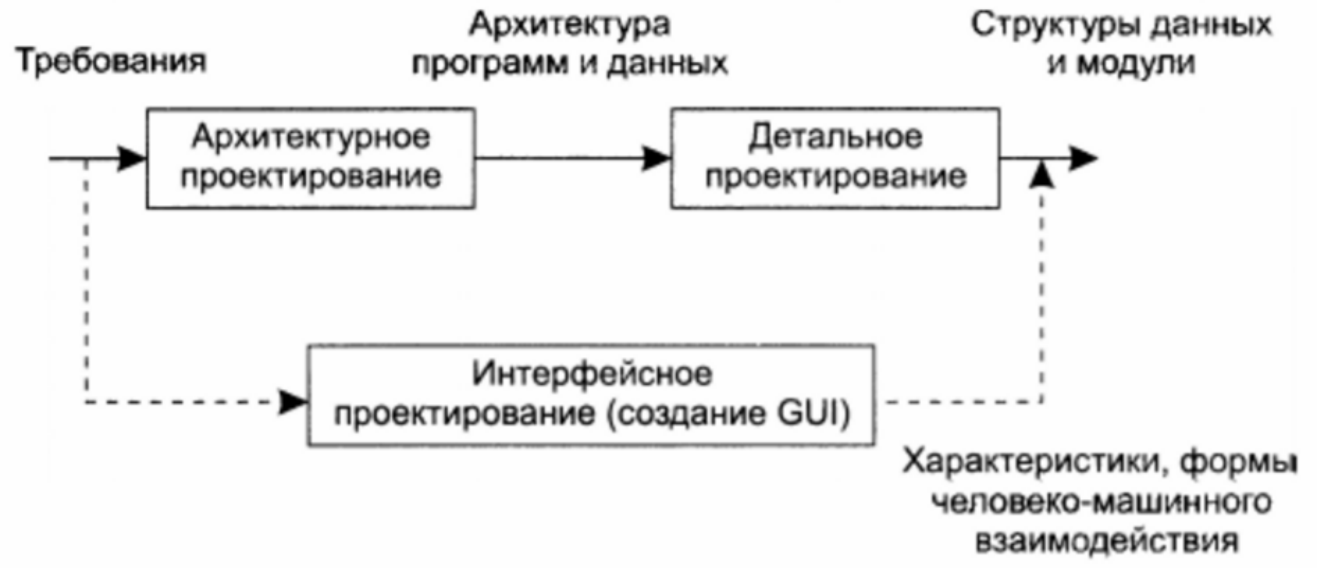
Проектирование ПО

Юрий Литвинов
yurii.litvinov@gmail.com

27.03.2017г

Проектирование ПО

Проектирование — деятельность, предшествующая разработке



Проектирование ПО

- Определение базового архитектурного стиля и структуры приложения
- Определение функционального поведения
- Осуществление декомпозиции на модули и планирование их взаимодействия
- Выбор стратегии и деталей реализации
  - Используется ли кодогенерация?
  - Используются ли сторонние фреймворки?
  - Используются ли сторонние библиотеки?
  - Какая часть функциональности будет реализовывать “вручную”?
- Вопросы распределённости/децентрализованности приложения
- Вопросы безопасности и прочие нефункциональные требования
- Вопросы локализации
- Вопросы размещения
- Вопросы обновления
- ...

Инструменты проектирования

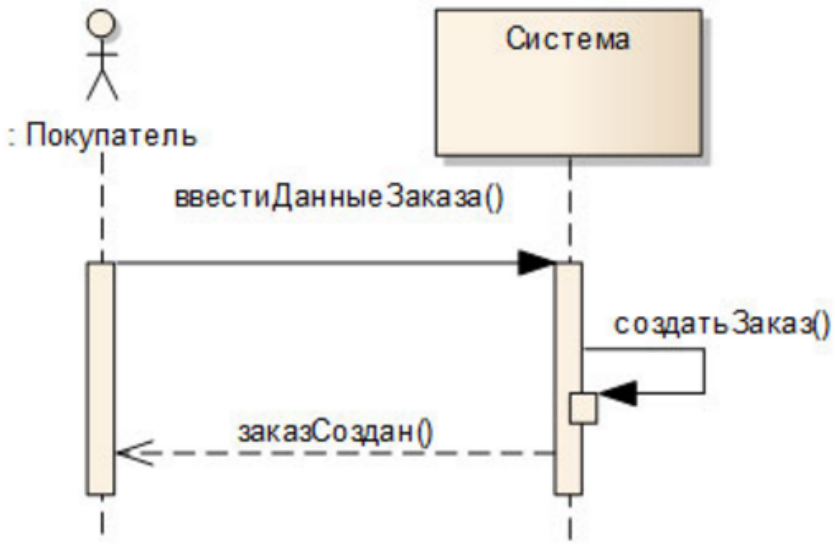
- Моделирование
  - Визуальное
  - Вербальное
  - Формальное
- Прототипирование
- Архитектурное описание
  - Неформальное
  - Формальное

Моделирование

- **Модель** — упрощённое подобие объектов или явлений
  - Позволяет изучать некоторые их свойства
- Свойства моделей
  - Сжатие информации
  - Целенаправленность
  - Субъективность
  - Ограниченность
  - Способность к расширению

