

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

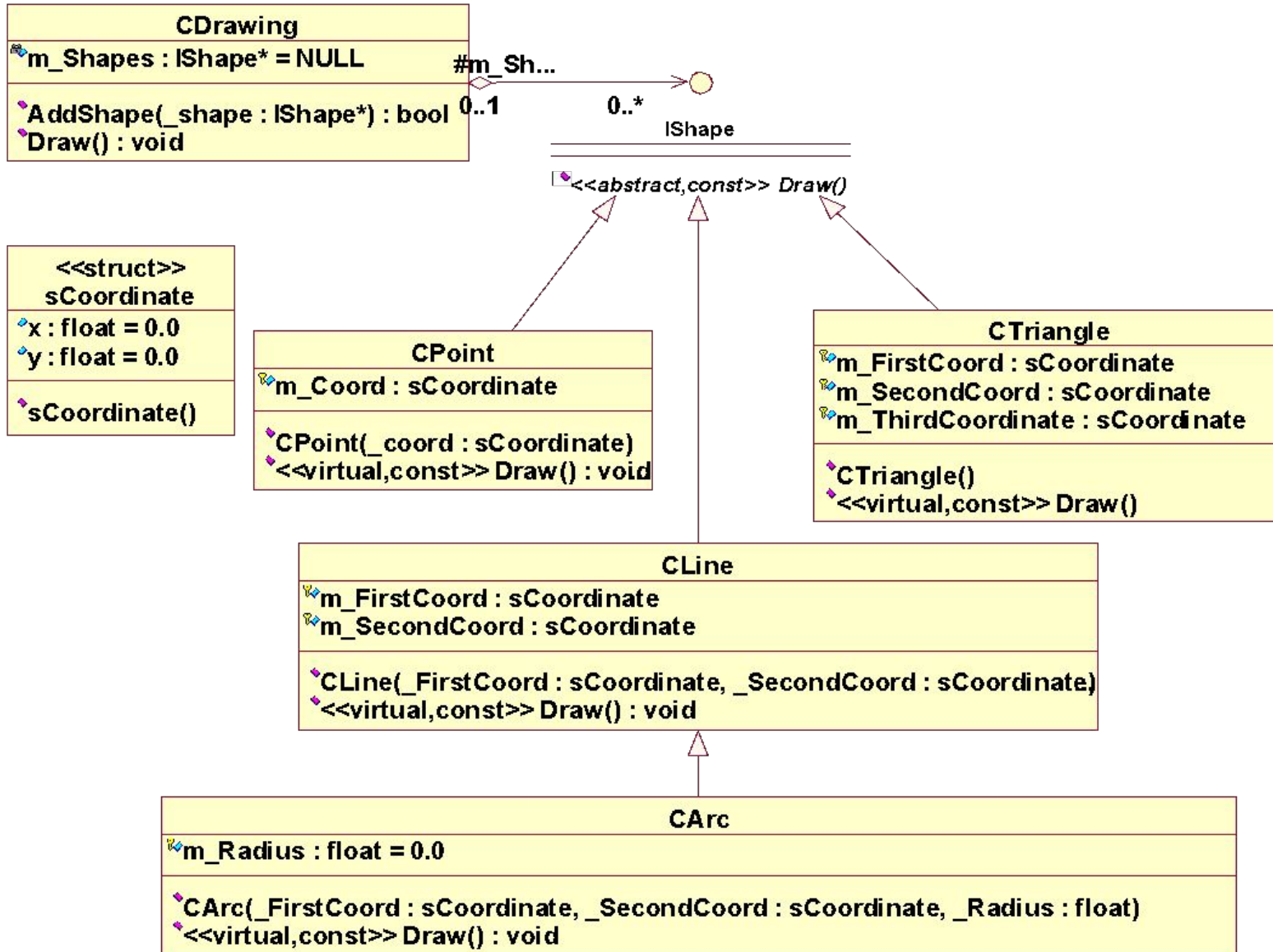
# ОСНОВЫ РАЗРАБОТКИ ПРОГРАММНЫХ СИСТЕМ

## **Типовые решения стадии проектирования**

Смирнов Иван  
