Лекция 13

Введение в объектно-ориентированный анализ и проектирование

Литература

1) Г. Буч, Р.А. Максимчук, М.У. Энгл, Б.Дж. Янг, Дж. Коналлен, К.А. Хьюстон. Объектно-ориенти-рованный анализ и проектирование с примерами приложений. - М.: Вильямс, 2008. - 720 с.

2) К. Ларман. Применение UML и шаблонов проектирования. - М.: Вильямс, 2004. - 624 с.

Разработка программного обеспечения (*англ. software engineering, software development*) — род деятельности, направленный на создание и поддержание работоспособности, качества и надежности ПО, используя технологии из информатики, управления, математики, инженерии и других областей.

Этапы разработки ПО

- Выбор парадигмы программирования
- Анализ требований
- Планирование
- Разработка архитектуры
- Кодирование
- Тестирование и отладка
- Документирование
- Внедрение
- Сопровождение

Модели процесса разработки

- каскадная (небольшие проекты),
- итеративная (крупные проекты).

Сложность ПО

Сравнима со сложностью современных технических средств.

Сложность, присущая ПО:

- Сложность предметной области (множество различных понятий и характер взаимодействий между ними)
- Трудности управления проектированием (большой объем кода)
- Необходимость обеспечения гибкости (изменяющиеся требования)

Признаки сложной структуры

- Иерархическая структура (компоненты + связи)
- Относительность выбора элементарных компонентов
- Разделение функций между компонентами (связи внутри компонентов сильнее связей между ними)
- Общая структура (повторное использование компонентов)
- Устойчивые промежуточные формы (залог быстрого и успешного развития системы)

Декомпозиция

Декомпозиция как основная стратегия борьбы со сложностью ПО.

Метод борьбы — построение иерархий

Пример:
опытный летчик и новый самолет.

Виды иерархий

1. Иерархия "является частью..." ("part of ...") — самолет, двигатель, фюзеляж, крыло и т.д.

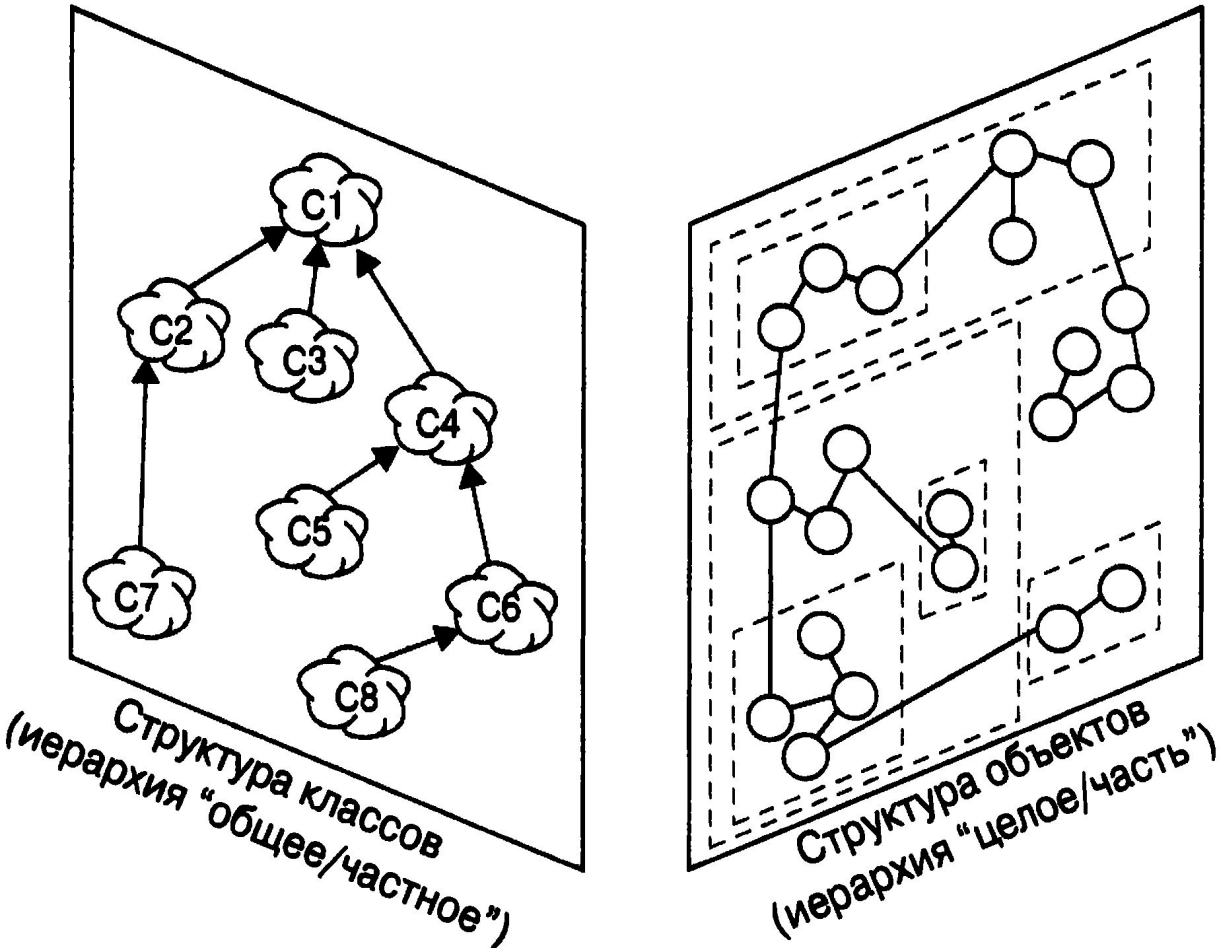
"целое/часть"