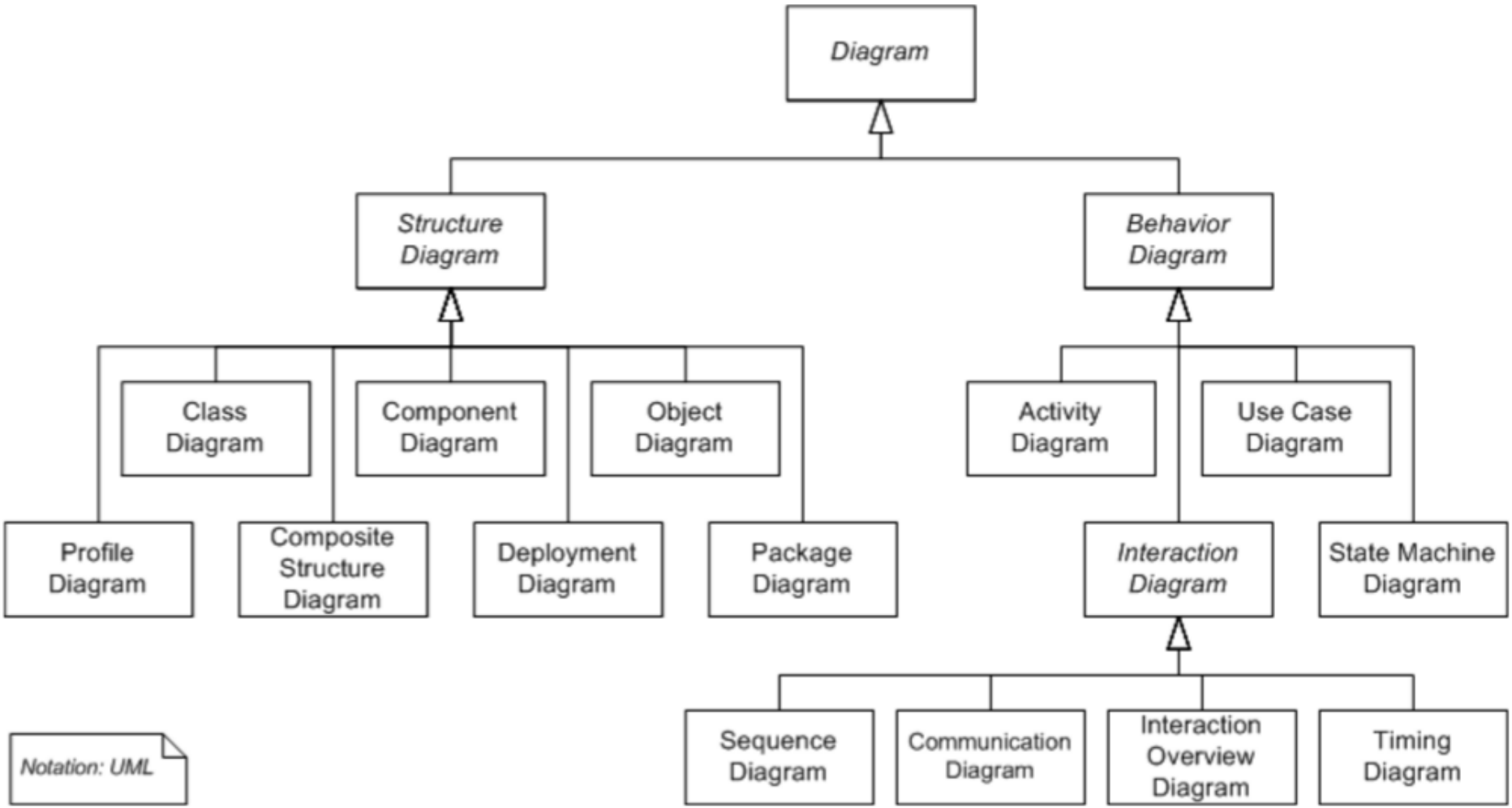
Визуальные модели

- Различный состав моделей и степень детальности
- Метафора визуализации
- Точка зрения моделирования
- Назначение
  - Одноразовые модели
  - Документация
  - Графические исходники