Соченков Илья



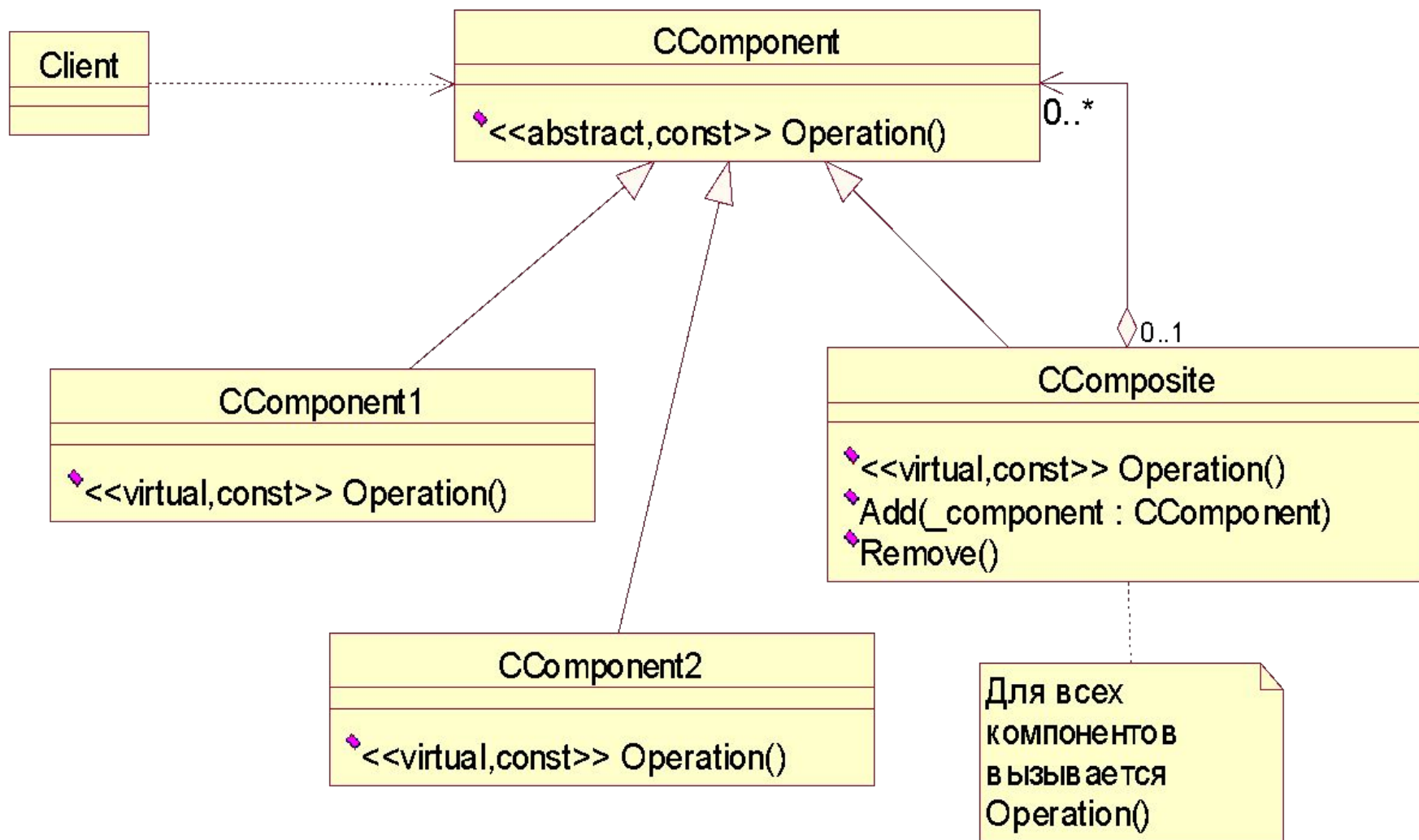
# Рисунки

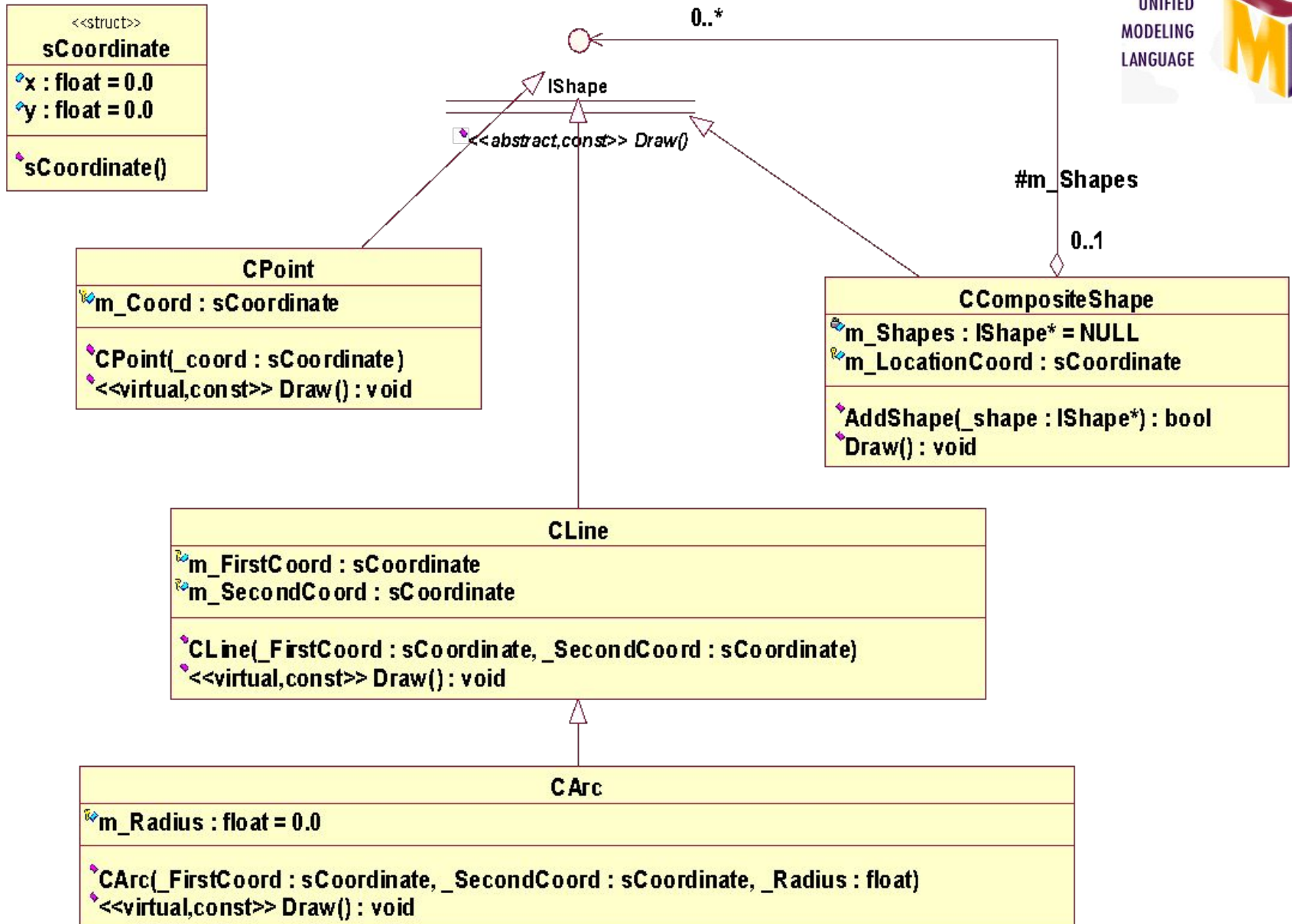


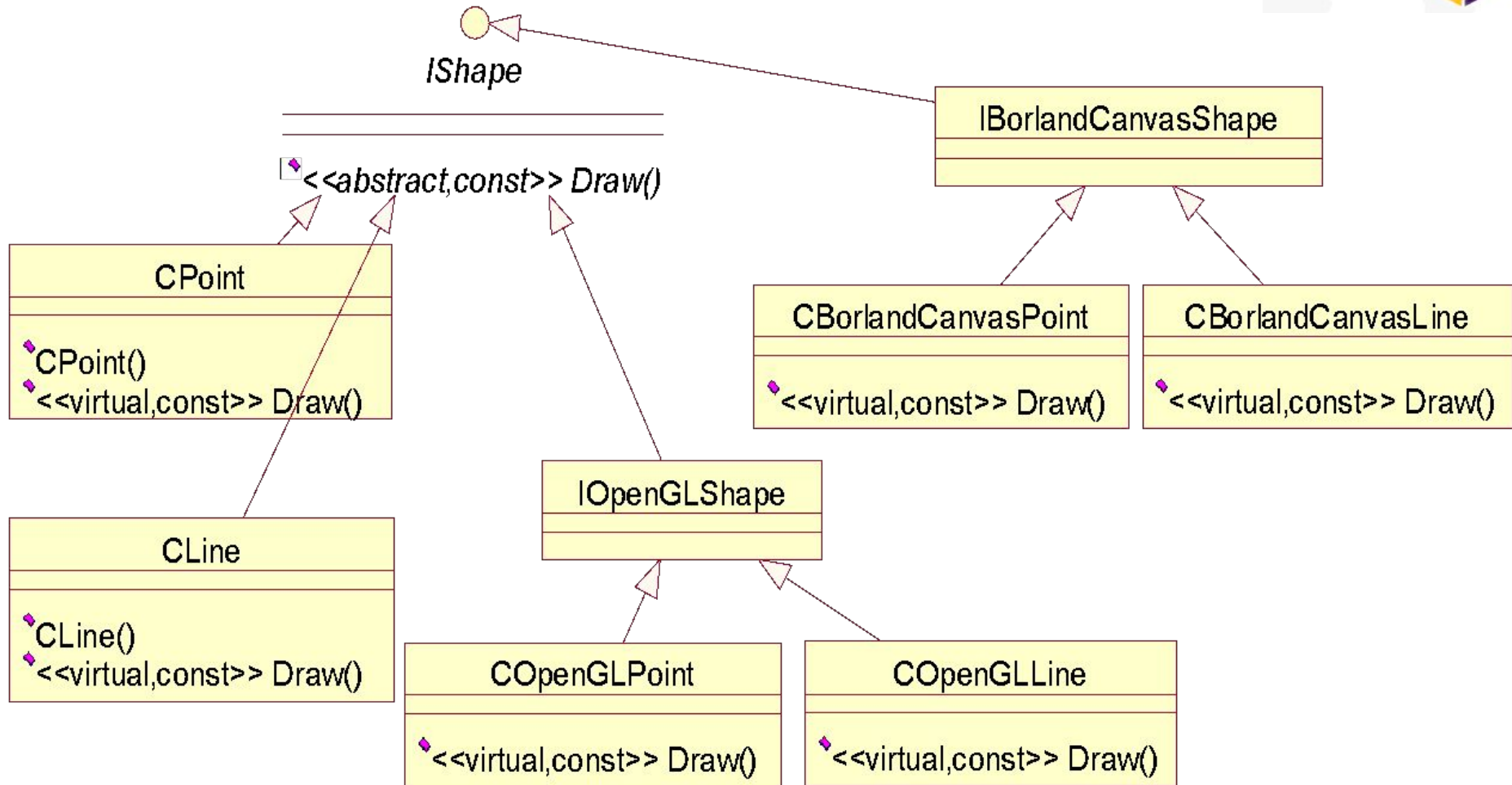
# Компоновщик (Composite)

