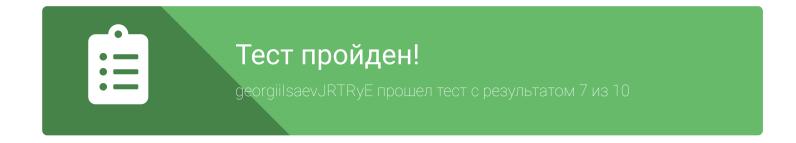
ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ 🔗



Как вам тест?

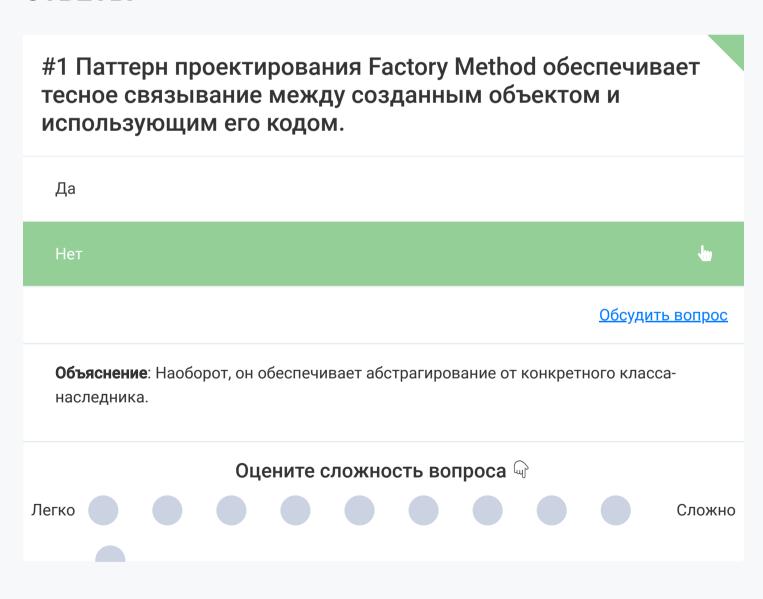


РЕЗУЛЬТАТЫ

28 Заработанные очки junior 70%

Правильных ответов

ОТВЕТЫ

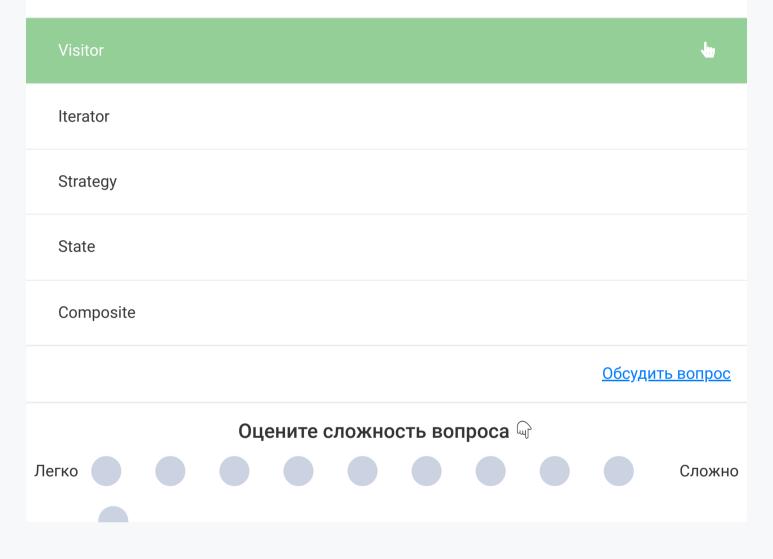


#2 Какие из этих паттернов являются паттернами создания объектов?



Decorator		
Chain of Responsibility		
	<u>Обсудит</u>	<u>ь вопрос</u>
Оцените сложность вопроса 🖓		
Легко		Сложно

#3 Какой из перечисленных паттернов применяется в ситуации, когда нужно произвести некоторые действия над всеми объектами структуры в зависимости от фактического типа каждого объекта?



#4 Когда следует использовать паттерн "приспособленец"?

когда большинство состояний объектов могут быть сохранены на диске или рассчитаны во время исполнения

когда нужно сократить затраты при работе с большим количеством мелких объектов.

