**Паттерны проектирования**

**ИСиТ,ПОИБМС-2**

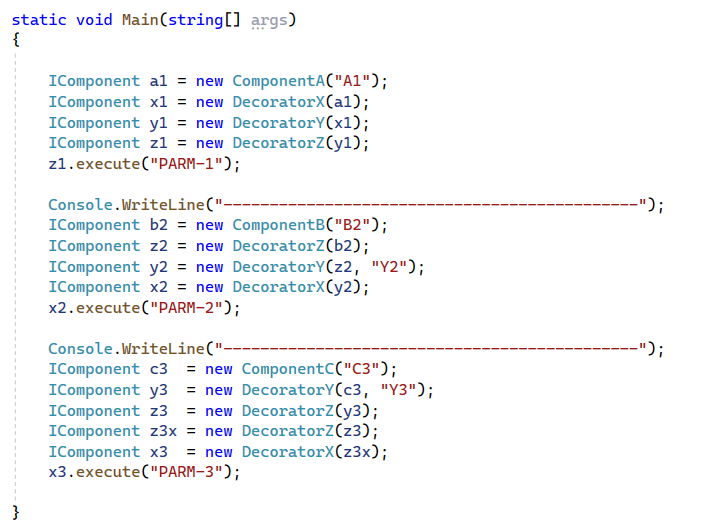
**Лекция 7**

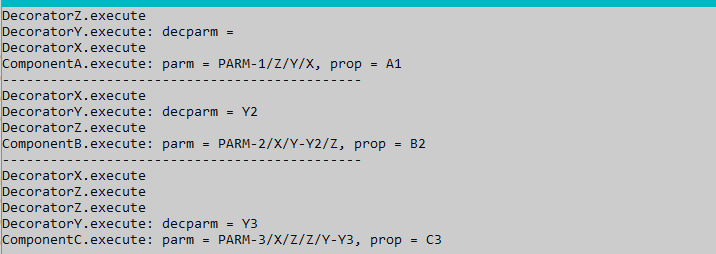
**Структурные паттерны**

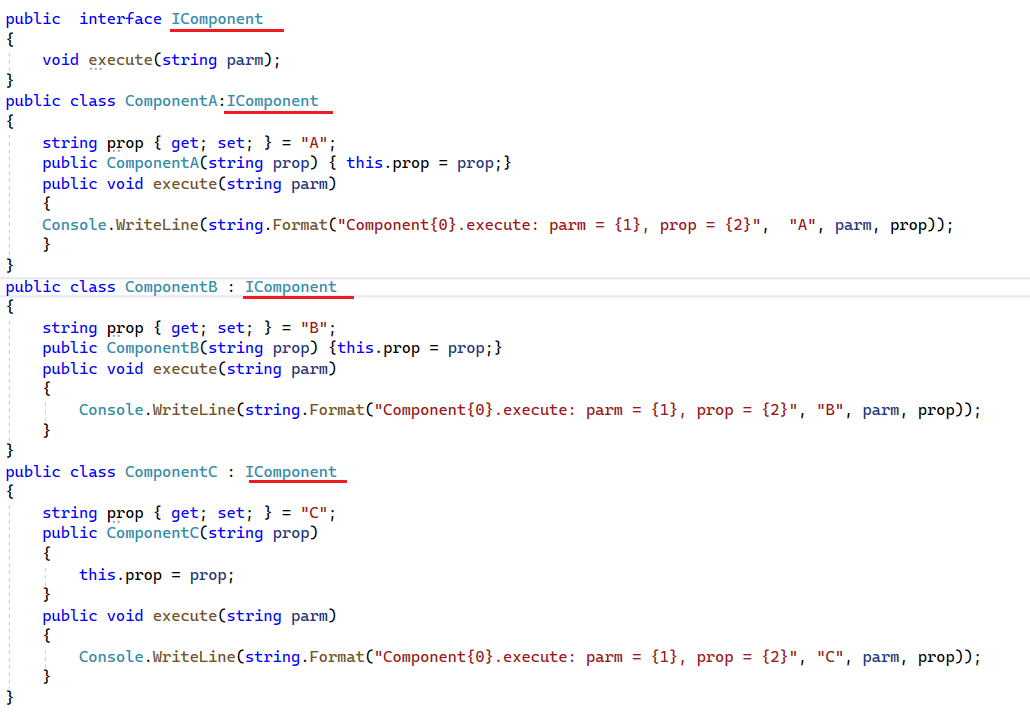
1. **Структурные паттерны:** построение иерархии классов.
2. **Структурные паттерны:** Adapter (Адаптер), Bridge (Мост), Composite (Компоновщик), Decorator (Декоратор), Facade (Фасад), Flyweight (легковес), Proxy (Заместитель).