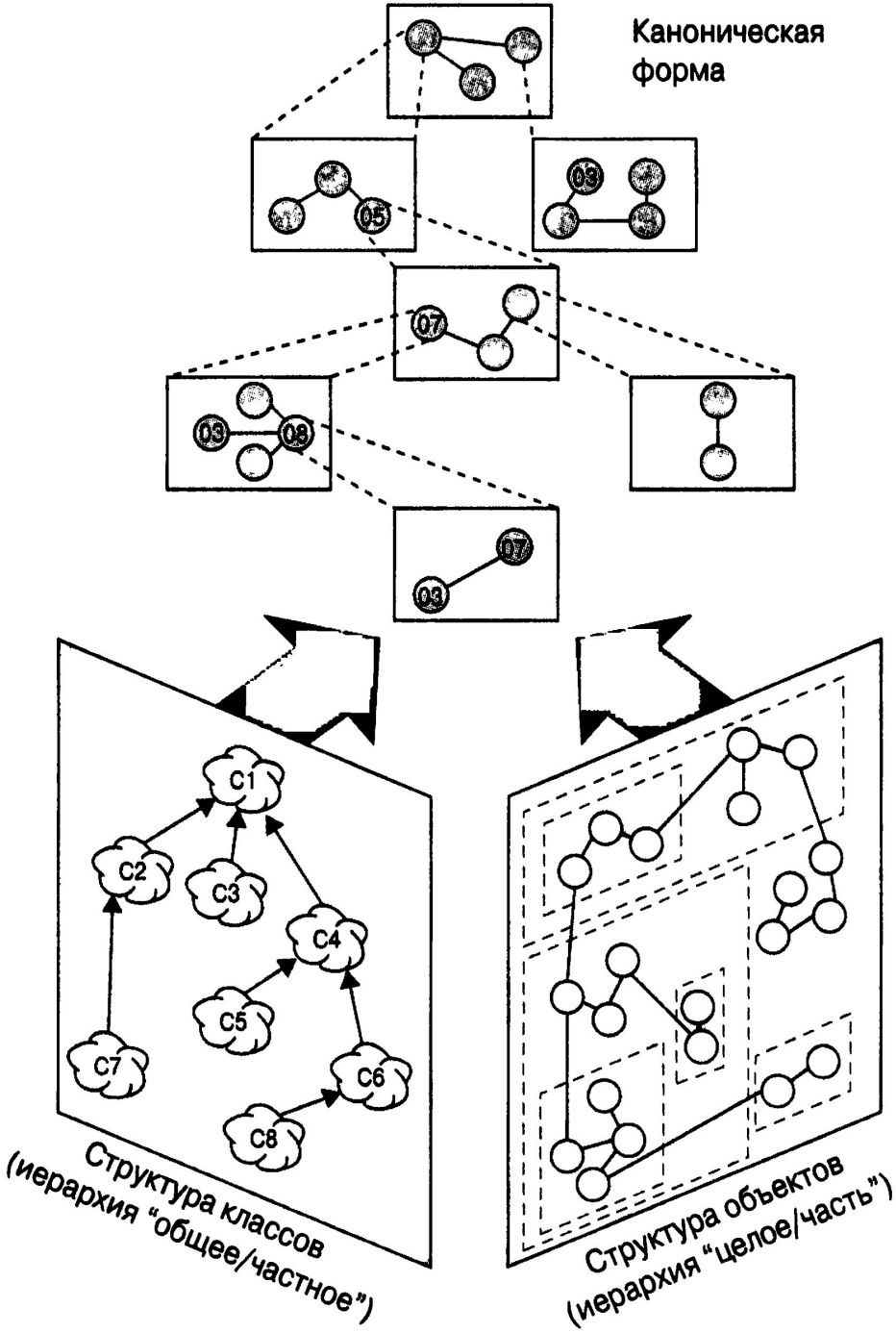
2. Иерархия "является..." ("is a ...") — двигатель самолета → турбореактивный двигатель → разновидность реактивного двигателя и т.д.

