**1.** Термин **"SOLID"** представляет собой акроним для построения гибкой и адаптивной программы.

Solid принципы:

1)S (Принцип единственной обязанности (ответственности)): Каждый компонент (в качестве компонента может выступать класс, структура, метод) должен иметь одну и только одну причину для изменения (т.е выполнять одну единственную задачу). Весь функционал компонента должен быть целостным, обладать высокой связностью.

2)O (Принцип открытости/закрытости): если понадобилось добавить новую функциональность к классу, то существующий класс не модифицируем, а создаем наследника класса с новыми возможностями. То есть у нас должна быть возможность расширять класс без изменения самого класса.

3)L (Принцип подстановки Лисков): дочерний класс должен следовать принципам родительского класса и не изменять их.

4)I (Принцип разделения интерфейсов): Создавайте узкоспециализированные интерфейсы и не вынуждайте клиента зависеть от неиспользуемых интерфейсов.

5)D (Принцип инверсии зависимостей): Сущности должны зависеть от абстракций, а не от чего-то конкретного.

**3.** Паттерн "Абстрактная фабрика" (Abstract Factory) предоставляет интерфейс для создания семейств взаимосвязанных объектов с определенными интерфейсами без указания конкретных типов данных объектов.

**4.** **Паттерн Фабричный Метод определяет интерфейс создания объекта, но позволяет субклассам выбрать класс создаваемого экземпляра.**

Паттерн "Абстрактная фабрика" (Abstract Factory) предоставляет интерфейс для создания семейств взаимосвязанных объектов с определенными интерфейсами без указания конкретных типов данных объектов.

Abstract Factory производит группы продуктов, а Factory Method производит одиночный конкретный тип продукта.

Ассоциация - это отношение, при котором объекты одного типа неким образом связаны с объектами другого типа. Например, объект одного типа содержит или использует объект другого типа.