Огородник Артем

Группа №6

Вариант 2

1.2) Это порождающий шаблон проектирования, который предоставляет интерфейс для создания семейства взаимосвязанных или зависимых объектов без указания их конкретных классов .

Аспекты инкапсуляции:

Инкапсулирует процесс создания семейств объектов, скрывая конкретные классы продуктов и фабрик от клиента.

Аспекты изменчивости и сложности:

Облегчает замену семейства объектов новыми версиями, так как клиентский код зависит только от абстрактных интерфейсов.

Добавление новых семейств продуктов или изменение существующих требует изменения только фабрик, не затрагивая клиентский код.

Дополнительные гарантии:

Обеспечивает согласованность создаваемых объектов внутри одного семейства, что предотвращает нежелательные совместные использования объектов из разных семейств.

Позволяет легко заменять или добавлять новые фабрики и продукты, не затрагивая клиентский код.

2.3) Это один из основных принципов объектно-ориентированного программирования. Он заключается в том, что данные и методы , работающие с этими данными, объединяются в одном классе и скрыты от прямого доступа извне класса. Только методы называемые методами доступа или геттерами и сеттерами предоставляют доступ к этим данным.

Пример :

public class Person {

private String name;

public String getName() {

return name; // Геттер для доступа к имени

}

}

Здесь геттер getName дает доступ к данным строки name.

2.6) Это один из важных принципов объектно-ориентированного программирования. Он позволяет объектам разных классов обрабатываться как объекты одного и того же базового класса. Полиморфизм делится на два типа: истинный и псевдополиморфизм.

Истинный полиморфизм:

1. Связан с динамическим связыванием (вызов методов определяется во время выполнения).
2. Основан на наследовании и переопределении методов (полиморфных методов) в подклассах.

Псевдополиморфизм :

1. Связан с перегрузкой функций и статическим связыванием.
2. Позволяет одной функции иметь несколько версий с разными параметрами.
3. При вызове функции выбирается соответствующая версия на основе типов переданных аргументов.

2.9) Это процесс создания упрощенных, абстрактных представлений реальных объектов, систем или процессов с целью анализа, понимания, оптимизации и принятия решений.

Примеры моделирования в IT:

Моделирование баз данных :

Проектирование и создание схемы баз данных с таблицами, связями и атрибутами.

Моделирование объектов :

Применение объектно-ориентированного моделирования для создания классов и объектов, а также определения их взаимодействия.

3.5) Это процесс проверки отдельных модулей программы (например, функций, классов, методов) для подтверждения их корректной работы. Целью unit-тестов является проверка поведения отдельных компонентов кода для убеждения в их правильной работы в изоляции от других частей программы.

Три варианта использования unit-тестов:

1. Проверка функциональности
2. Поддержание качества кода и рефакторинг
3. Документация и спецификация

4.2) Файл с кодом лежит на Github в репозитории Java