"общее/частное"

Иерархии "целое/часть" и "общее/частное" иногда называют **структурой классов** и **структурой объектов.**

Структура классов и структура объектов вместе определяют **архитектуру системы**





Успешными являются программные системы, включающие признаки сложности и основанные на хорошо продуманных структурах классов и объектов.

Объектная модель

OOA – OOD – OOP

Объектно-ориентированное программирование (*англ. OOP, object-oriented programming*) — метод программирования, основанный на представлении программы в виде совокупности взаимодействующих объектов. Объекты являются экземплярами определенного класса, а классы — членами иерархии наследования.

Объектно-ориентированное проектирование (*англ. OOD, object-oriented design*) — метод проектирования, сочетающий процесс объектно-ориентированной декомпозиции и систему обозначений для представления модели.

Объектно-ориентированный анализ (*англ. OOA, object-oriented analysis*) — метод анализа, исследующий требования к системе с точки зрения классов и объектов, относящихся к словарю предметной области.

Абстрагирование

Метод борьбы со сложностью.

Абстракция выделяет существенные характеристики объекта, отличающие его от всех других видов объектов. Таким образом, четко описываются границы объекта с точки зрения наблюдателя.

Принцип "наименьшего удивления"
("least astonishment")

- Абстракция сущности

  Объект, представляющий собой полезную модель некой сущности в предметной области

- Абстракция действия

  Объект, состоящий из обобщенного множества операций, каждая из которых выполняет однотипные функции

Пример абстракции: температурный датчик

Абстракция:	Temperature Sensor
Важные свойства:	
	температура местоположение
Обязанности:	
	сообщать о текущей температуре калибровать

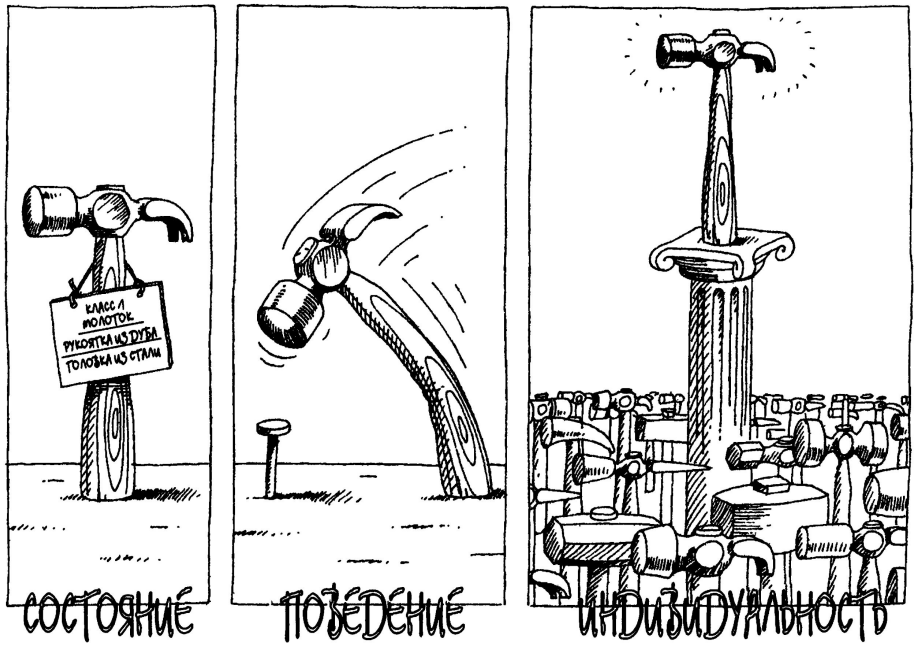
Абстракция:	Active Temperature Sensor
Важные свойства:	
	температура местоположение пороговое значение
Обязанности:	
	сообщать о текущей температуре калибровать установить пороговое значение