**Decorator**

1. **Decorator:** динамическоеоборачивание объектов.
2. **Decorator:** обертки заданы, оборачивание динамическое, обычно обертки образуют стек.
3. **C#:**

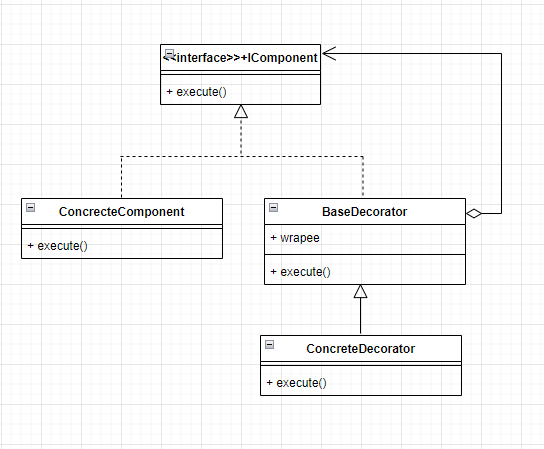
****

****

****

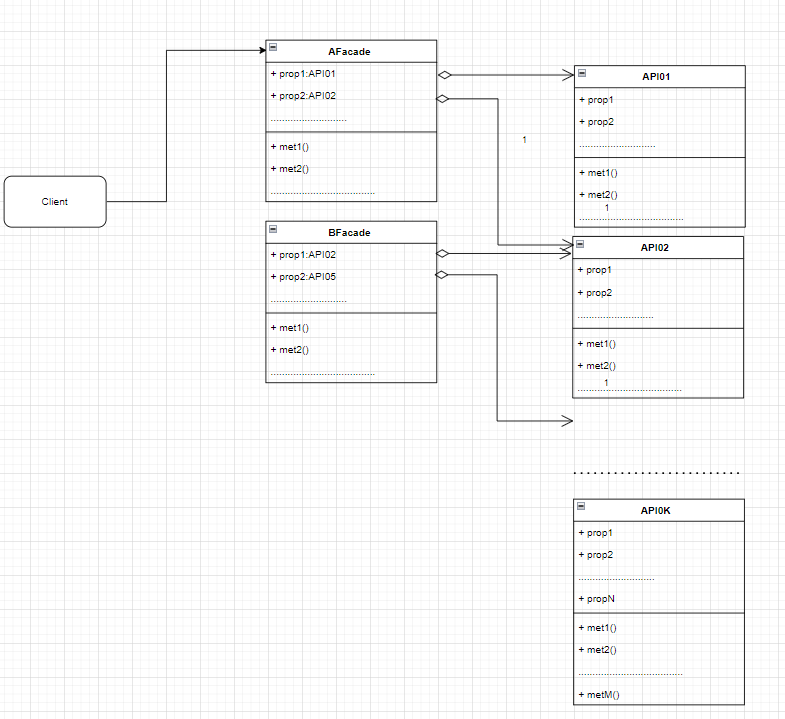
****

1. **UML:**

****

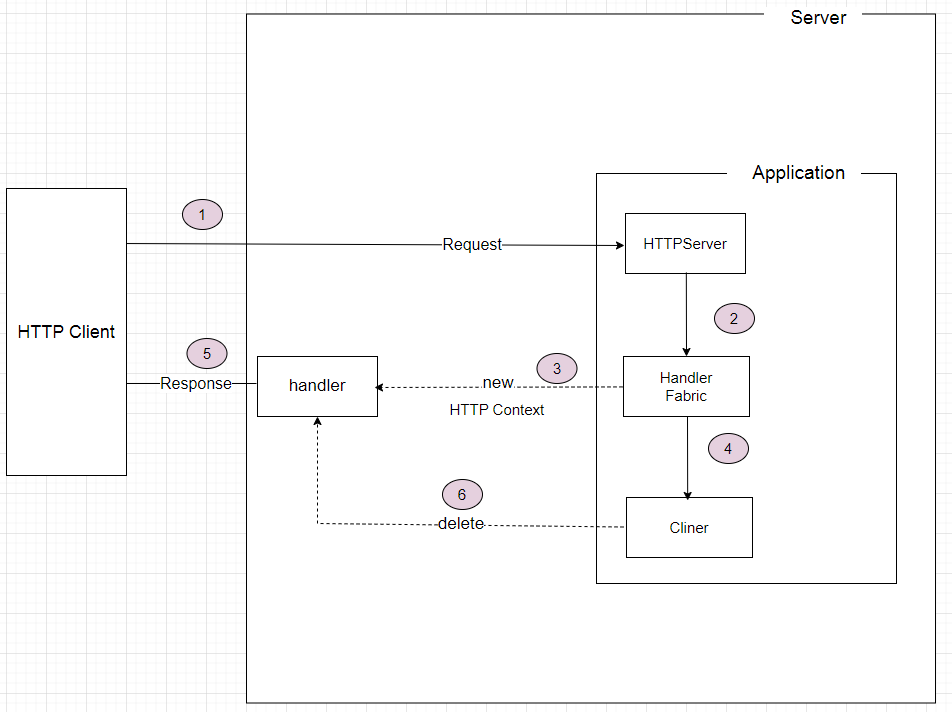
**Facade**

1. **Facade:** простой интерфейс к сложной системе классов (библиотеке, фреймворку)
2. **Facade:** сложный интерфейс библиотеки (много типов, методов, параметров), фасад – надстройка уменьшающая сложность.
3. **Facade:** репозиторий базы данных.

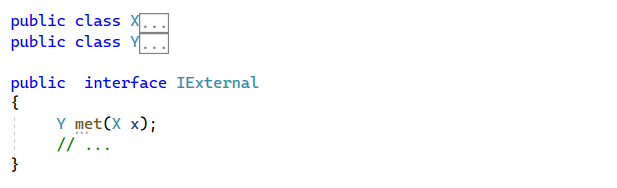
