение: см. паттерны поведения</i>		
Комментировать вопрос (всего: 1)		Автор: Alenka

✅	При малом количестве объектов лучше использовать паттерн Facade вместо Mediator.	Классификация паттернов
🗒️	❌ Да	1612 / 4536
🗒️	✅ Нет	2891 / 4536
<i>Пояснение: Эти паттерны относятся к разным категориям (Facade - структурный, Mediator - поведенческий) и между собой абсолютно не связаны.</i>		
Комментировать вопрос (всего: 3)		Автор: admin

✅	Что из перечисленного верно о паттерне Memento:	Паттерны поведения
🗒️	✅ Позволяет сохранить внутреннее состояние объекта	3390 / 4193
🗒️	❌ Позволяет динамически добавлять новый функционал к существующим объектам	542 / 4193
🗒️	❌ Это структурный паттерн проектирования	1080 / 4193
🗒️	❌ Его использование приводит к нарушению инкапсуляции	551 / 4193
🗒️	❌ Это порождающий паттерн проектирования	399 / 4193
<i>Пояснение: Memento относится к паттернам поведения (behavioral). Он позволяет сохранить (например, для последующего восстановления) внутреннее состояние объекта без нарушения инкапсуляции.</i>		
Комментировать вопрос (всего: 0)		Автор: c0nst

❌	Какие из перечисленных паттернов проектирования относятся к поведенческим?	
🗒️	✅ Интерпретатор	2478 / 3234
🗒️	❌ Компоновщик	662 / 3234
🗒️	✅ Хранитель	2094 / 3234
🗒️	❌ Строитель	301 / 3234
🗒️	✅ Посредник	2309 / 3234
🗒️	❌ Адаптер	716 / 3234
Комментировать вопрос (всего: 5)		Автор: ruslan5t

✅	Паттерны проектирования State и Strategy имеют одинаковую структуру классов.	Паттерны поведения
🗒️	✅ Да	2063 / 4657
🗒️	❌ Нет	2565 / 4657
Комментировать вопрос (всего: 2)		Автор: admin

✅	Какие из перечисленных паттернов относятся к паттернам поведения (behavioral) в соответствии с GoF?	Классификация паттернов
🗒️	❌ Proxy	1295 / 4196
🗒️	❌ Builder	475 / 4196
🗒️	❌ Flyweight	1389 / 4196
🗒️	✅ Memento	3066 / 4196
Комментировать вопрос (всего: 4)		Автор: admin

✅	Какие из перечисленных паттернов проектирования ограничивают платформенные зависимости?	
🗒️	❌ Flyweight	541 / 2465
🗒️	❌ Adapter	809 / 2465


<div> <div>✓</div> <div>К какой группе относится паттерн проектирования "хранитель" ("Memento") (в соответствии с GoF)?</div> </div>			
<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

[Предложить свой вопрос](#)

[Share](#)[tweet](#)

Важно: Ниже обсуждается только тест в целом: покрытие тематик, типы вопросов, пути улучшения теста и так далее.

Замечания, касающиеся отдельных вопросов теста, просьба оставлять в соответствующих топиках, которые доступны на странице результатов прохождения теста возле каждого вопроса.

20/20 [ОТВЕТИТЬ](#)  **myrko1992** 06.06.2013 | 02:16:59 #  

Верно, знания приходят при изучении теории и закрепления их на практике.
А тест - это оценка уже имеющихся знаний.
Если нужно проверить себя - то поможет тест, нужно научиться - практика и чтение.

Например реализовав что-то:

Стандартизация очень помогает при разработке ПО :)

Для успешного прохождения теста ознакомьтесь
http://ru.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1

А когда откроете ? Я бы тоже добавил вопросы

KvanGee 27.04.2010 | 17:21:40 #

ОТВЕТИТЬ  admin 29.04.2010 | 01:07:32 #

[ответить](#)

Добавить комментарии

Отправить