Природа объекта

Объектом может быть

- Материальный и (или) видимый предмет.
- Сущность, поддающаяся осмыслению.
- Сущность, на которую направлена мысль или действие.

основные характеристики объекта



Природа класса

Класс — множество объектов, обладающих общей структурой и поведением. Отдельный объект является экземпляром класса.

Интерфейс класса:

1) Открытый раздел (public) — объявления, доступные всем клиентам.

2) Защищенный раздел (protected) — объявления, доступные самому классу и его подклассам.

3) Закрытый раздел (private) — объявления, доступные только самому классу.

4) Пакет (package) — объявления, доступные только классам, входящим в один и тот же пакет.

Объектно-ориентированный анализ

На этапе анализа осуществляется **первичное исследование** задачи и моделирование реальности с помощью классов и объектов, образующих словарь предметной области. На этапе проектирования изобретаются абстракции и механизмы реализации принятого решения.

Классический подход к идентификации классов и объектов

• Материальные предметы	Автомобили, телеметрические данные, датчики давления.
• Роли	Мать, учитель, политик.
• События	Посадка, прерывание, запрос.
• Взаимодействие	Заем, встреча, пересечение.

От понятия предметной области к программному классу.

Анализ предметной области

Анализ предметной области как "попытка идентифицировать объекты, операции и отношения, которые эксперты считают наиболее важными".

Этапы анализа предметной области:

- Построение модели предметной области в ходе консультаций с экспертами в этой области.
- Изучение систем, существующих в данной предметной области, и представление результатов в стандартном виде.
- Идентификация сходства и различий между системами при участии экспертов предметной области.