****

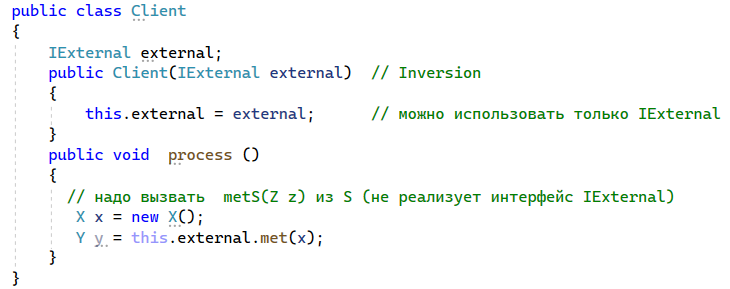
**Flyweight**

1. **Flyweight:** объединение общих данных нескольких исходных классов в общем классе, а в исходных классах остаются данные отражающее текущее состояние (контекст); общий класс обычно называется резидентным (постоянно располагается в памяти), а объекты других классов создаются по мере необходимости с помощью фабрики при этом контекст передается через параметры конструктора.
2. **Flyweight:** примерWeb-сервер

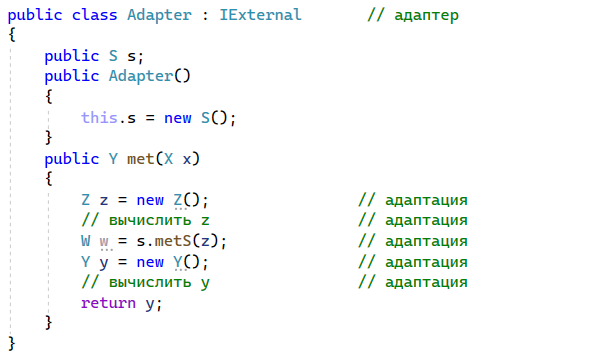
****

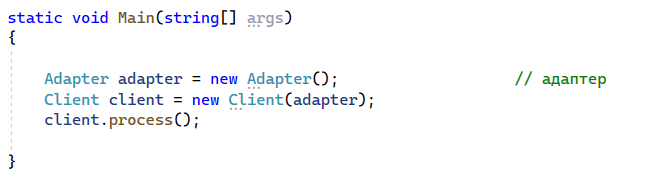
1. **Adapter:** сочленение объектов с несовместимыми интерфейсами.
2. **Схема проблемы**