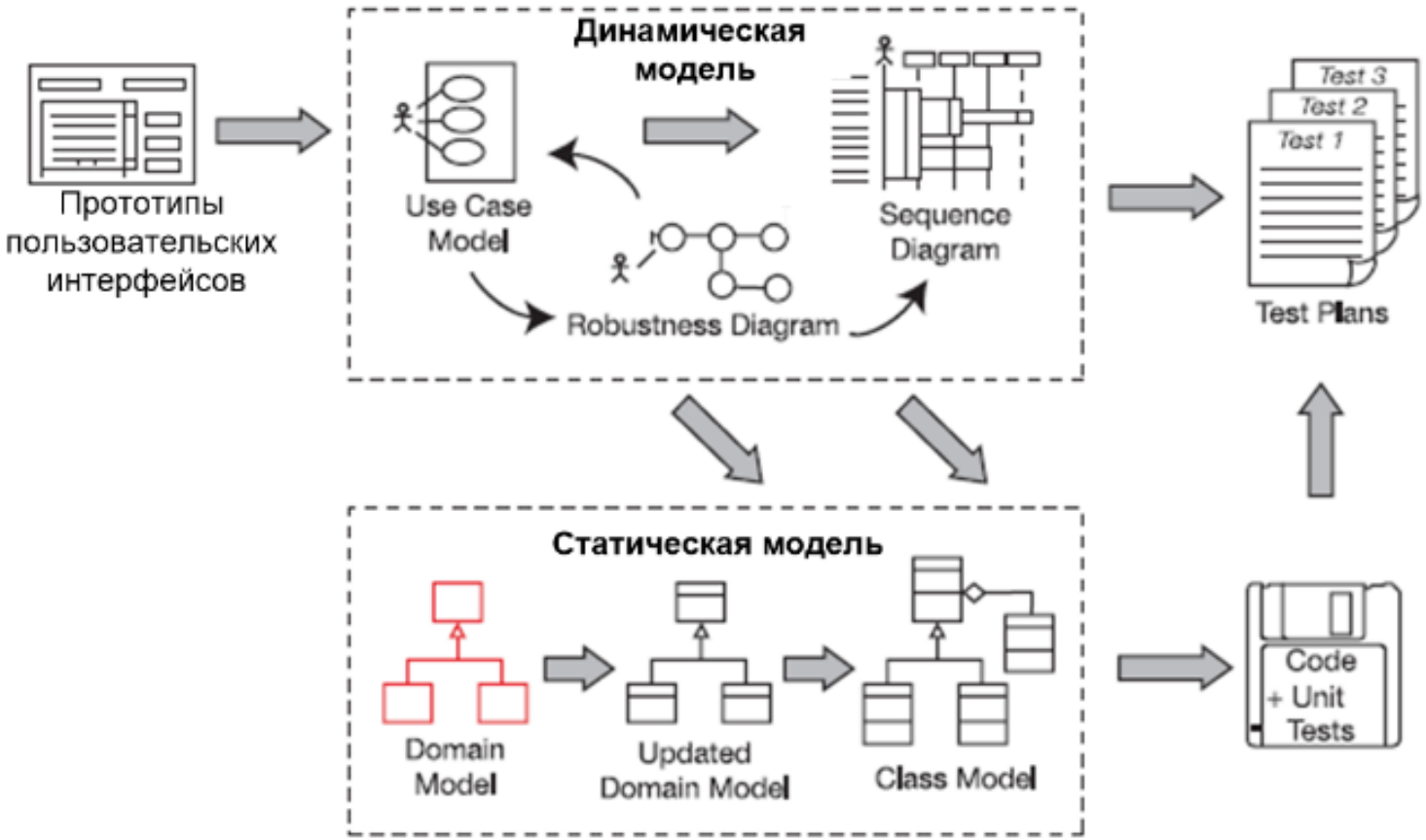


Язык UML

Unified Modeling Language



Представления системы



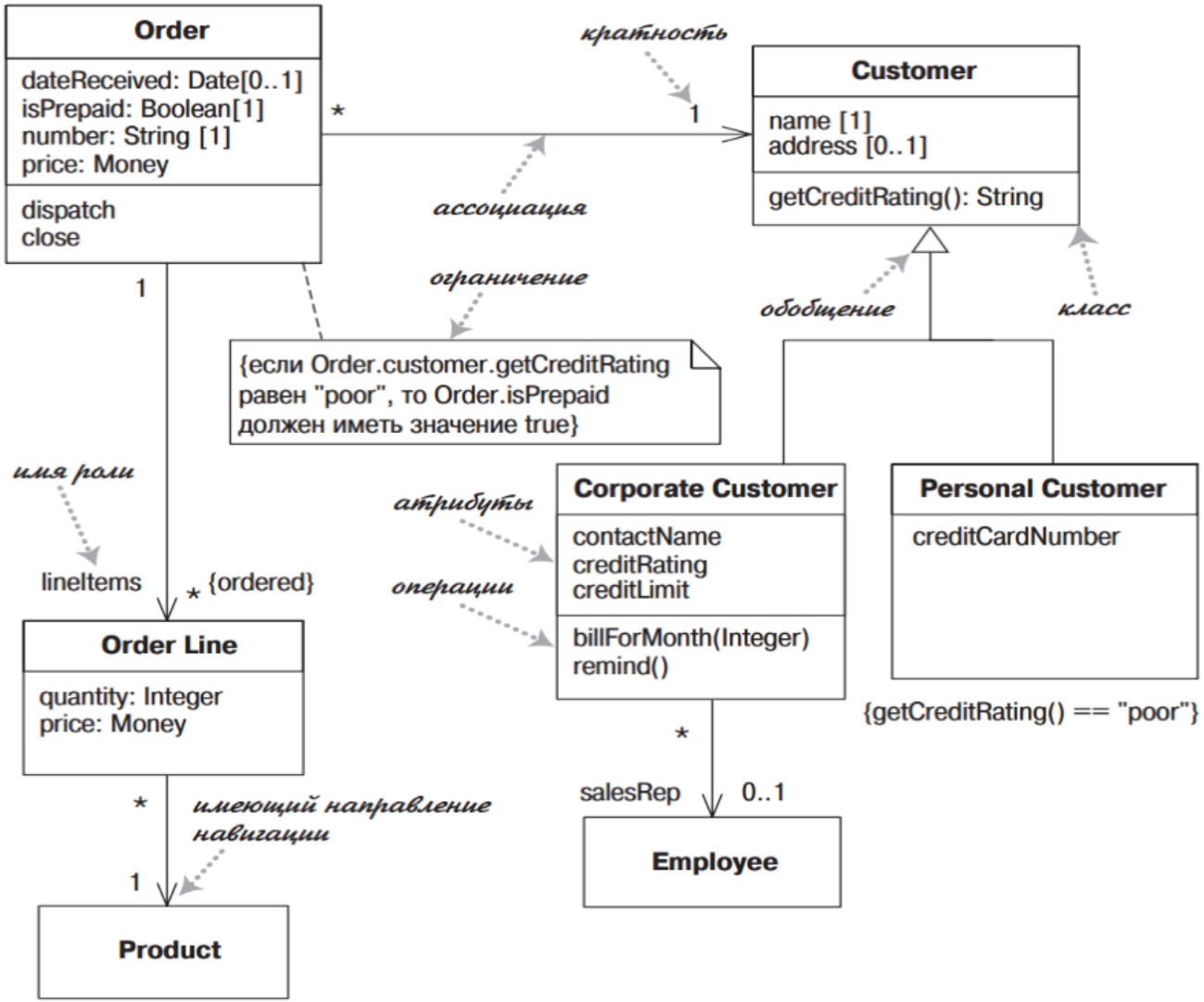
Моделирование структуры

- Отношения “часть-целое”
- Статические свойства
- Не рассматриваем
  - “Причина-следствие”
  - “Раньше-позже”
- Основные сущности — пакеты, классы и отношения между ними

