

## Вариант 2

### Первая группа вопросов

1.2) Abstract Factory представляет собой порождающий паттерн проектирования, который предоставляет интерфейс для создания семейств взаимосвязанных или взаимозависимых объектов, не указывая их конкретных классов. Этот паттерн используется для создания объектов без явного указания их классов, что способствует повышению уровня абстракции в системе.

Инкапсулирует:

- Создание семейств связанных объектов.
- Скрытие деталей реализации конкретных классов.

Аспекты изменчивости и сложности:

Изменчивость:

Добавление новых семейств продуктов.

Изменение семейств продуктов.

Сложность:

Увеличивается с увеличением количества абстрактных продуктов и семейств.

Примеры:

Веб-фреймворк, где семейства продуктов могут представлять разные компоненты пользовательского интерфейса (кнопки, текстовые поля и т. д.) для различных тем оформления.

Дополнительные гарантии:

- Гарантирует согласованность создаваемых объектов.
- Обеспечивает простоту замены семейств продуктов.

### Вторая группа вопросов

2.3) Инкапсуляция - это концепция объектно-ориентированного программирования, которая объединяет данные и методы, работающие с этими данными, внутри класса и скрывает детали реализации от внешнего мира. Инкапсуляция позволяет ограничить доступ к внутренним компонентам объекта и предоставить интерфейс для их взаимодействия.

2.6) Истинный полиморфизм:

В истинном полиморфизме объекты одного и того же типа могут вести себя по-разному в зависимости от контекста. Это достигается через использование наследования и переопределения методов.

"Псевдо" полиморфизм:

В широком смысле термин "псевдо" полиморфизм может использоваться для описания любой формы различного поведения в зависимости от контекста, включая использование перегрузки методов или функций.

2.9) Моделирование в информационных технологиях - это процесс создания абстракций или моделей, которые представляют систему, её компоненты, процессы или взаимодействия. Моделирование используется для анализа, проектирования и документирования систем.

Примеры:

- Диаграммы UML для визуализации структуры и поведения системы.
- Модели данных для проектирования баз данных.

2.1) Процедурная декомпозиция - это методология разработки программ, при которой программа разбивается на набор подпрограмм (процедур), каждая из которых выполняет конкретную задачу.

Примеры:

Разделение программы на функции для обработки ввода, вывода, и выполнения конкретных операций.

Использование функций для реализации отдельных шагов в алгоритме.

Третья группа вопросов

3.5) Unit тестирование - это процесс тестирования отдельных частей программы для обеспечения их корректности. Целью является проверка того, что каждая часть программы работает правильно в изоляции.

3 варианта использования Unit Test-ов:

Проверка корректности:

Гарантия, что отдельные части программы ведут себя правильно.

Обнаружение ошибок:

Выявление ошибок и дефектов в коде на ранних этапах разработки.

Обеспечение стабильности:

Предотвращение регрессий, удостоверение, что изменения в коде не влияют на уже работающую функциональность.

3.7) DataProvider в тестировании - это механизм, позволяющий передавать различные наборы входных данных в тестовый метод для выполнения множества тестов с разными данными.

Примеры:

В JUnit, аннотация @DataProvider может использоваться для предоставления данных для тестового метода.

3.10) Generic в Java - это механизм, который позволяет создавать классы, интерфейсы и методы, которые могут работать с различными типами данных. Позволяет программистам создавать обобщенный код, повышая его повторное использование и безопасность.

Использование в контексте Mock frameworks:

Generic типы могут использоваться в mock frameworks для создания заглушек, которые могут работать с различными типами данных. Например, использование generic методов для создания заглушек, которые могут заменять различные реализации интерфейсов.

3.2) Цикломатическая сложность - это числовая мера сложности программы, выраженная в количестве линейно независимых путей через её код. Чем выше цикломатическая сложность, тем сложнее поддерживать и тестировать код.