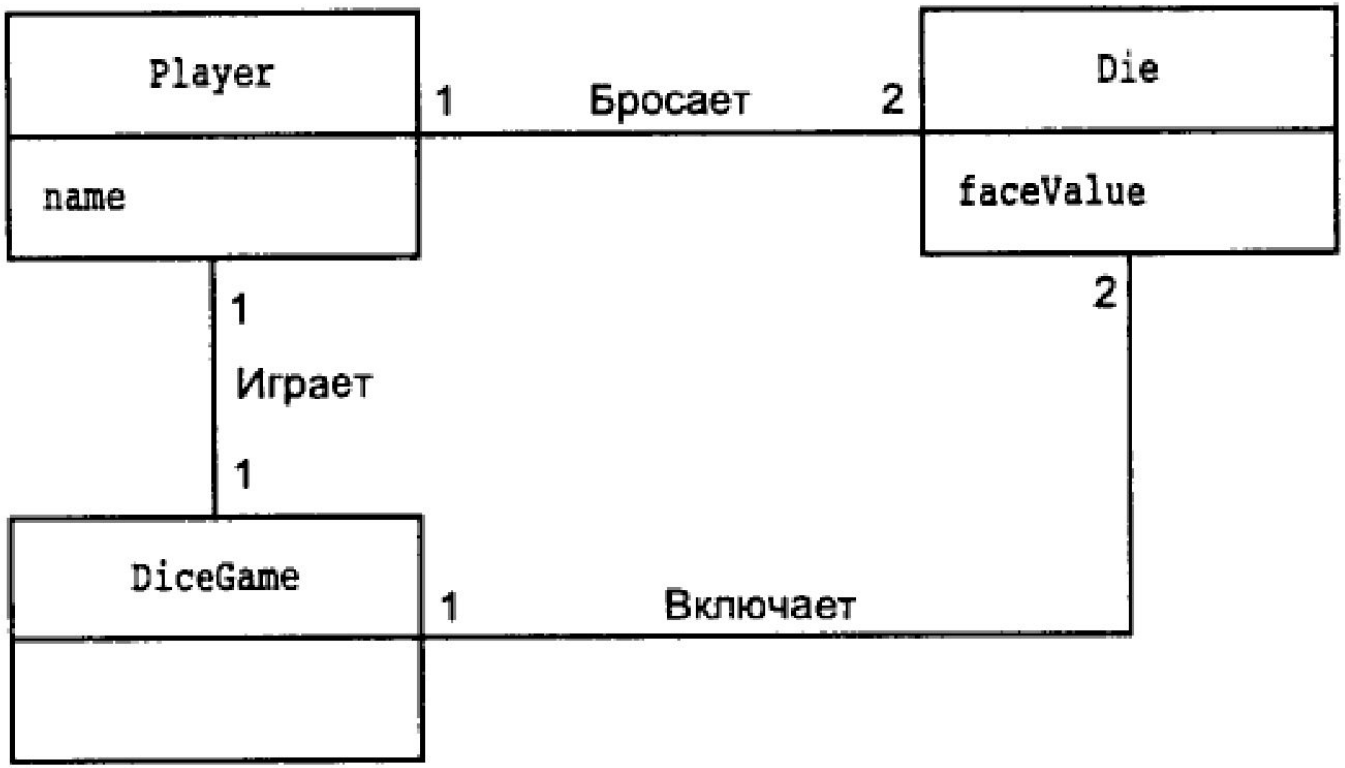
Короткий пример ООА/П: игра в кости.

Описание игры. Игрок бросает две кости. Если сумма очков равна 7, то игрок побеждает, в противном случае - проигрывает.