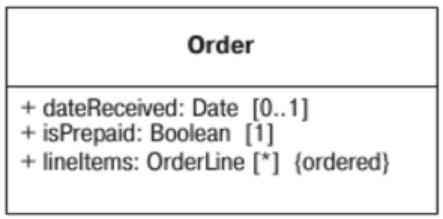
Что моделируем

- Структура связей между объектами во время выполнения программы
- Структура хранения данных
- Структура программного кода
- Структура компонентов в приложении
- Структура артефактов в проекте
- Структура используемых вычислительных ресурсов

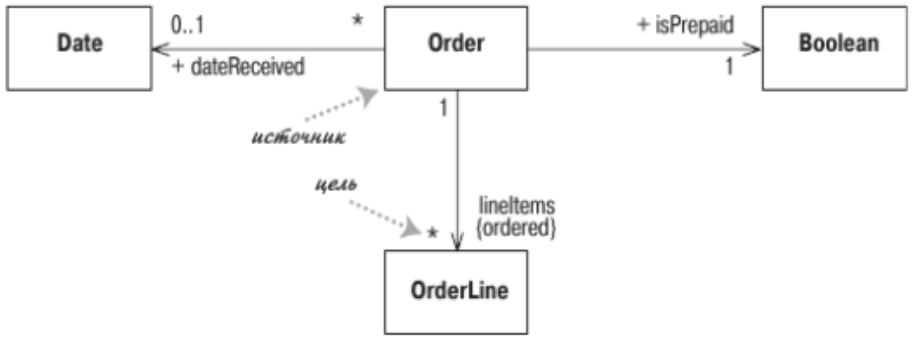
Диаграммы классов UML



Свойства



Атрибуты



Ассоциации

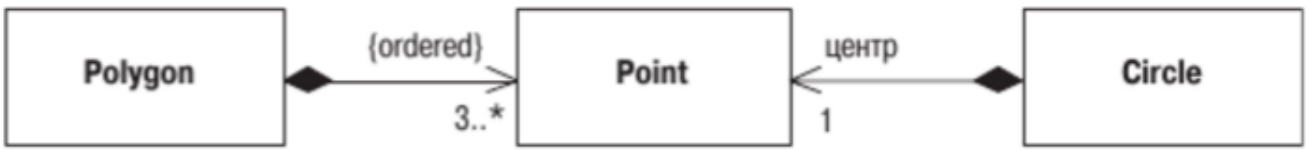
Синтаксис:

- видимость имя: тип кратность = значение по умолчанию {строка свойств}
- Видимость: + (public), - (private), # (protected), ~(package)
- Кратность: 1 (ровно 1 объект), 0..1 (ни одного или один), * (сколько угодно), 1..*, 2..*

Агрегация и композиция

Агрегация – объект “знает” о другом (не управляет его временем жизни, имеет на него ссылку или указатель)

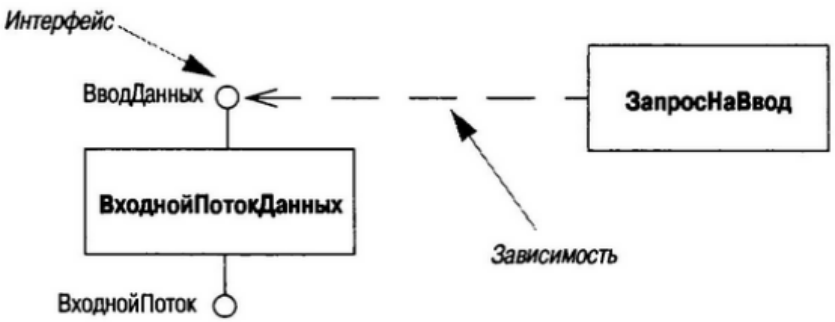
Композиция — объект владеет другим объектом (управляет его временем жизни, хранит его по значению или по указателю, делая delete)



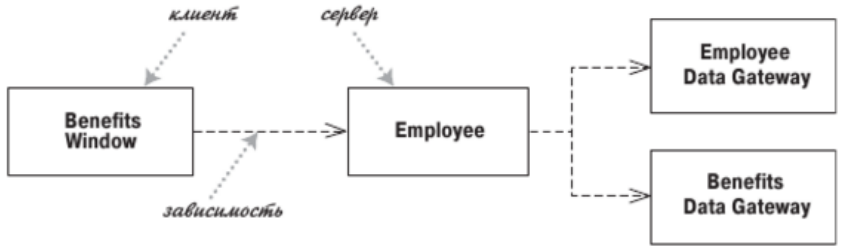
Уточнение обычной ассоциации, используется только если очень надо

