



IT-Студия WebMax.BY, г. Минск

Объектно-ориентированный подход в разработке ПО

Курс online-тренингов

Практический анализ и моделирование на UML

Тема 1

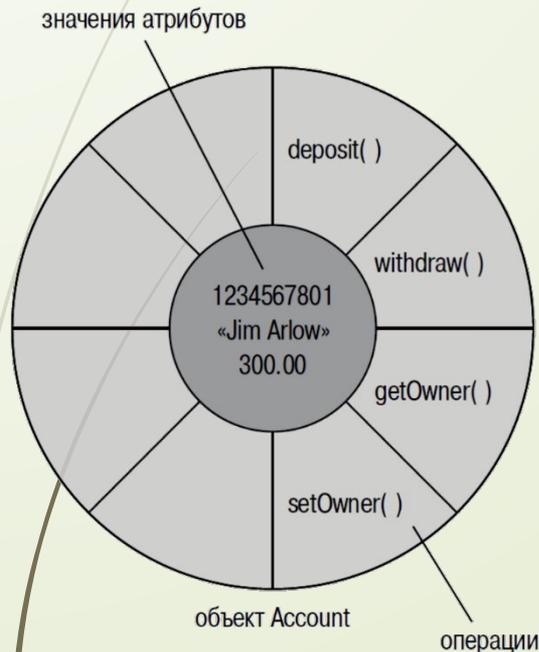
Николай Киреев

Минск – 2015

- Что такое объектно-ориентированная программная система
- Объекты и классы
- Нотации объектов и классов в UML и RUP
- Связи объектов и отношения классов

Что такое объектно-ориентированная программная система?

▣ **Архитектура** (*architecture*) – логическая (классы и их отношения) и физическая структуры системы (объекты, связи, сообщения), сформированная всеми стратегическими и тактическими проектными решениями. [Г. Буч, 1]

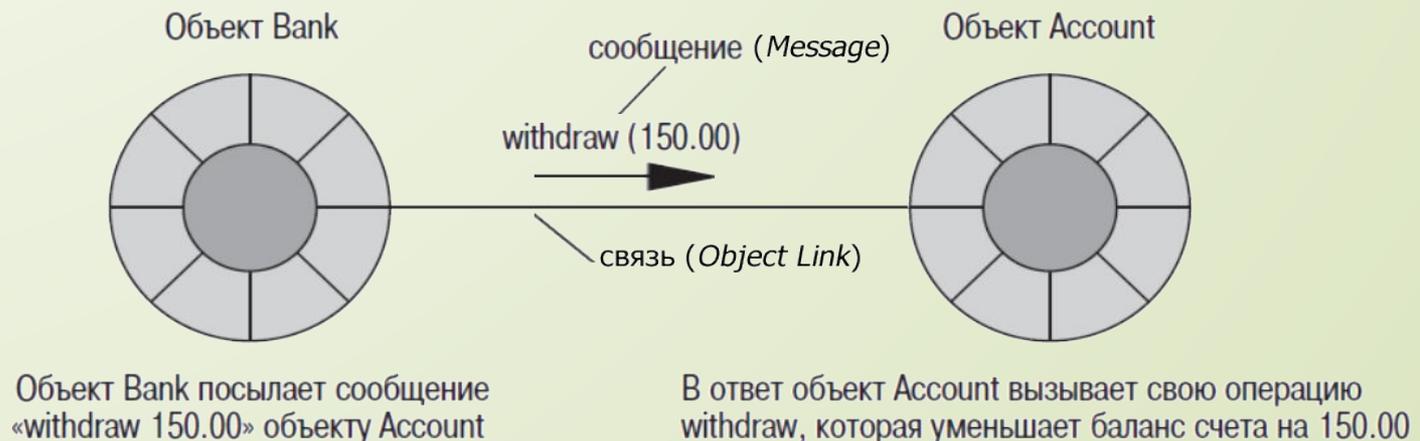


- ▣ Объектно-ориентированная программная система – это совокупность взаимосвязанных взаимодействующих объектов, каждый из которых является экземпляром класса.
- ▣ Объект объединяет (**инкапсулирует**) данные и функциональность в единый блок, это сущность с явно выраженными границами, обладающая индивидуальностью, которая характеризуется состоянием и поведением;
- ▣ Класс (*class*) – это абстракция совокупности реальных объектов, которые имеют общий набор свойств и обладают одинаковым поведением.
- ▣ Классы взаимосвязаны между собой и являются членами иерархии наследования.

Что такое объектно-ориентированная программная система?