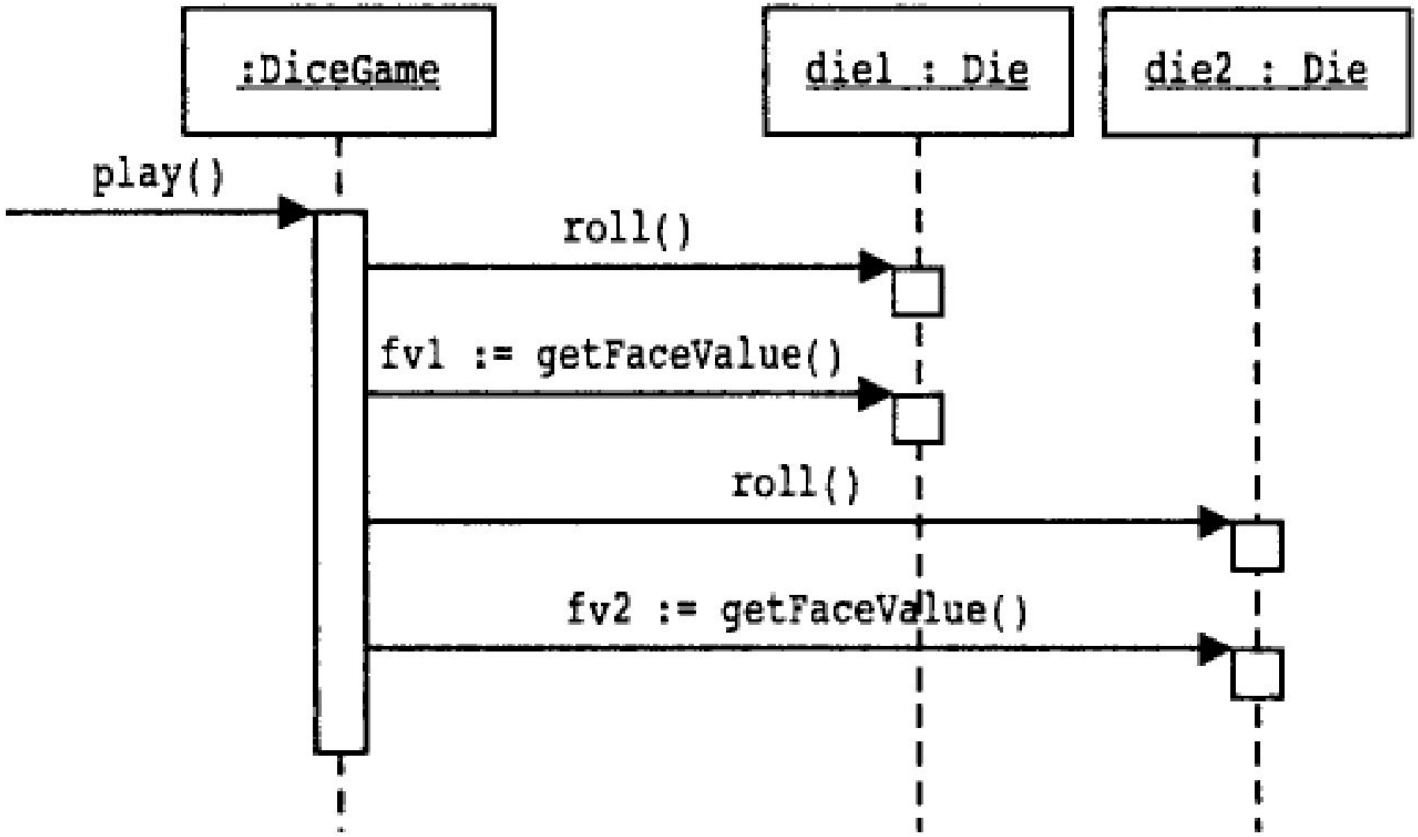
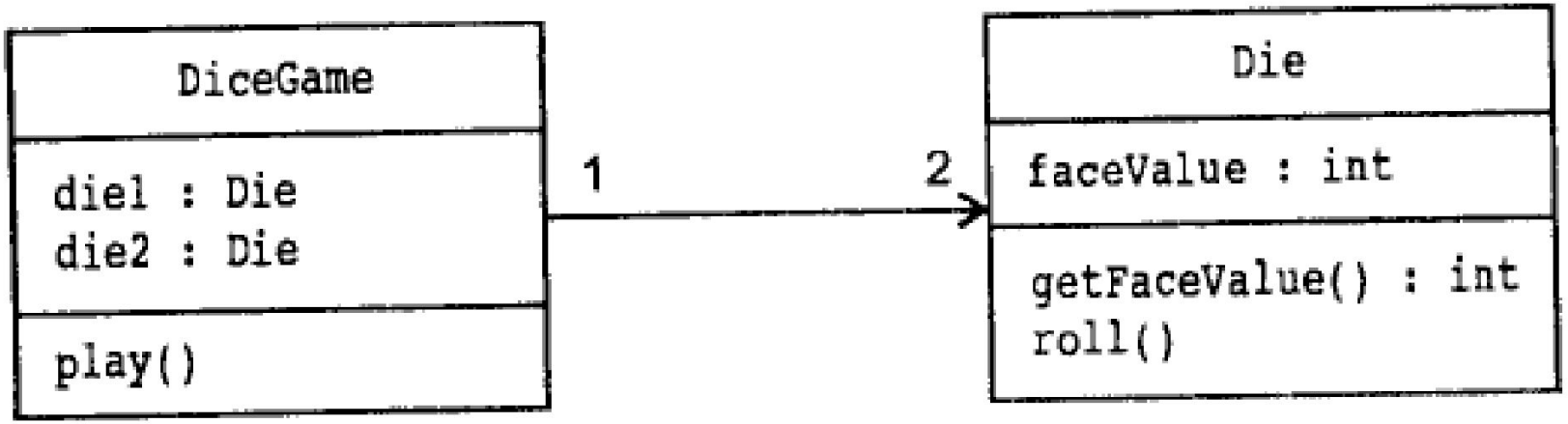
1) Анализ прецедентов

Прецедент: "Игра в кости". **Описание:** Игрок берет и бросает кости. Вычисляется сумма выпавших очков. Если сумма равна 7, то игрок считается победителем, иначе - проигравшим.

2) Определение модели предметной области (domain model)



3) Построение диаграмм взаимодействий (collaboration diagram)



4) Построение диаграмм классов проектирования (design class diagram)