Прочее

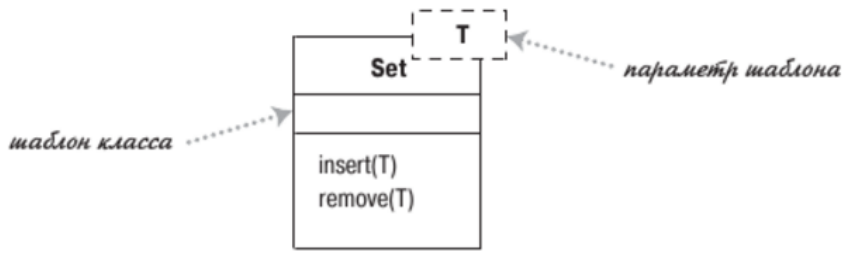
Интерфейсы



Зависимости

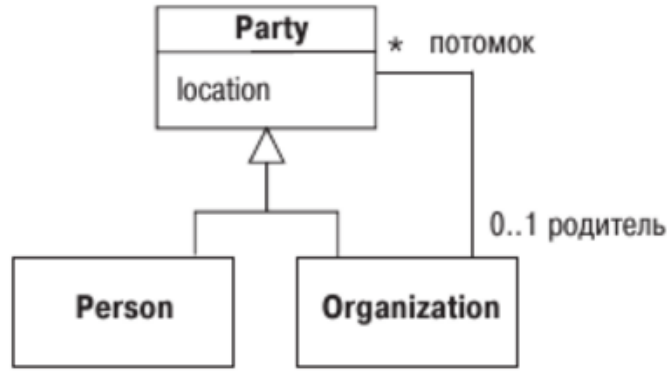


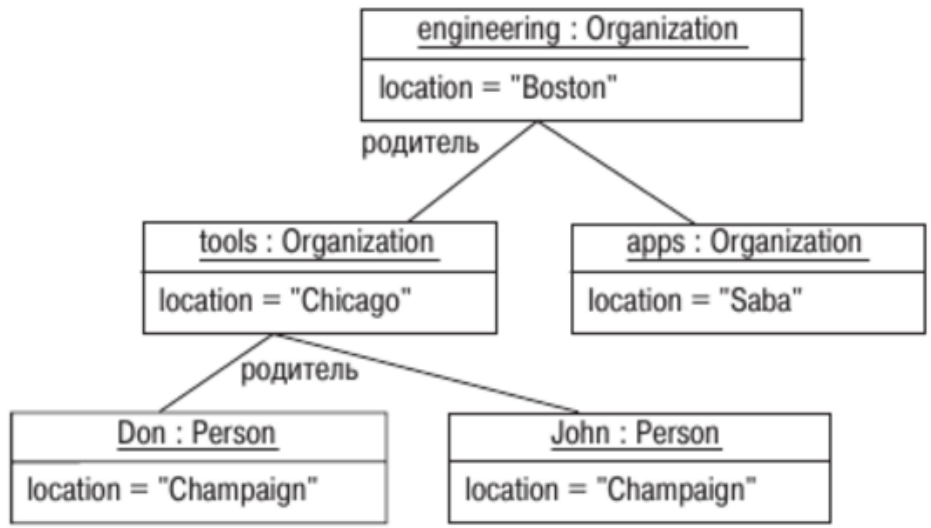
Шаблоны



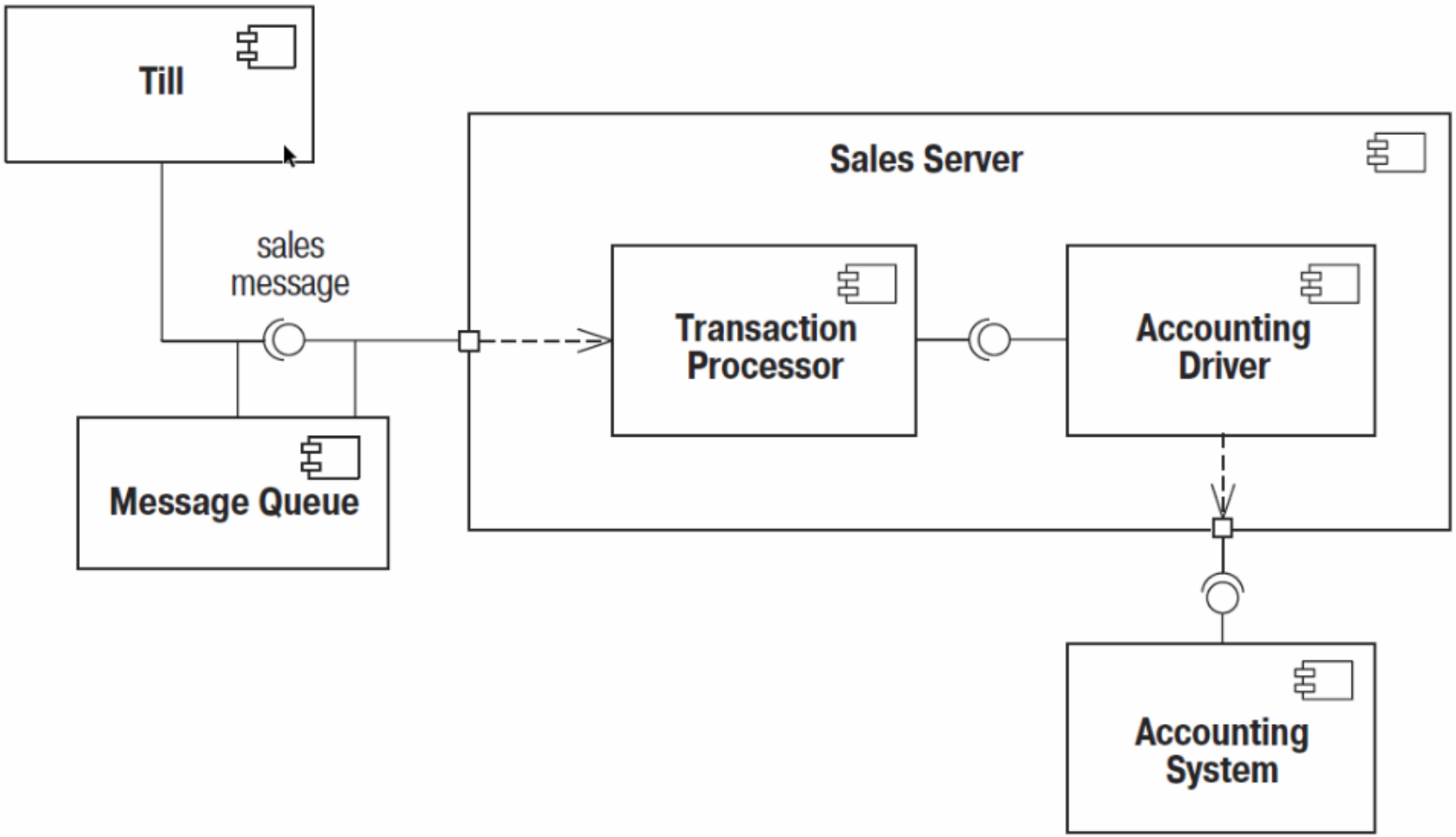
Диаграммы объектов

- snapshot структуры классов во время выполнения
- Используются обычно чтобы пояснить диаграмму классов
- Полезны на этапе анализа предметной области, ещё до диаграмм классов





Диаграммы компонентов



Computer-Aided Software Engineering

- В 80-е годы термином CASE называли всё, что помогает разрабатывать ПО с помощью компьютера
  - Даже текстовые редакторы
- Теперь — прежде всего средства для визуального моделирования (UML-диаграммы, ER-диаграммы и т.д.)
- Отличаются от графических редакторов тем, что “понимают”, что в них рисуют
- Нынче чаще используются термины “MDE tool”, “UML tool” и т.д.