- Рекурсивная композиция
- Единый интерфейс для простых и сложных компонентов

# Компоновщик: Структура





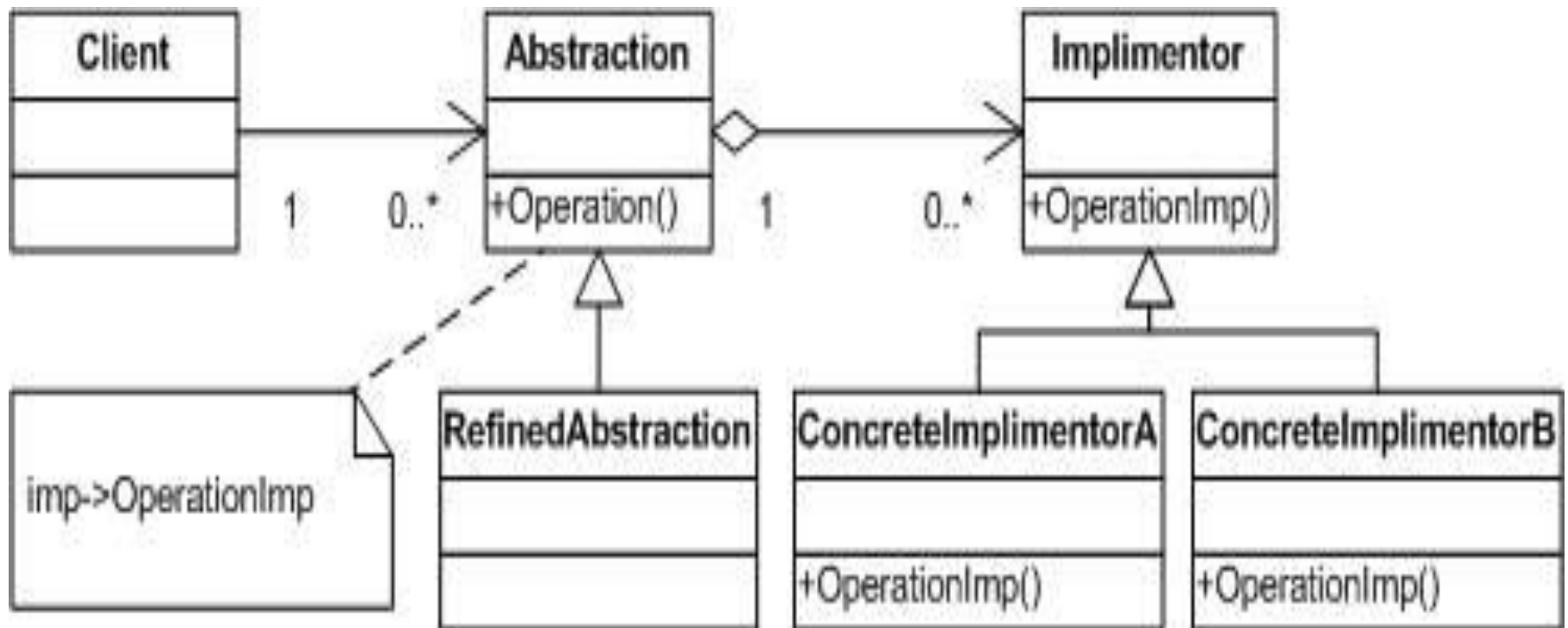


# Мост (bridge)

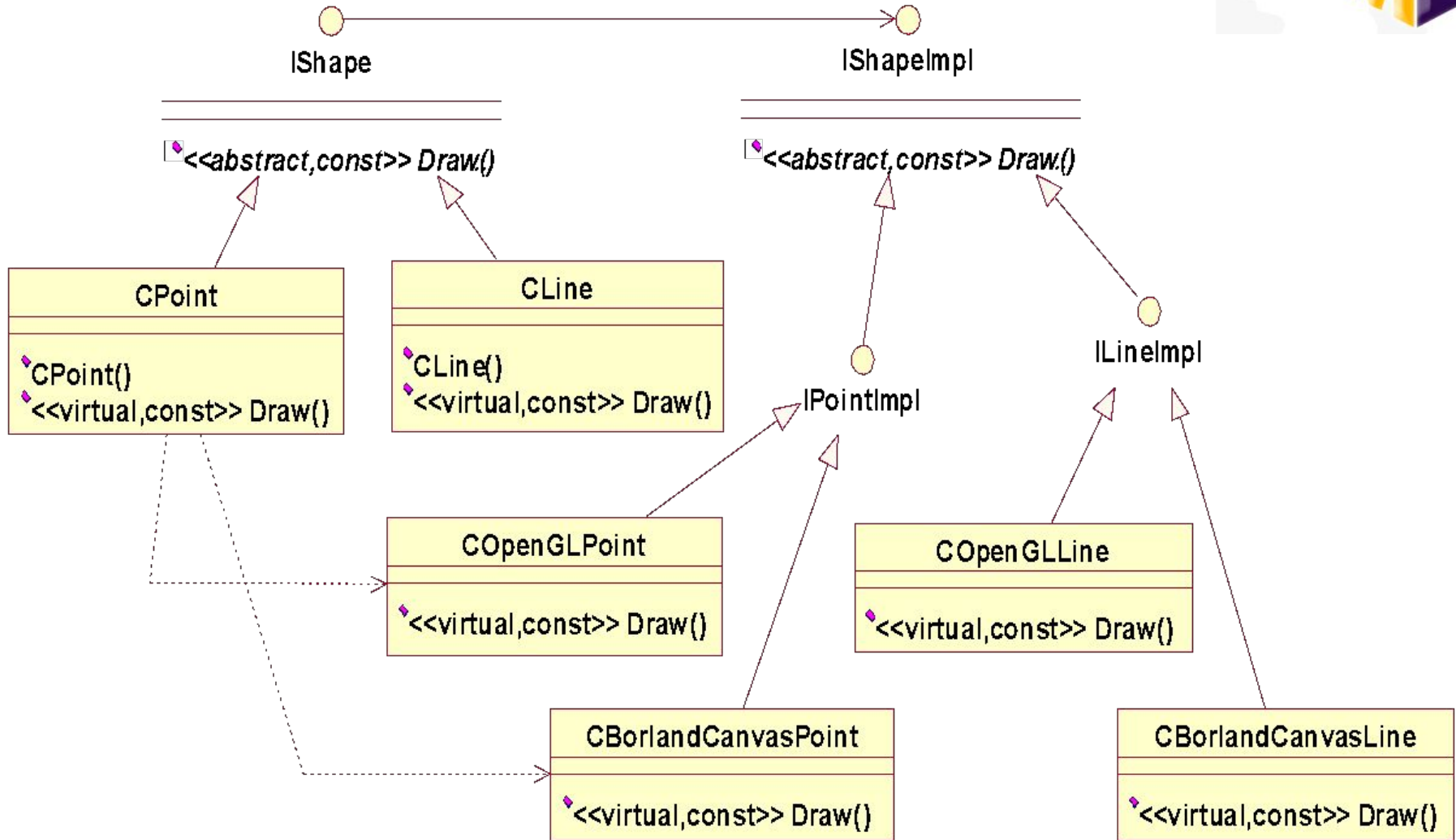
- Проблемы
  - Несколько реализаций порождает много классов
  - Привязка реализации к абстракции
  - Трудно менять реализацию, не затронув абстракцию