- В значениях атрибутов сохраняются данные (существенные характеристики), ими определяется состояние объекта.
- Единственный путь добраться до данных объекта – вызвать одну из предоставляемых им функций. Эти функции называются операциями (*operations*).
- **Функциональное поведение программной системы реализуется благодаря взаимодействию объектов, при котором объект «клиент» посылает сообщение объекту «сервер» с целью вызова соответствующего метода последнего.**

Модель взаимодействия объектов «Клиент – Сервер» [2]



Объекты и классы

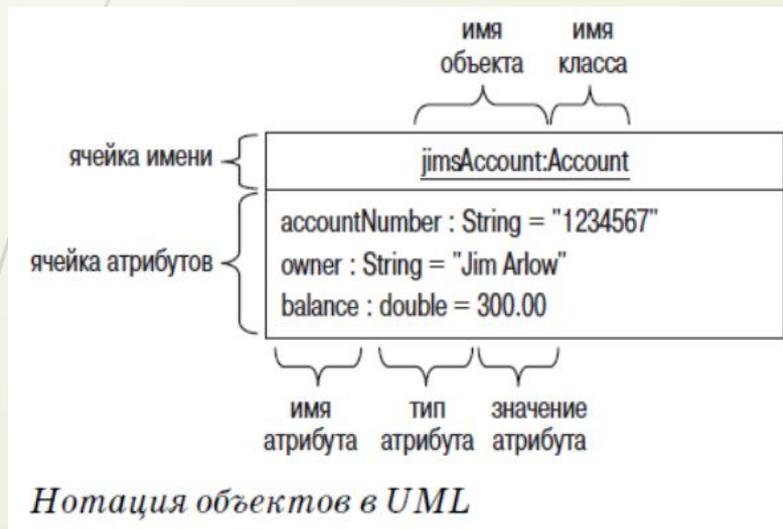
- Каждый объект имеет уникальный идентификатор.
- Поведение объекта – это то, «что он может сделать для нас», т. е. его методы.
- Объекты формируют поведение системы путем обмена сообщениями по связям. Это – кооперация.
- Каждый объект – это экземпляр только одного класса.
- Класс описывает свойства группы однотипных объектов.
- Первоначально, классы создаются на основе сущностей (объектов) предметной области.
- При создании экземпляра класса создается новый объект. Класс используется как шаблон.
- Экземплярами класса могут быть объекты, другие классы, класс может не иметь экземпляров (абстрактный класс).
- Объект – это экземпляр класса, связь между объектами – экземпляр отношений классов, каждое сообщение это вызов метода, метод – это реализация операции класса.

Объекты и классы

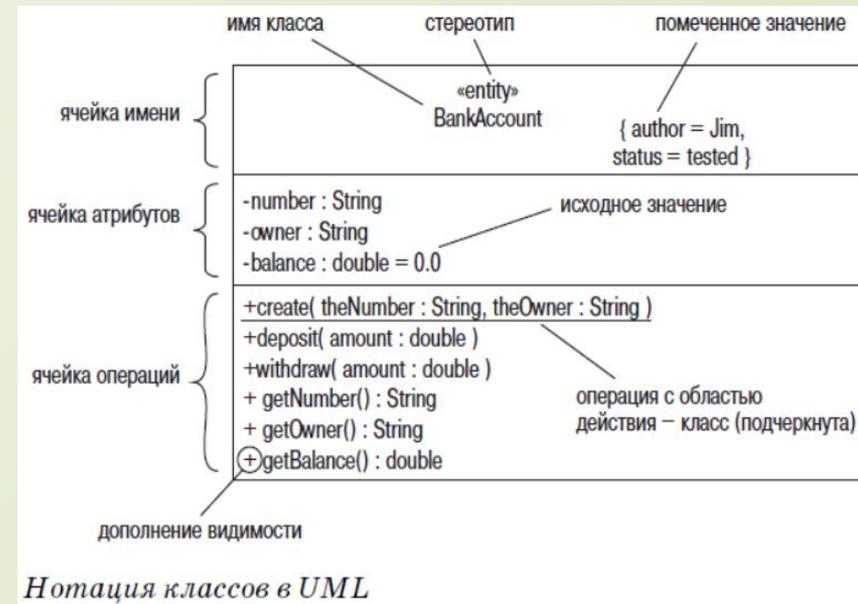
- Имя класса отражает его назначение.
- Класс является четкой абстракцией, моделирующей один конкретный элемент предметной области.
- Обязанности класса описывают связный набор операций. Обязанность – это контракт или обязательство класса по отношению к его клиентам. По существу, обязанность – это сервис, который класс предлагает другим классам. Обязанности сущностей, в том числе классов, в UML могут быть разграничены с помощью **стереотипов**.
- *Объектно-ориентированный анализ – это метод анализа, исследующий требования к системе с точки зрения классов и объектов, относящихся к словарю предметной области.*
- Сущности со стереотипом **business entity** моделируют группу однотипных объектов предметной области. Они являются основными кандидатами в классы программной системы.

Нотации объектов и классов в UML и RUP

- Базовая нотация объектов (нотация проектировщика)



- Базовая нотация классов (классы проектирования)



Нотации стереотипов объектов и классов в UML и RUP

- Объект не соотнесённый с классом

Счёт 35-24

A yellow rectangular box with a red border and a red underline, containing the text 'Счёт 35-24'.