Типичная функциональность CASE-инструментов

- Набор визуальных редакторов
- Репозиторий
- Набор генераторов
- Текстовый редактор
- Редактор форм
- Средства обратного проектирования (reverse engineering)
- Средства верификации и анализа моделей
- Средства эмуляции и отладки
- Средства обеспечения командной разработки
- API для интеграции с другими инструментами
- Библиотеки шаблонов и примеров

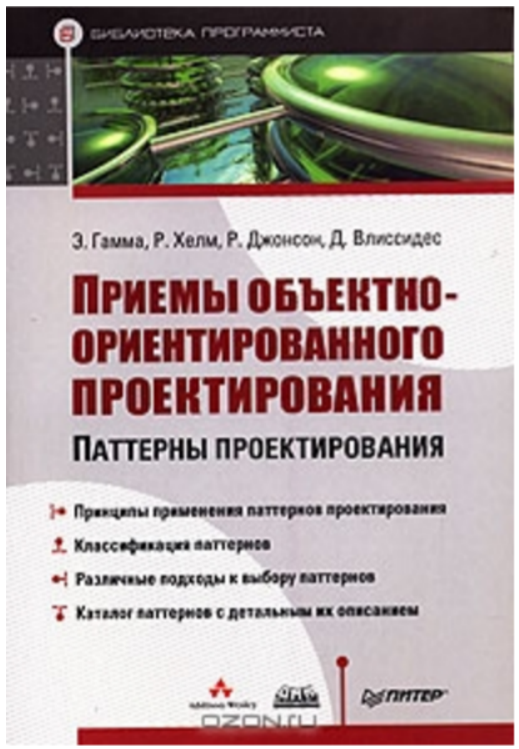
Примеры CASE-инструментов

- “Рисовалки”
  - Visio
  - Dia
  - SmartDraw
  - Creately
- Полноценные CASE-системы
  - Enterprise Architect
  - Rational Software Architect
  - MagicDraw
  - Visual Paradigm
  - GenMyModel
- Забавные штуки
  - https://www.websequencediagrams.com/
  - http://yuml.me/
  - http://plantuml.com/

Паттерны проектирования

Книжка

Приемы объектно-ориентированного проектирования. Паттерны проектирования
Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес
Design Patterns: Elements of Reusable Object-Oriented Software



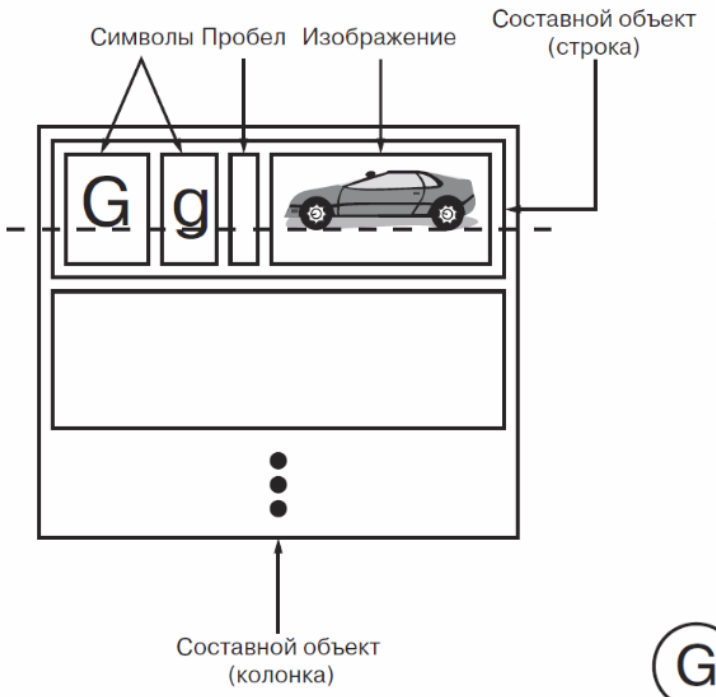
Паттерны проектирования

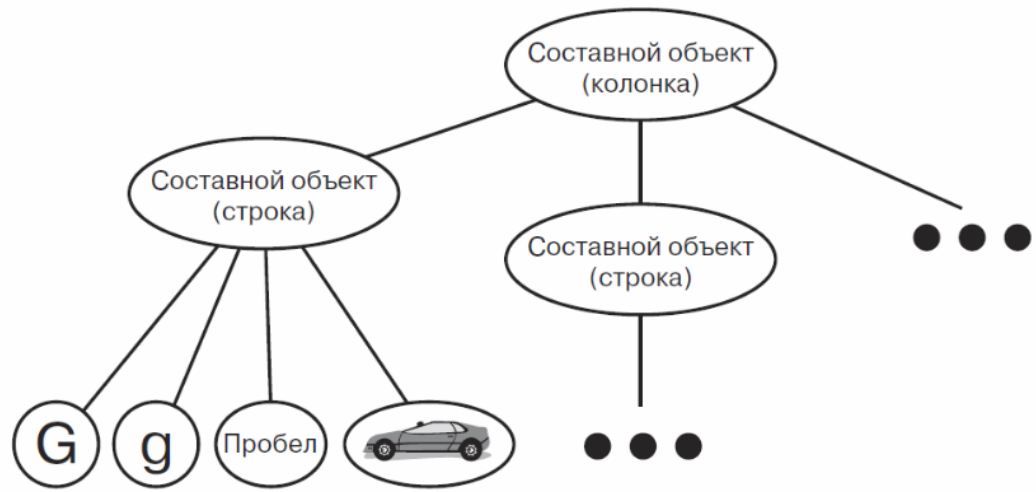
Паттерн проектирования — повторимая архитектурная конструкция, являющаяся решением некоторой типичной технической проблемы

- Подходит для решения целого класса проблем
- Переиспользуемость знаний
- Унификация терминологии
- Простота в изучении
- Опасность карго-культа!

