Паттерны проектирования

Всего 13/14

Тестирование на знание паттернов проектирования

Имя *

Исаев Георгий

Паттерны проектирования

Тестирование на знание паттернов проектирования

✓ Отметьте верные утверждения * 1 из 1

- [x] Абстрактная фабрика основана на композиции, ее задача - создание семейств взаимосвязанных объектов. ✓
- [x] Фабричный метод основан на наследовании, его задача - перемещение создания экземпляров в субклассы. ✓
- [ ] Фабричный метод основан на композиции, его задача - создание семейств взаимосвязанных объектов.
- [ ] Абстрактная фабрика основана на наследовании, ее задача - перемещение создания экземпляров в субклассы.

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

- ( ) Да
- (•) Нет ✓

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

- Adapter
- Decorator
- Proxy
- Singleton ✓

✓ Поведенческий паттерн проектирования, который позволяет передавать запросы последовательно по цепочке обработчиков. Каждый последующий обработчик решает, может ли он обработать запрос сам и стоит ли передавать запрос дальше по цепи. Какой это паттерн? * 1 из 1

- Цепочка обязанностей ✓
- Адаптер
- Стратегия
- Хранитель
- Итератор

✗ Какие основные задачи решает применение паттернов проектирования? * 0 из 1

- [x] Описывается решение целого класса абстрактных проблем. ✓
- [x] Показываются отношения и взаимодействия между классами или объектами. ✓
- [x] Облегчается дискуссия об абстрактных структурах данных между разработчиками. ✓
- [x] Производится унификация терминологии, названий модулей и элементов проекта. ✓
- [x] Используются как основа архитектуры или дизайна приложений ✗

Правильный ответ

- [x] Описывается решение целого класса абстрактных проблем.
- [x] Показываются отношения и взаимодействия между классами или объектами.
- [x] Облегчается дискуссия об абстрактных структурах данных между разработчиками.
- [x] Производится унификация терминологии, названий модулей и элементов проекта.

Комментарий

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.

✓ При малом количестве объектов лучше использовать паттерн Facade вместо Mediator. * 1 из 1

- ( ) Да
- (•) Нет ✓

✓ Какие из перечисленных утверждений верны по отношению к паттерну проектирования Composite: * 1 из 1

- [ ] Composite паттерн содержит в себе несколько моделей поведения объекта и предоставляет возможность переключаться между ними.
- [x] Паттерн Composite позволяет работать с композитными и простыми объектами одинаковым образом. ✓
- [x] Паттерн Composite организовывает объекты в древовидные структуры. ✓
- [ ] Composite паттерн упрощает работу с группой объектов не связанных типов.

✓ Паттерн Strategy имеет следующие преимущества: * 1 из 1

- [ ] Предоставляет разделение между моделью и представлением.
- [x] Отделяет алгоритм от класса, в котором он используется. ✓
- [ ] Гарантирует использование единственной стратегии во время выполнения программы.
- [x] Позволяет переключаться между алгоритмами во время выполнения программы. ✓

✓ Сколько существует разных типов (и соответственно принципиально разных способов реализации) паттерна Adapter? * 1 из 1

- ( ) 1
- (•) 2 ✓
- ( ) 3
- ( ) 4
- ( ) 5

Комментарий

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

✓ Что является преимуществом использования паттернов проектирования? 1 из 1

- ○ Они упрощают разработку и поддержку пользовательских интерфейсов
- ○ Они снижают затраты на разработку, так как они уже реализованы и их можно использовать без изменений
- ○ Они уменьшают количество проектной документации
- ○ Они предоставляют механизмы для тестирования модулей системы
- ◉ Они предоставляют проверенные техники решения задач ✓

✓ Какой паттерн проектирования используется для создания семейств зависимых между собой объектов? * 1 из 1

- ○ Builder
- ◉ Abstract Factory ✓
- ○ Factory Method
- ○ Prototype

✓ Целью какого паттерна является расширение функциональности класса или же ее изменение без использования механизма наследования? * 1 из 1

- ◉ Decorator ✓
- ○ Flyweight
- ○ Proxy
- ○ Composite

✓ Когда следует использовать паттерн "приспособленец"? * 1 из 1

- ■ 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

✓ К какой группе относится паттерн проектирования "Decorator" (в соответствии с GoF)? * 1 из 1

- ○ Порождающие паттерны (creational)
- ○ Паттерны поведения (behavioral)
- ○ Паттерн не относится ни к одной из перечисленных групп
- ◉ Структурные паттерны (structural) ✓