



Топ контрибуторов

За последние сутки

Нет данных

Получение Q

Знаете ли Вы, что

Список полученных сертификатов находится на странице Вашего профиля. Сертификаты можно распечатать или разместить на Вашем сайте.

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 отличиЕ от

Статистика

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

Тест "Паттерны проектирования" - пройден

пройти еще раз

Правильных ответов: 19 / 20 (95 %)

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	6 / 6 (100 %)

Дата прохождения теста: 20.07.2020 23:01

<div>Паттерн Strategy имеет следующие преимущества:</div>	Паттерны поведения
<div>Позволяет переключаться между алгоритмами во время выполнения программы.</div>	3466 / 4530
<div>Отделяет алгоритм от класса, в котором он используется.</div>	3124 / 4530
<div>Гарантирует использование единственной стратегии во время выполнения программы.</div>	1000 / 4530
<div>Предоставляет разделение между моделью и представлением.</div>	614 / 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

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

Комментировать вопрос

(всего: 0)

Автор: ruslan5t

<div>Паттерн проектирования ... обеспечивает существование одного экземпляра некоторого класса и предоставляет единую точку доступа к нему.</div>	Порождающие паттерны
<div>Adapter</div>	140 / 6925
<div>Proxy</div>	157 / 6925
<div>Bridge</div>	166 / 6925
<div>Singleton</div>	6368 / 6925
<div>Decorator</div>	74 / 6925

Комментировать вопрос

(всего: 0)

Автор: c0nst

<div>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 "?</div>	
<div>Mediator</div>	3595 / 7611
<div>Low Coupling</div>	1884 / 7611
<div>High Cohesion</div>	346 / 7611
<div>Information Expert</div>	244 / 7611
<div>Observer</div>	1431 / 7611

Комментировать вопрос

(всего: 2)

Автор: gurn

<div>Какой из перечисленных паттернов применяется в ситуации, когда нужно произвести некоторые действия над всеми объектами структуры в зависимости от фактического типа каждого объекта?</div>	Паттерны поведения
<div>State</div>	254 / 4337
<div>Composite</div>	619 / 4337
<div>Iterator</div>	926 / 4337
<div>Visitor</div>	1801 / 4337
<div>Strategy</div>	707 / 4337

Комментировать вопрос

(всего: 1)

Автор: alenaiva

<div>Что является преимуществом использования паттернов проектирования?</div>	
<div>Они уменьшают количество проектной документации</div>	935 / 7824
<div>Они предоставляют проверенные техники решения задач</div>	6495 / 7824
<div>Они упрощают разработку и поддержку пользовательских интерфейсов</div>	1825 / 7824
<div>Они снижают затраты на разработку, так как они уже реализованы и их можно использовать без изменений</div>	1567 / 7824
<div>Они предоставляют механизмы для тестирования модулей системы</div>	515 / 7824

Комментировать вопрос

(всего: 14)

Автор: admin

<div>Какая разница между паттернами Facade и Adapter?</div>	Структурные паттерны
<div>Оба паттерна это разновидности одного и того же общего случая.</div>	217 / 4615
<div>Adapter оборачивает один класс, тогда как Facade - несколько.</div>	469 / 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 изменяет интерфейс класса так, чтобы он удовлетворял требованиям пользователя, а Facade предоставляет унифицированный интерфейс к некоторому множеству объектов.</div>	8236 / 4615

Комментировать вопрос

(всего: 1)

Автор: admin

<div>Сколько существует разных типов (и соответственно принципиально разных способов реализации) паттерна Adapter?</div>	Структурные паттерны
--	----------------------

