

# ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ



Тест пройден!

georgiilsaev прошел тест с результатом 8 из 10

## Как вам тест?



## РЕЗУЛЬТАТЫ

32

Заработанные очки

junior

Сложность

80%

Правильных ответов

## ОТВЕТЫ

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

Adapter

Bridge

Singleton



Proxy

Decorator

[Обсудить вопрос](#)

**Объяснение:** Паттерн Одиночка (англ. Singleton) — это порождающий паттерн проектирования, гарантирующий, что у класса есть только один экземпляр, и предоставляющий к нему глобальную точку доступа. Данный паттерн решает сразу две проблемы, нарушая принцип единственной ответственности класса:

- Гарантирует наличие единственного экземпляра класса. Чаще всего это полезно для доступа к какому-то общему ресурсу, например, базе данных.
- Предоставляет глобальную точку доступа. Это не просто глобальная переменная, через которую можно достучаться к определённому объекту. Глобальные переменные не защищены от записи, поэтому любой код может подменять их значения без вашего ведома.

Оцените сложность вопроса

Легко



Сложно

#2 Структурный паттерн проектирования, разделяющий один или несколько классов на две отдельные иерархии — абстракцию и реализацию, позволяя изменять их независимо друг от друга. О каком паттерне идёт речь?

декоратор

МОСТ

легковес

нет верного ответа

[Обсудить вопрос](#)

**Объяснение:** Речь идёт о паттерне Мост. Чтобы понять его суть, необходимо определиться с понятиями абстракция и реализация. Абстракция (или интерфейс) — это образный слой управления чем-либо. Он не делает работу самостоятельно, а делегирует её слою реализации (иногда называемому платформой).

Оцените сложность вопроса

Легко



Сложно

#3 Какие группы паттернов существуют по версии GoF?

Organizational

Behavioral

Reusable

Core

Creational

Structural

[Обсудить вопрос](#)

Оцените сложность вопроса

Легко



Сложно

#4 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.

да

нет

[Обсудить вопрос](#)

**Объяснение:** см. паттерны поведения

Оцените сложность вопроса

Легко

Сложно

#5 Какой паттерн проектирования используется для создания семейств зависимых между собой объектов?

Factory Method

Abstract Factory

Prototype

Builder

Dependency Injection

[Обсудить вопрос](#)

Оцените сложность вопроса

Легко

Сложно

#6 При малом количестве объектов лучше использовать паттерн Facade вместо Mediator.

Да

Нет

[Обсудить вопрос](#)

Оцените сложность вопроса

Легко

Сложно

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

Да

Нет

[Обсудить вопрос](#)

Оцените сложность вопроса

Легко

Сложно

#8 Какие основные задачи решает применение паттернов проектирования?

Производится унификация терминологии, названий модулей и элементов проекта.

Показываются отношения и взаимодействия между классами или объектами.

Облегчается дискуссия об абстрактных структурах данных между разработчиками.

Описывается решение целого класса абстрактных проблем.

[Обсудить вопрос](#)

**Объяснение:** 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.

Наш сайт использует куки чтобы обеспечить качественную работу функционала сайта. Продолжая использовать сайт ProgHub, вы соглашаетесь с нашей политикой конфиденциальности.

OK

#9 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.

Оцените сложность вопроса

Легко

Сложно

#10 Поведенческий паттерн проектирования, позволяющий последовательно обходить элементы составных объектов, не раскрывая их внутреннего представления. О каком паттерне идёт речь?

посетитель

посредник

итератор

[Обсудить вопрос](#)

**Объяснение:** Идея паттерна Итератор состоит в том, чтобы вынести поведение обхода коллекции из самой коллекции в отдельный класс. Объект-итератор будет отслеживать состояние обхода, текущую позицию в коллекции и сколько элементов ещё осталось обойти. Одну и ту же коллекцию смогут одновременно обходить различные итераторы, а сама коллекция не будет даже знать об этом.

Оцените сложность вопроса

Легко

Сложно



[support@mg.proghub.ru](mailto:support@mg.proghub.ru)