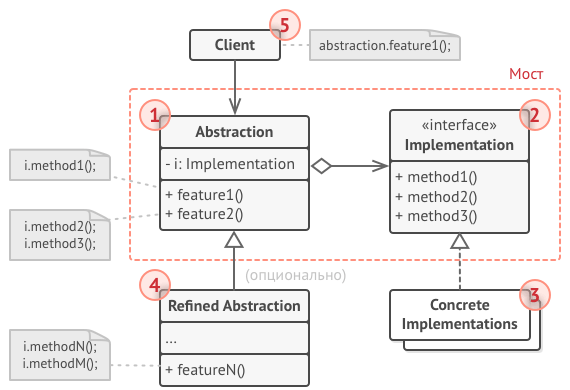
**МОСТ** - разделяет один или несколько классов на две отдельные иерархии — абстракцию и реализацию, позволяя изменять их независимо друг от друга.

ПРИМЕНЕНИЕ:

- когда нужно разделить монолитный класс, который содержит несколько различных реализаций какой-то функциональности. Например, если класс может работать с разными системами баз данных

- когда класс нужно расширять в 2 независимых плоскостях

- когда вы хотите, чтобы реализацию можно было бы изменять во время выполнения программы

РЕАЛИЗАЦИЯ:

**1. АБСТРАКЦИЯ**: содержит управляющую логику. Код абстракции делегирует реальную работу связанному объекту реализации.

**2. РЕАЛИЗАЦИЯ**: задаёт общий интерфейс для всех реализаций. Все методы, которые здесь описаны, будут доступны из класса абстракции и его подклассов. Интерфейсы АБСТРАКЦИИ и РЕАЛИЗАЦИИ могут как совпадать, так и быть совершенно разными, но обычно в реализации живут базовые операции, на которых строятся сложные операции абстракции.

**3. КОНКРЕТНЫЕ РЕАЛИЗАЦИИ**: содержат платформо-зависимый код.

**4. РАСШИРЕННЫЕ АБСТРАКЦИИ**: содержат различные вариации управляющей логики. Как и родитель, работает с реализациями только через общий интерфейс реализации.

**5. КЛИЕНТ**: работает только с объектами абстракции. Не считая начального связывания абстракции с одной из реализаций, клиентский код не имеет прямого доступа к объектам реализации.

ПЛЮСЫ:

- позволяет строить платформо-независимые программы

- скрывает лишние или опасные детали реализации от клиентского кода

- реализует принцип открытости/закрытости

МИНУСЫ:

- усложняет код программы из-за введения дополнительных классов