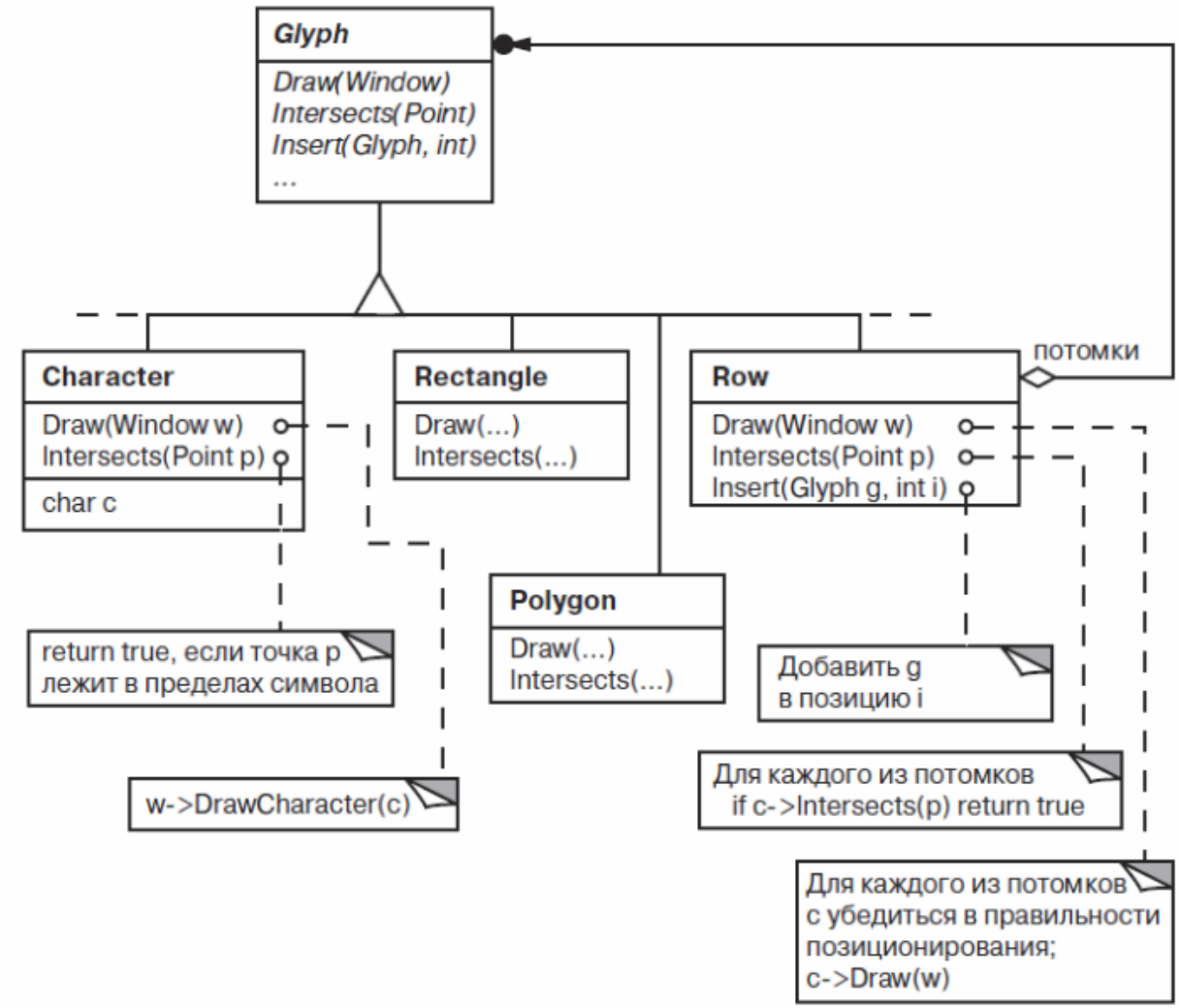
А ещё есть **антипаттерны** — часто встречающиеся неправильные решения типичной проблемы

Пример паттерна, компоновщик (1)

