

ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ



Тест пройден!

georgiisaevJRTRyE прошел тест с результатом 7 из 10

Как вам тест?



РЕЗУЛЬТАТЫ

28

Заработанные очки

junior

Сложность

70%

Правильных ответов

ОТВЕТЫ

#1 Паттерн проектирования Factory Method обеспечивает тесное связывание между созданным объектом и использующим его кодом.

Да

Нет



[Обсудить вопрос](#)

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

Оцените сложность вопроса 

Легко



Сложно



#2 Какие из этих паттернов являются паттернами создания объектов?

Abstract Factory



Adapter

Decorator

Chain of Responsibility

[Обсудить вопрос](#)

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?

Visitor

Iterator

Strategy

State

Composite

[Обсудить вопрос](#)

Оцените сложность вопроса

Легко

Сложно

#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

[Обсудить вопрос](#)

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

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.

Да



Нет

[Обсудить вопрос](#)

Оцените сложность вопроса

Легко



Сложно

#10 Назначением какого паттерна проектирования является предоставление удобного интерфейса к громоздкому и сложному API?

Factory Method

Iterator

Decorator

Strategy

Facade



[Обсудить вопрос](#)

Оцените сложность вопроса

Легко



Сложно



support@mg.proghub.ru