****

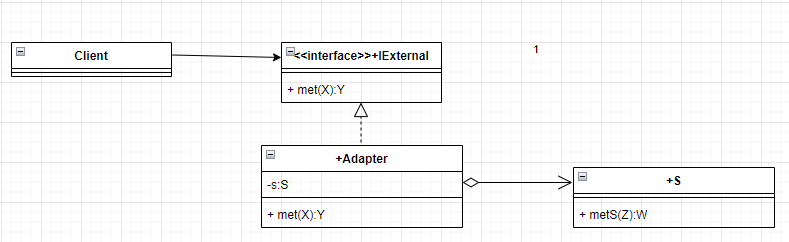
****

1. **Adapter:**

****

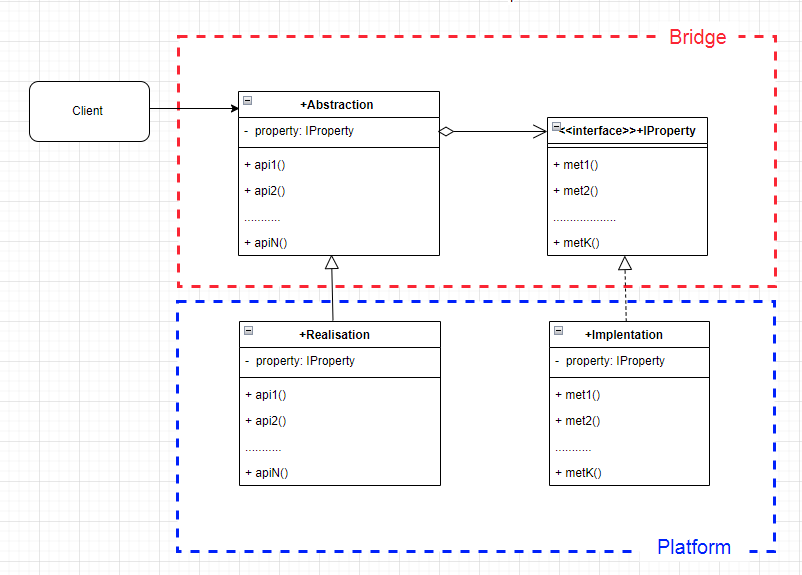
****

1. **UML**

****

**Bridge**

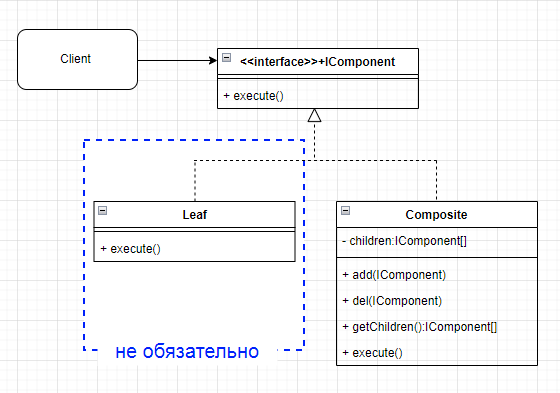
1. **Bridge:** отделить абстракцию и реализацию.
2. **Bridge:** отделить API от платформы
3. **UML:**

****

1. **Bridge:** пример: QT, общий С++-интерфейс, работает на разных платформах (Windows, Unix, Linux).

**Composite**

1. **Composite:** построение древовидной иерархии объектов, для вычислений имеющих древовидную структуру.
2. **UML:**



1. **Composite:** пример: MRP-система (Material Requirements Planning), модель изделия (продукта) имеет иерархическую модель; модель используется для планирования складских запасов комплектующих изделия, вычисления трудоемкости изделия, вычисления себестоимости изделия, …
2. **КОНЕЦ**
3. **Decorator**
4. **Decorator:** динамическоеоборачивание объектов.
5. **Decorator:** обертки заданы, оборачивание динамические, обычно обертки образуют стек (можно очередь).