



Топ контрибуторов

За последние сутки

Нет данных

Получение Q

Знаете ли Вы, что

Вы можете подписаться на RSS ленту новых тестов сервиса Quizful, в том числе и отдельно по каждой категории

Опросы

Какая область из перечисленных Вам наиболее интересна?

- Мобильные технологии
- Тестирование ПО
- Защита информации
- SEO / Интернет маркетинг
- Компьютерные сети

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 отличииE от

Статистика

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

Тест "Паттерны проектирования" - пройден

пройти еще раз

Правильных ответов: 16 / 20 (80 %)    требуется: 15

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

Дата прохождения теста: 22.07.2020 18:19

Какие из перечисленных паттернов проектирования относятся к поведенческим?

<input checked="" type="checkbox"/> Интерпретатор	2478 / 3234
<input type="checkbox"/> Компоновщик	662 / 3234
<input checked="" type="checkbox"/> Хранитель	2094 / 3234
<input type="checkbox"/> Строитель	301 / 3234
<input checked="" type="checkbox"/> Посредник	2309 / 3234
<input type="checkbox"/> Адаптер	716 / 3234

Комментировать вопрос (всего: 5)

Автор: ruslan5t

Какие из перечисленных паттернов проектирования относятся к порождающим?

Классификация паттернов

<input type="checkbox"/> Command	201 / 4272
<input type="checkbox"/> Strategy	277 / 4272
<input checked="" type="checkbox"/> Factory	3814 / 4272
<input checked="" type="checkbox"/> Singleton	3253 / 4272
<input checked="" type="checkbox"/> Prototype	3307 / 4272

Пояснение: 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.

Комментировать вопрос (всего: 5)

Автор: TestTol

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

Классификация паттернов

<input checked="" type="checkbox"/> Паттерн преобразует интерфейс одного класса в интерфейс другого, который ожидают клиенты.	1899 / 2223
<input type="checkbox"/> Паттерн определяет представителя другого объекта, не изменяя его интерфейс.	585 / 2223
<input checked="" type="checkbox"/> Структура паттерна Адаптер (Adapter) аналогична структуре паттерна Мост (Bridge).	779 / 2223
<input type="checkbox"/> Адаптер (Adapter) - это частичный случай реализации Замесителя (Proxy).	611 / 2223

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

Комментировать вопрос (всего: 2)

Автор: shadowss

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

Паттерны поведения

<input checked="" type="radio"/> Да	2851 / 4494
<input type="radio"/> Нет	1609 / 4494

Комментировать вопрос (всего: 2)

Автор: admin

Какой шаблон подменяет собой сложный объект и контролирует доступ к нему?

Структурные паттерны

<input type="radio"/> Bridge	344 / 4401
<input type="radio"/> Decorator	404 / 4401
<input checked="" type="radio"/> Proxy	2975 / 4401
<input type="radio"/> Adapter	653 / 4401

Комментировать вопрос (всего: 1)

Автор: TestTol

Сколько существует разных типов (и соответственно принципиально разных способов реализации) паттерна Adapter?

Структурные паттерны

<input type="radio"/> 1	767 / 4317
<input checked="" type="radio"/> 2	2687 / 4317
<input type="radio"/> 3	634 / 4317
<input type="radio"/> 4	184 / 4317

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

Комментировать вопрос (всего: 2)

Автор: c0nst

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

Порождающие паттерны

<input type="radio"/> Proxy	157 / 6925
<input type="radio"/> Decorator	74 / 6925
<input checked="" type="radio"/> Singleton	6368 / 6925
<input type="radio"/> Bridge	166 / 6925
<input type="radio"/> Adapter	140 / 6925






Комментировать вопрос (всего: 0)







Автор: c0nst







Какая разница между паттернами Facade и Adapter?








Структурные паттерны








<input type="radio"/> Ни один из перечисленных ответов не является верным.	183 / 4615
--	------------





	Adapter обращивает один класс, тогда как Facade - несколько.	469 / 4615
	 Adapter изменяет интерфейс класса так, чтобы он удовлетворял требованиям пользователя, а Facade предоставляет унифицированный интерфейс к некоторому множеству классов.	8236 / 4615
	Паттерн Adapter убирает зависимость между клиентской реализацией и интерфейсом системы, а Facade вынуждает пользователя реализовывать определенный интерфейс.	470 / 4615
	Оба паттерна это разновидности одного и того же общего случая.	217 / 4615
<a href="#">Комментировать вопрос</a> (всего: 1)		Автор: <a href="#">admin</a>





	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
<a href="#">Комментировать вопрос</a> (всего: 4)		Автор: <a href="#">admin</a>









	К какой группе относится паттерн проектирования "хранитель" ("Memento") (в соответствии с GoF)?	
	к структурной	594 / 3263
	 к поведенческой	2354 / 3263
	к порождающей	154 / 3263
	ни к одной из перечисленных	136 / 3263
<a href="#">Комментировать вопрос</a> (всего: 2)		Автор: <a href="#">roman_y1</a>





	Когда следует использовать паттерн "приспособленец"?	Классификация паттернов
	 когда большинство состояний объектов могут быть сохранены на диске или рассчитаны во время исполнения	715 / 2169
	 когда нужно сократить затраты при работе с большим количеством мелких объектов.	1264 / 2169
	когда объект может иметь несколько представлений	739 / 2169
	когда нужно изменить реализацию без изменения абстракции	618 / 2169
<a href="#">Комментировать вопрос</a> (всего: 5)		Автор: <a href="#">levilaulada</a>




	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	3595 / 7611
	Low Coupling	1884 / 7611
	High Cohesion	346 / 7611
	Information Expert	244 / 7611
	Observer	1431 / 7611
<a href="#">Комментировать вопрос</a> (всего: 2)		Автор: <a href="#">gurn</a>

	Какие из следующих утверждений верны по отношению к паттерну проектирования Mediator (Медиатор):	Паттерны поведения
	 Паттерн Mediator помогает изолировать несколько классов друг от друга.	2346 / 4520
	 Паттерн Mediator не позволяет нескольким классам взаимодействовать напрямую, а выступает посредником между ними.	3417 / 4520
	 Одна из целей паттерна Mediator - уменьшение зависимостей между объектами в системе.	3364 / 4520
	В реализации паттерна Mediator используется паттерн Observer для уменьшения зависимостей между объектами.	642 / 4520
	 Паттерн Mediator не спасает от зависимостей между классами системы и медиатором.	1848 / 4520
<a href="#">Комментировать вопрос</a> (всего: 5)		Автор: <a href="#">admin</a>

	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>		
<a href="#">Комментировать вопрос</a> (всего: 1)		Автор: <a href="#">Alenka</a>

	Какие из перечисленных паттернов проектирования ограничивают платформенные зависимости?	
	Flyweight	541 / 2465
	Adapter	809 / 2465
	 Abstract Factory	927 / 2465
	 Bridge	1454 / 2465
	Interpreter	640 / 2465
<i>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.</i>		
<a href="#">Комментировать вопрос</a> (всего: 1)		Автор: <a href="#">I_shevchenko</a>

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

	Целью какого паттерна является расширение функциональности класса или же ее изменение без использования механизма наследования?	Структурные паттерны
	Flyweight	232 / 4271
	Composite	648 / 4271
	 Decorator	3044 / 4271
	Proxy	325 / 4271
<i>Пояснение: Структурный паттерн Decorator используется в случаях, когда необходимо без применения механизма наследования расширить функциональность класса или же изменить</i>		



Отправить