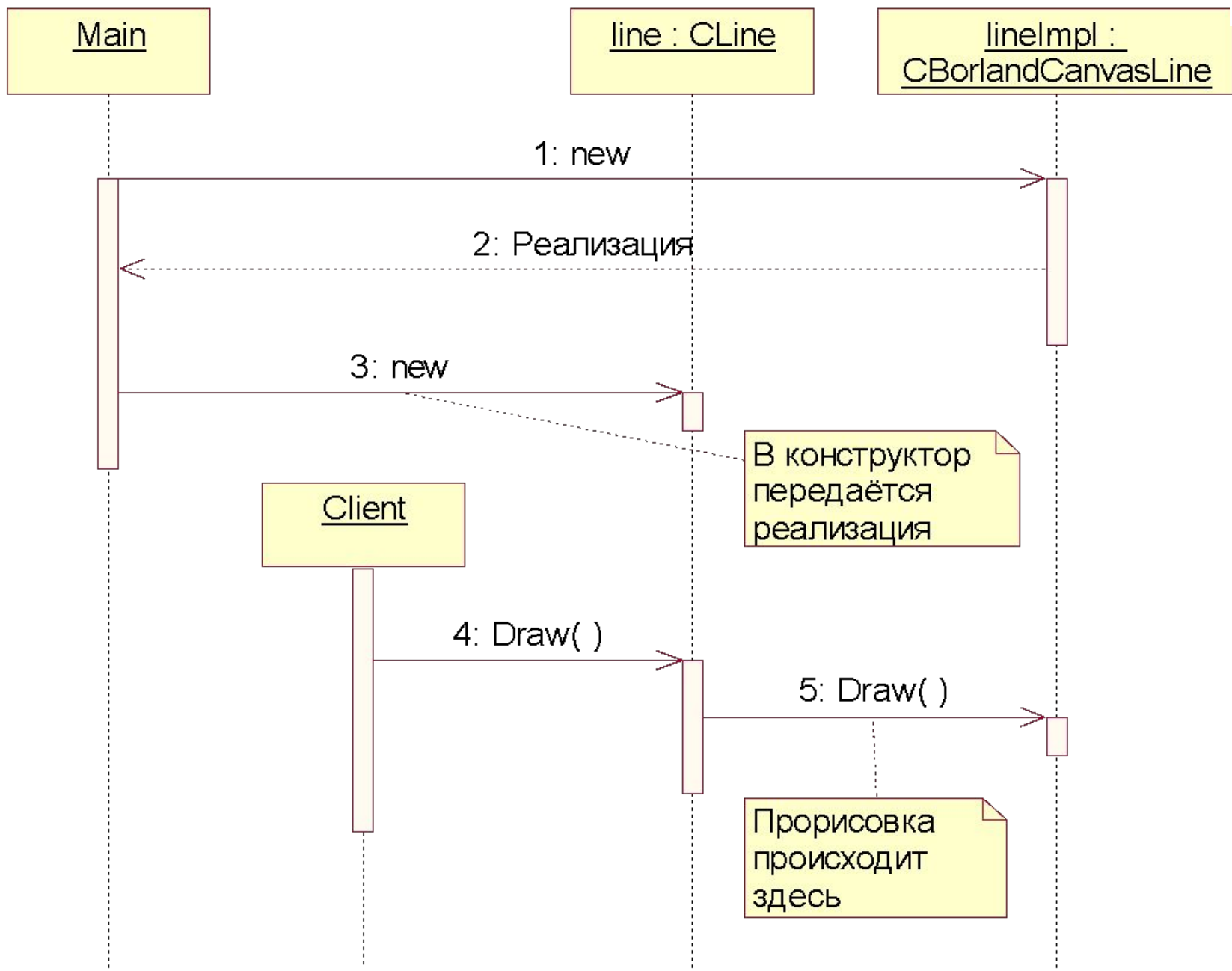
# Мост: решение

- Отделяем абстракцию от реализаций









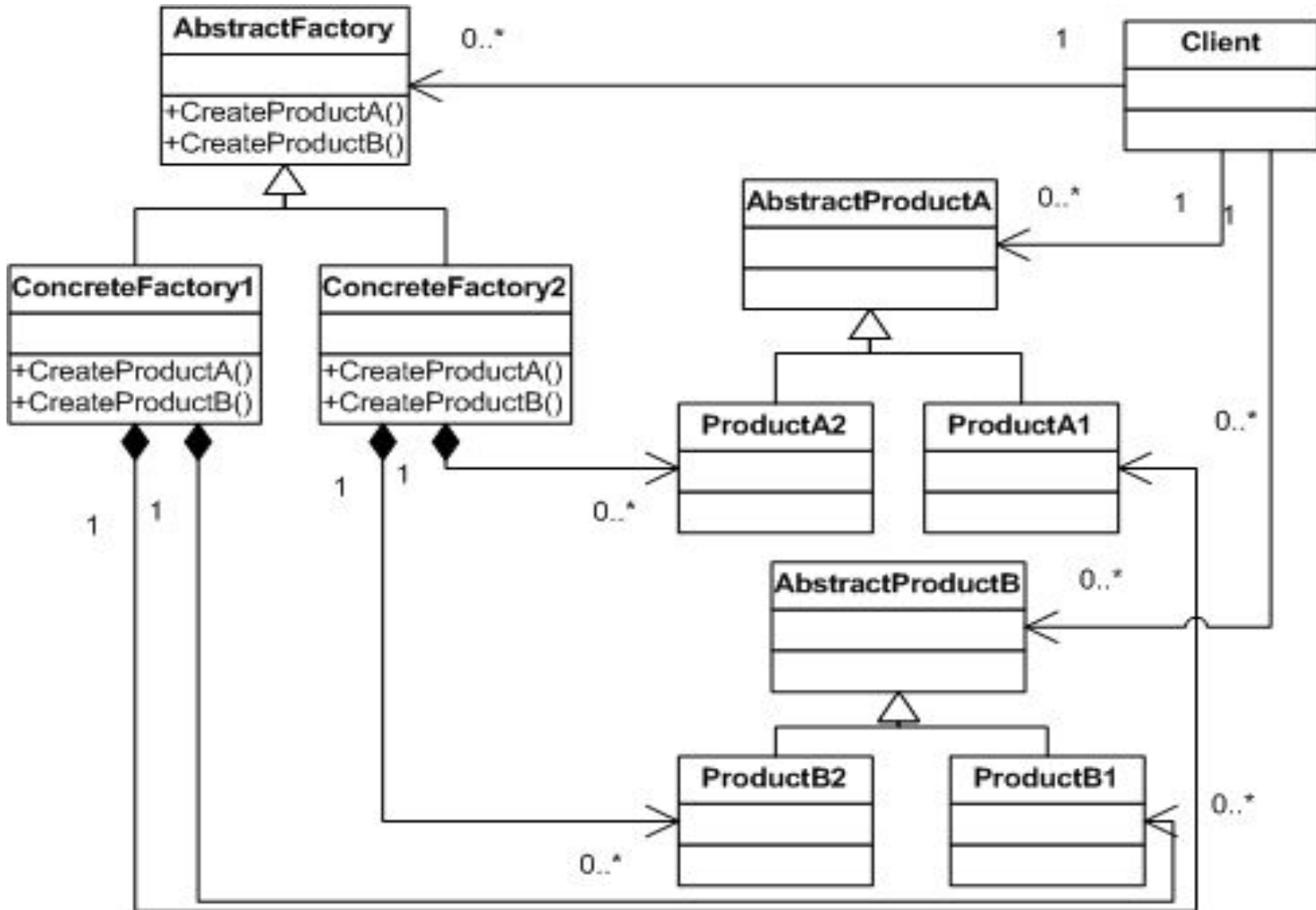
# Фабрика (Factory): проблема

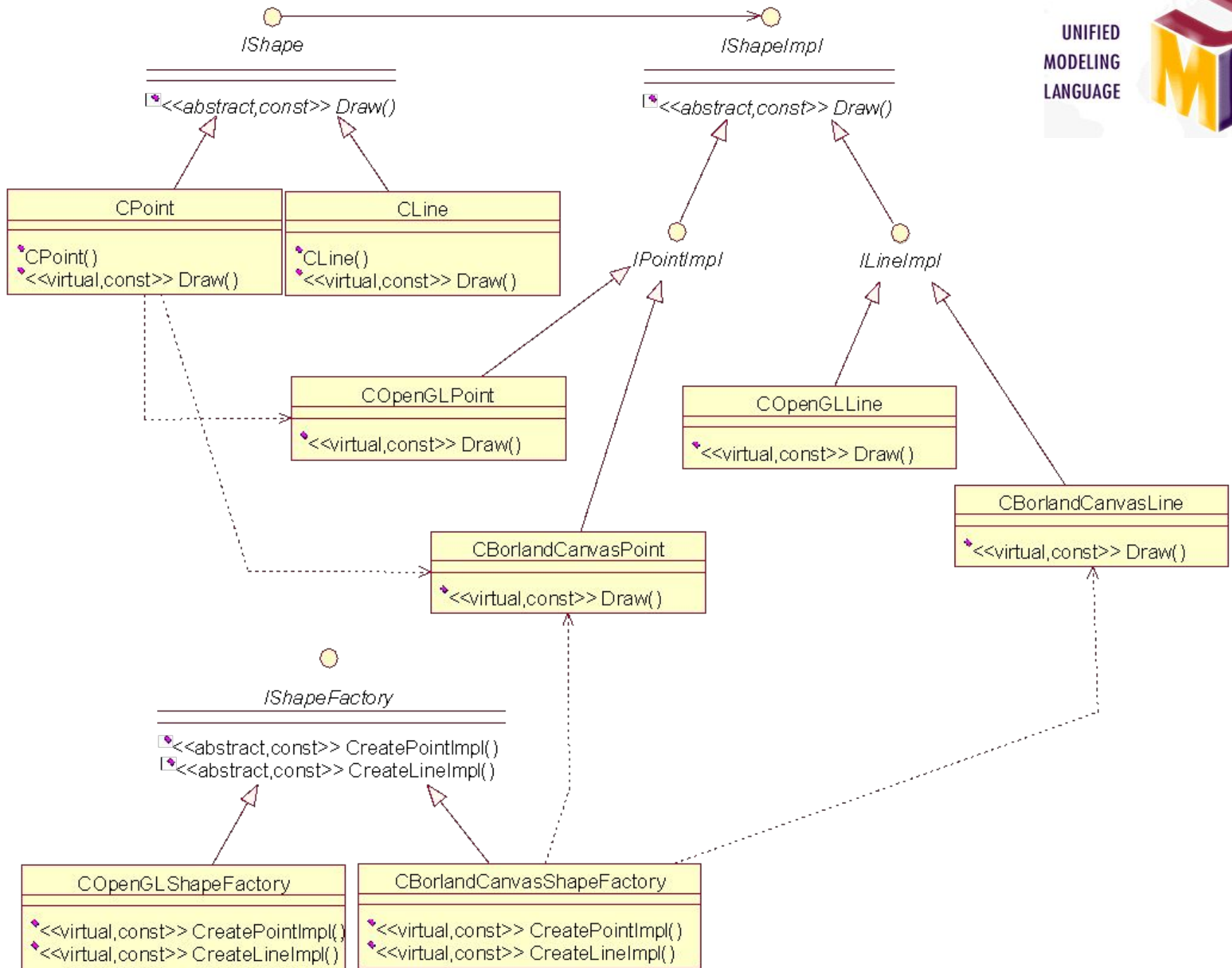
- Существует несколько семейств объектов
- В одно время необходимо создавать объекты только одного семейства, т.е. система конфигурируется этим семейством
- Система не должна зависеть от выбора семейства