когда объект может иметь несколько представлений

когда нужно изменить реализацию без изменения абстракции

Оцените сложность вопроса 🖓

Легко Сложно

#5 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

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

Proxy

Decorator

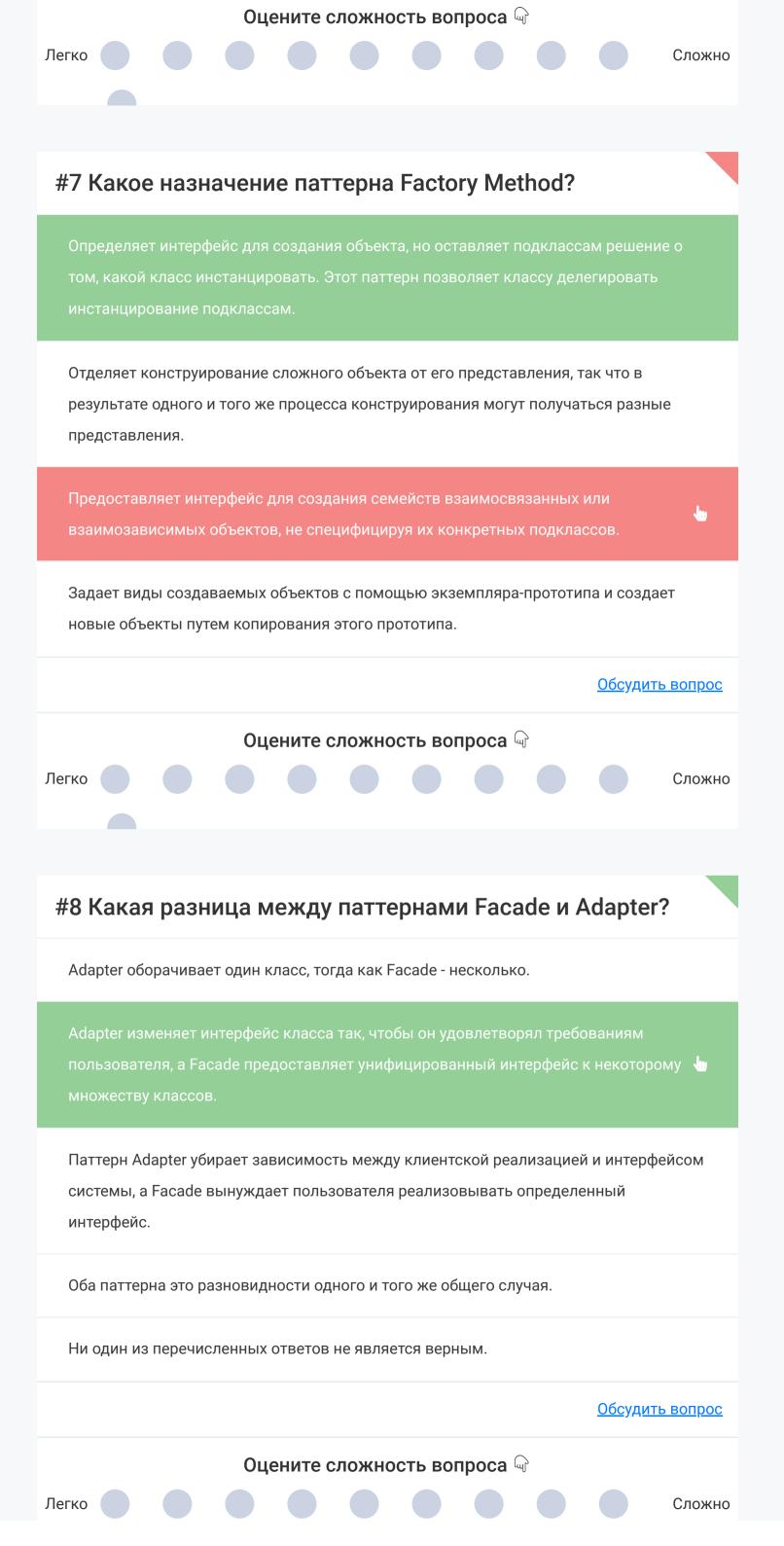


Flyweight

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.



#9 Паттерны проектирования State и Strategy имеют одинаковую структуру классов.