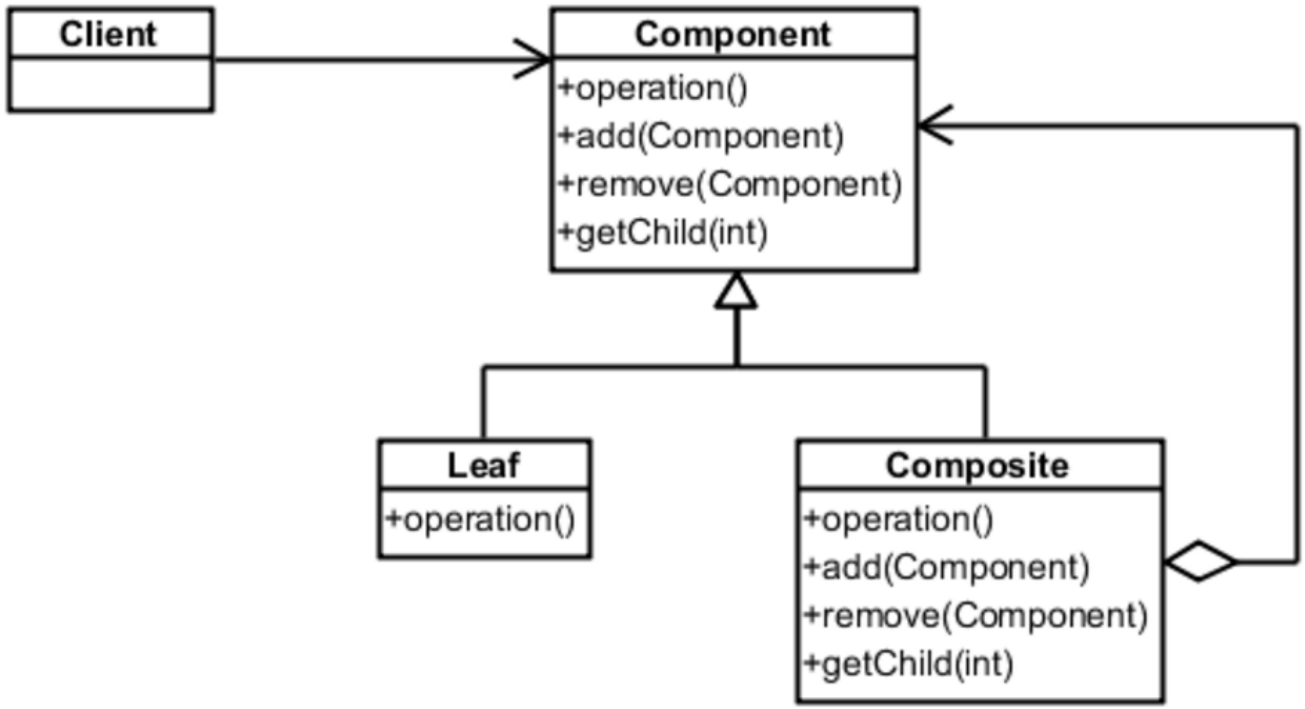




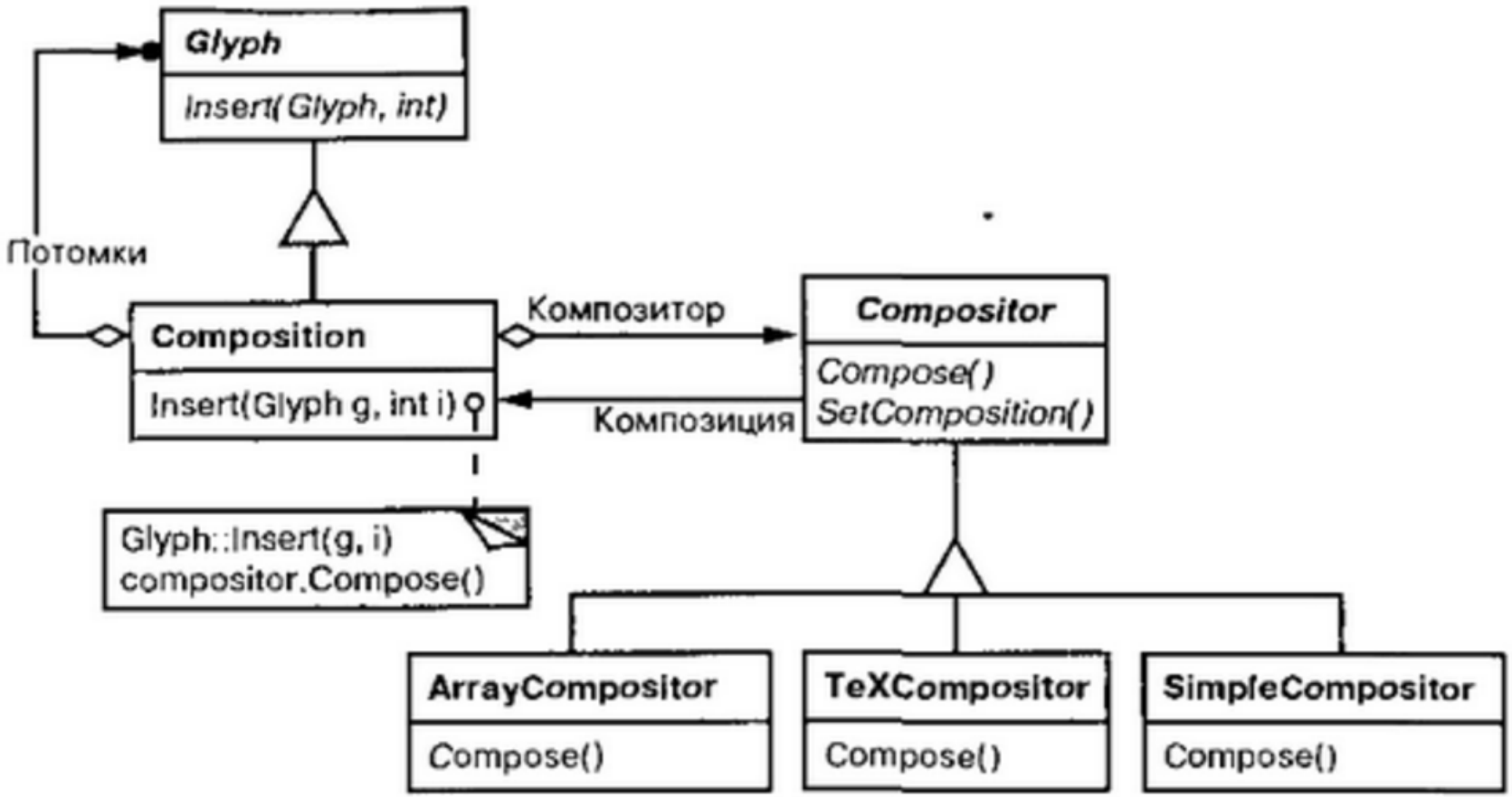
Пример паттерна, компоновщик (2)



Пример паттерна, компоновщик (3)



Ещё пример, паттерн “Стратегия” (1)



Ещё пример, паттерн “Стратегия” (2)