1	767 / 4317	
<div><div><div><div><div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div></div></div></div></div><div><div>2</div><div>2687 / 4317</div></div></div>	Паттерны поведения	
<div><div><div><div><div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div></div></div></div></div><div><div>3</div><div>634 / 4317</div></div></div>		
<div><div><div><div><div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div></div></div></div></div><div><div>4</div><div>184 / 4317</div></div></div>		
<div>Пояснение: Существует ClassAdapter, где используется множественное наследование, и ObjectAdapter, где используется композиция.</div>		
<div>Комментировать вопрос (всего: 2)</div>		Автор: c0nst
<div><div><div><div><div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div></div></div></div></div><div><div>Паттерны проектирования State и Strategy имеют одинаковую структуру классов.</div><div>Да</div><div>Нет</div></div><div>2063 / 4657</div><div>2565 / 4657</div></div>	Паттерны поведения	
<div>Комментировать вопрос (всего: 2)</div>		
Автор: admin		
<div><div><div><div><div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div></div></div></div></div><div><div>Для какого из перечисленных паттернов лучше всего подходит метафора "матрешки"?</div><div>Decorator</div><div>Abstract Factory</div><div>Singleton</div><div>Proxy</div><div>Prototype</div></div><div>2930 / 4414</div><div>469 / 4414</div><div>81 / 4414</div><div>340 / 4414</div><div>574 / 4414</div></div>	Структурные паттерны	
<div>Комментировать вопрос (всего: 4)</div>		
Автор: admin		
<div><div><div><div><div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div></div></div></div></div><div><div>К какой группе относится паттерн проектирования "Decorator" (в соответствии с GoF)?</div><div>Паттерны поведения (behavioral)</div><div>Структурные паттерны (structural)</div><div>Паттерн не относится ни к одной из перечисленных групп</div><div>Паттерны создания объектов (creational)</div></div><div>720 / 4224</div><div>3011 / 4224</div><div>149 / 4224</div><div>312 / 4224</div></div>	Классификация паттернов	
<div>Комментировать вопрос (всего: 0)</div>		
Автор: admin		
<div><div><div><div><div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div></div></div></div></div><div><div>Какие из перечисленных утверждений верны по отношению к паттерну проектирования Адаптер (Adapter):</div><div><div><div><div><div><div></div></div></div><div><div><div></div></div></div></div>Паттерн преобразует интерфейс одного класса в интерфейс другого, который ожидают клиенты.</div><div><div><div><div><div><div></div></div></div><div><div><div></div></div></div></div>Паттерн определяет представителя другого объекта, не изменяя его интерфейс.</div><div><div><div><div><div><div></div></div></div><div><div><div></div></div></div></div>Структура паттерна Адаптер (Adapter) аналогична структуре паттерна Мост (Bridge).</div><div><div><div><div><div><div></div></div></div><div><div><div></div></div></div></div>Адаптер (Adapter) - это частичный случай реализации Замесителя (Proxy).</div></div><div>1899 / 2223</div><div>585 / 2223</div><div>779 / 2223</div><div>611 / 2223</div></div></div></div></div></div>	Классификация паттернов	
<div>Пояснение: 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</div>		
<div>Комментировать вопрос (всего: 2)</div>		
Автор: shadowss		
<div><div><div><div><div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div></div></div></div></div><div><div>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.</div><div>да</div><div>нет</div></div><div>3284 / 4270</div><div>959 / 4270</div></div>	Паттерны поведения	
<div>Пояснение: см. паттерны поведения</div>		
<div>Комментировать вопрос (всего: 1)</div>		
Автор: Alenka		
<div><div><div><div><div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div></div></div></div></div><div><div>Какие из этих паттернов являются паттернами создания объектов?</div><div>Abstract Factory</div><div>Adapter</div><div>Decorator</div><div>Chain of Responsibility</div></div><div>4212 / 4529</div><div>383 / 4529</div><div>582 / 4529</div><div>417 / 4529</div></div>	Классификация паттернов	
<div>Комментировать вопрос (всего: 0)</div>		
Автор: admin		
<div><div><div><div><div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div><div></div></div></div><div><div><div></div></div></div></div></div><div><div>Целью какого паттерна является расширение функциональности класса или же ее изменение без использования механизма наследования?</div><div>Flyweight</div><div>Decorator</div><div>Composite</div><div>Proxy</div></div><div>232 / 4271</div><div>3044 / 4271</div><div>648 / 4271</div><div>325 / 4271</div></div>	Структурные паттерны	
<div>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.</div>		
<div>Комментировать вопрос (всего: 2)</div>		
Автор: nobleclone		

Отметьте верные утверждения

Абстрактная фабрика основана на композиции, ее задача - создание семейств взаимосвязанных объектов.

Фабричный метод основан на наследовании, его задача - перемещение создания экземпляров в subclasses.

Фабричный метод основан на композиции, его задача - создание семейств взаимосвязанных объектов.

Абстрактная фабрика основана на наследовании, ее задача - перемещение создания экземпляров в subclasses.

1681 / 2537

1671 / 2537

632 / 2537

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

Паттерн проектирования Factory Method обеспечивает тесное связывание между созданным объектом и использующим его кодом.		Порождающие паттерны
	Да	1561 / 7255
	✓ Нет	5662 / 7255
Пояснение: Наоборот, он обеспечивает абстрагирование от конкретного класса-наследника.		
Комментировать вопрос (всего: 1)		Автор: admin

Какой из данных паттернов проектирования является структурным паттерном?		Классификация паттернов
<input type="radio"/> Singleton		439 / 4511
<input checked="" type="radio"/> Facade		3382 / 4511
<input type="radio"/> Command		357 / 4511
<input type="radio"/> Visitor		313 / 4511

[Предложить свой вопрос](#)

ShareTweet

Комментариев: 8 [↑ обновить](#)

 vasy1Klinduhov 10.02.2015 | 03:01:03 #

 **myrko1992** 06.06.2013 | 02:16:59 #

А какой смысл этого теста?

[ОТВЕТИТЬ](#)

 shadowss

18.06.2014 | 13:31:49

#

Незнание паттернов не делает плохим программистом.

1. специалист который владеет терминологией паттернов скажет: "Для приведения интерфейса А в нужный нам я реализовал класс адаптера АдаптерА ...".
2. или же скажет: "Для приведения интерфейса класса А к подходящему нам, я реализовал класс АдаптерА ...".

YuriyDunko 29.06.2012 | 13:10:10 #

[ОТВЕТИТЬ](#)
 KvanGee 27.04.2010 | 17:21:40

Добавление вопросов открыто для теста "Паттерны проектирования в Java". Этот тест временно закрыт для добавления, но в скором времени откроем и его.

ответить

Добавить комментарий