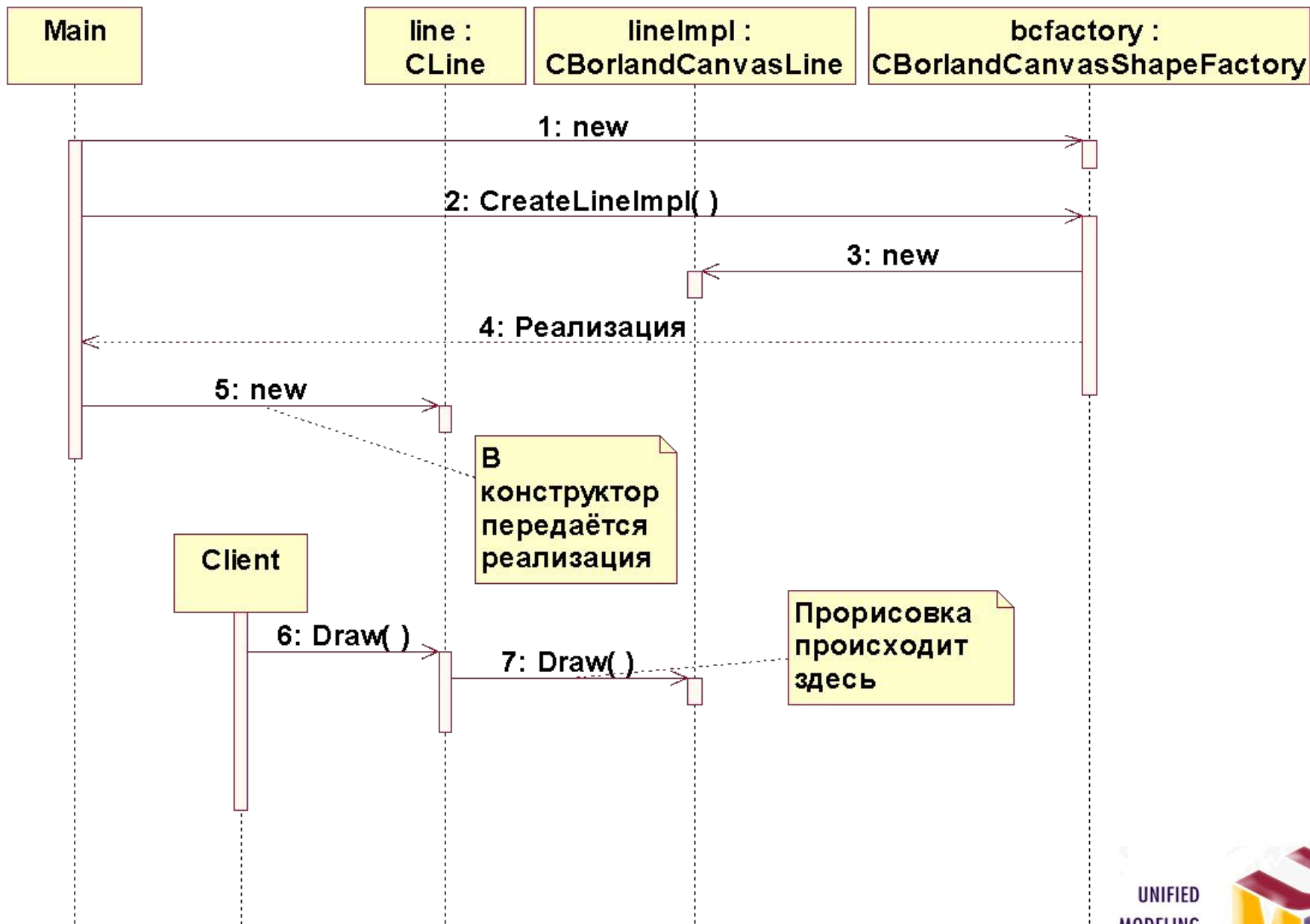
# Фабрика: решение

- Для каждого семейства объектов создаём классы-фабрики, создающие объекты

# Фабрика: структура





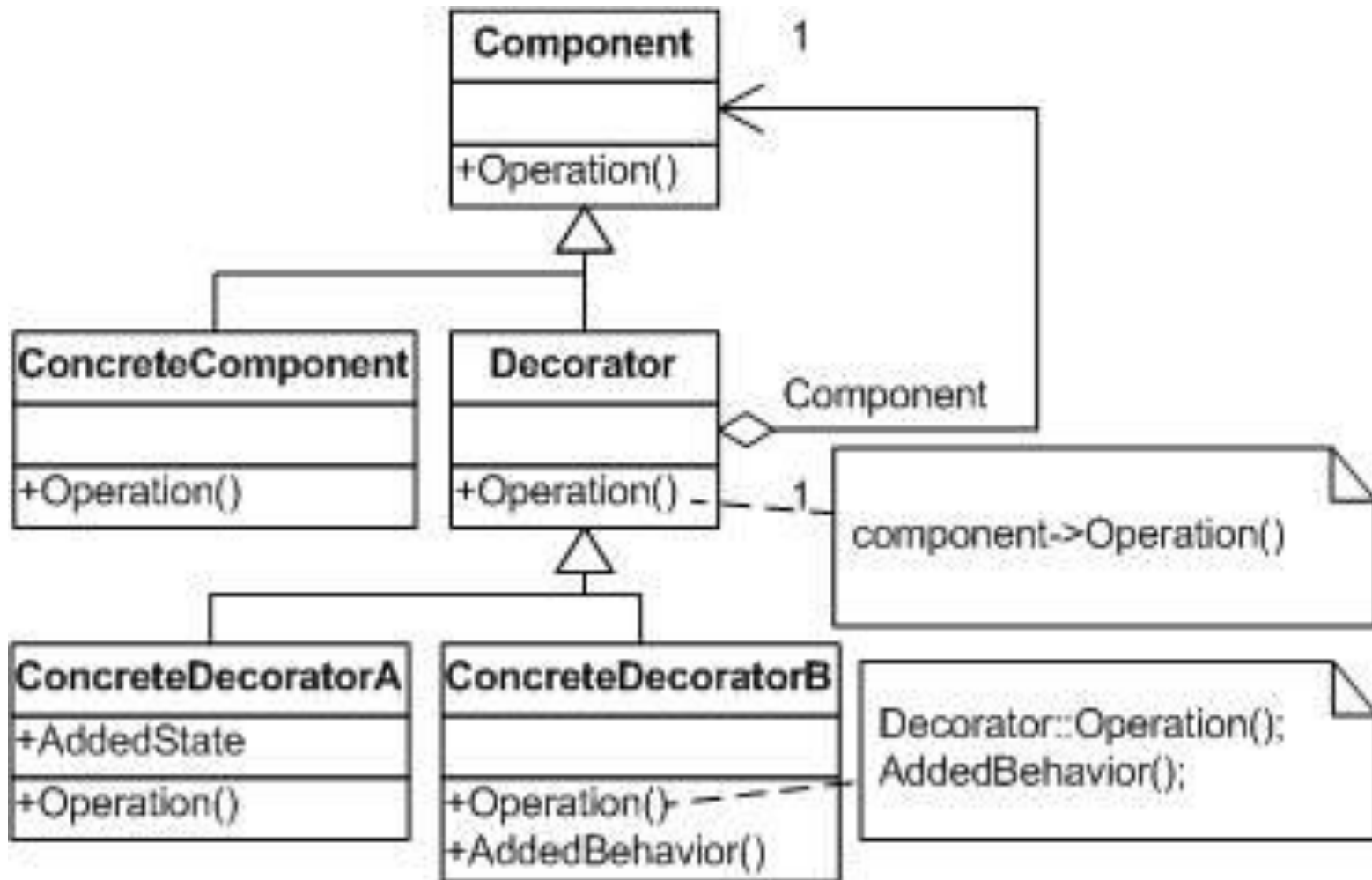


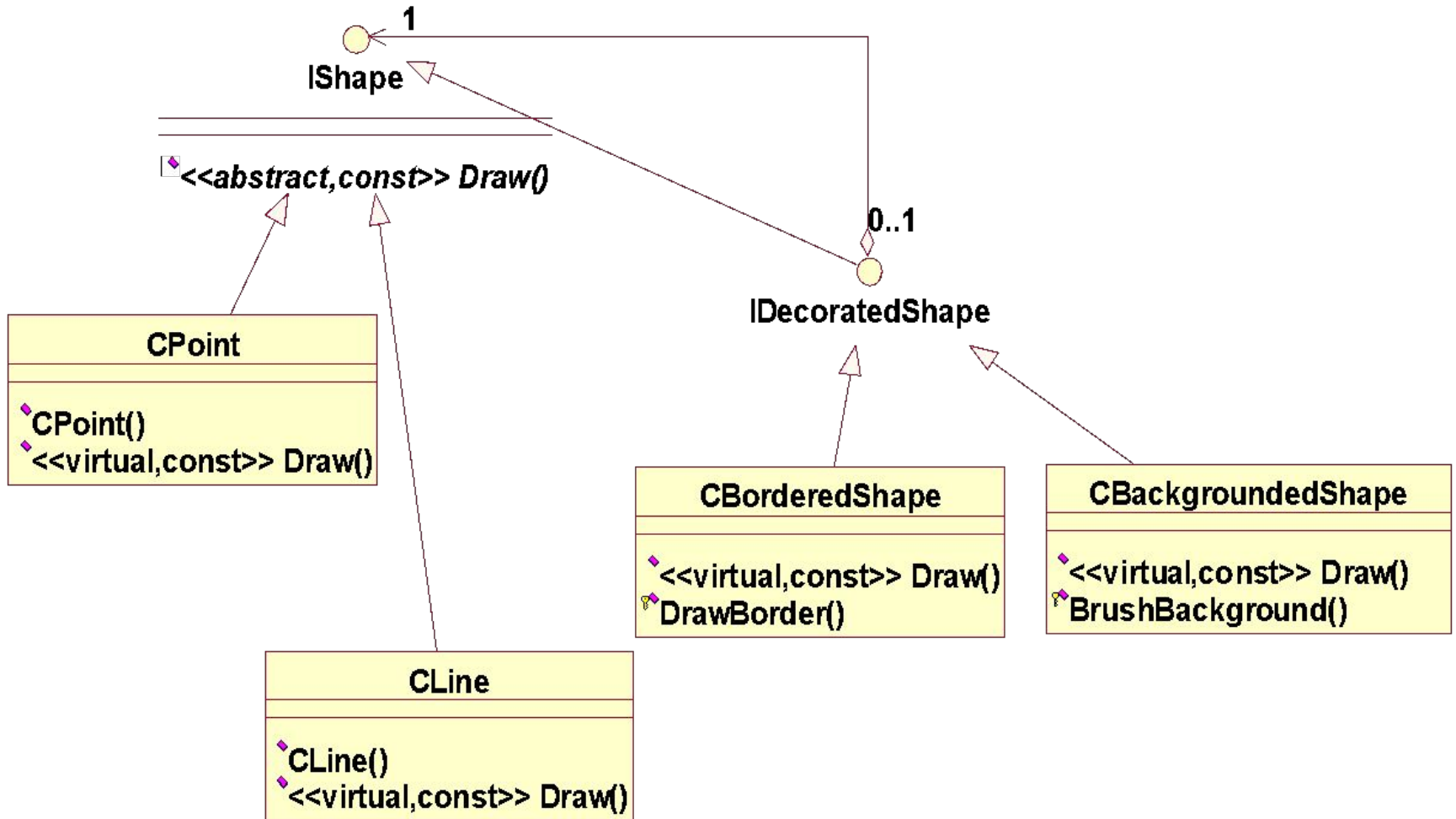
# Декоратор (Decorator)

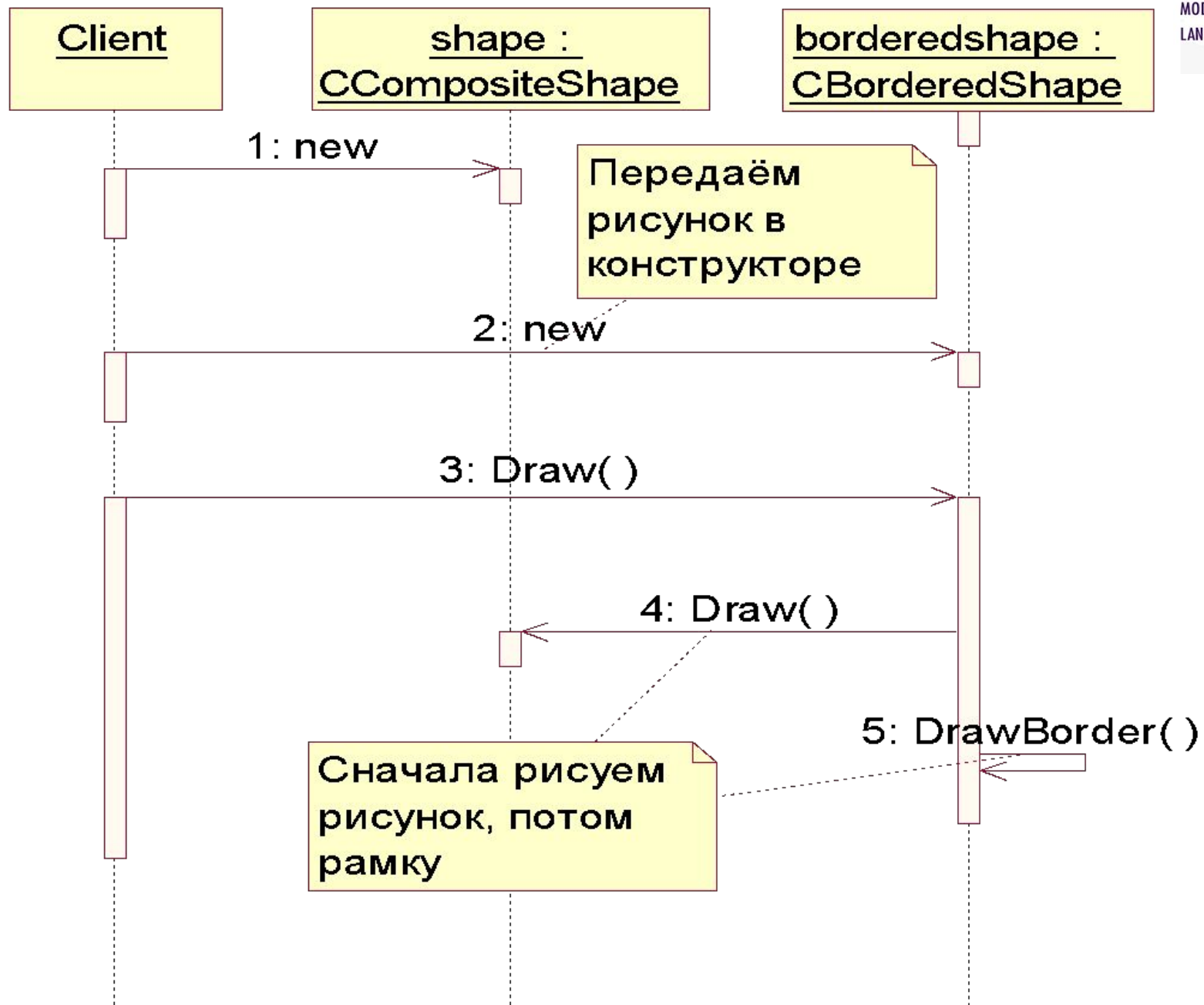
- Проблема
  - Необходимо добавить обязанности одному объекту, но не классу
- Решение
  - Создаём класс-декоратор, который наследует интерфейсу декорируемого объекта



# Декоратор: структура







# Заключение

- Рассмотрели решения, позволяющие создавать масштабируемые программы
  - Компоновщик – организует вложенные классы в иерархию с *единым интерфейсом*
  - Мост – отделяет абстракцию от её реализации, делает систему гибче
  - Фабрика – создаёт объекты нужного семейства
  - Декоратор – легко расширяет поведение объекта