- Примеры объектов, соотнесённых с классами (базовый элемент)

Марта : Кошка

A yellow rectangular box with a red border and a red underline, containing the text 'Марта : Кошка'.

Колли : Собака

A yellow rectangular box with a red border and a red underline, containing the text 'Колли : Собака'. It is part of a stack of three overlapping boxes.

- Примеры объектов, соотнесённых с стереотипизированными классами

Иванов П.П. : ClientAkk

A light blue circle with a black diagonal line from the bottom-left to the top-right, positioned above a horizontal red line. Below the line is the text 'Иванов П.П. : ClientAkk'.

Счёт 35-24 : Account

A yellow circle positioned above a horizontal red line. Below the line is the text 'Счёт 35-24 : Account'.

ЭФ Ввода Данных : GUI

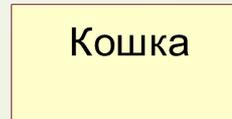
A yellow circle with a vertical red line extending from its left side, positioned above a horizontal red line. Below the line is the text 'ЭФ Ввода Данных : GUI'.

Сумматор : Счётчик

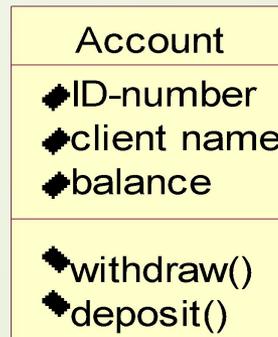
A yellow circle with a black diagonal line from the bottom-left to the top-right, positioned above a horizontal red line. Below the line is the text 'Сумматор : Счётчик'.

Нотации стереотипов объектов и классов в UML и RUP

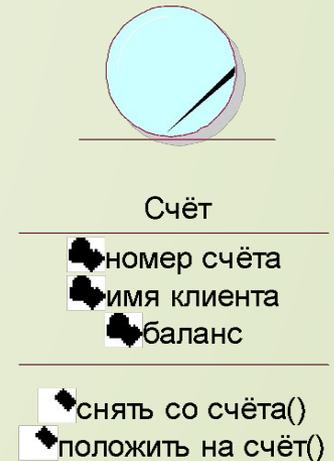
Класс (без атрибутов и операций)



Класс проектирования (базовый элемент)

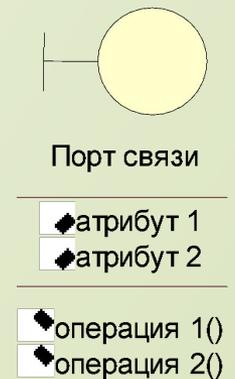


Стереотип *business entity*



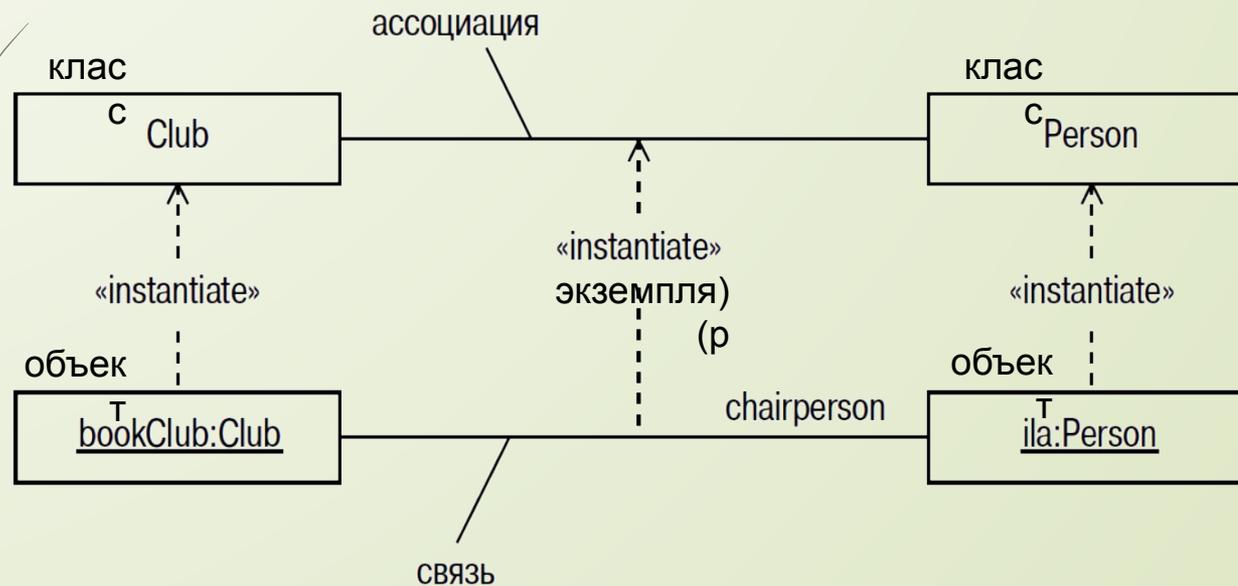
Стереотипы анализа в RUP:

- entity
- control
- boundary



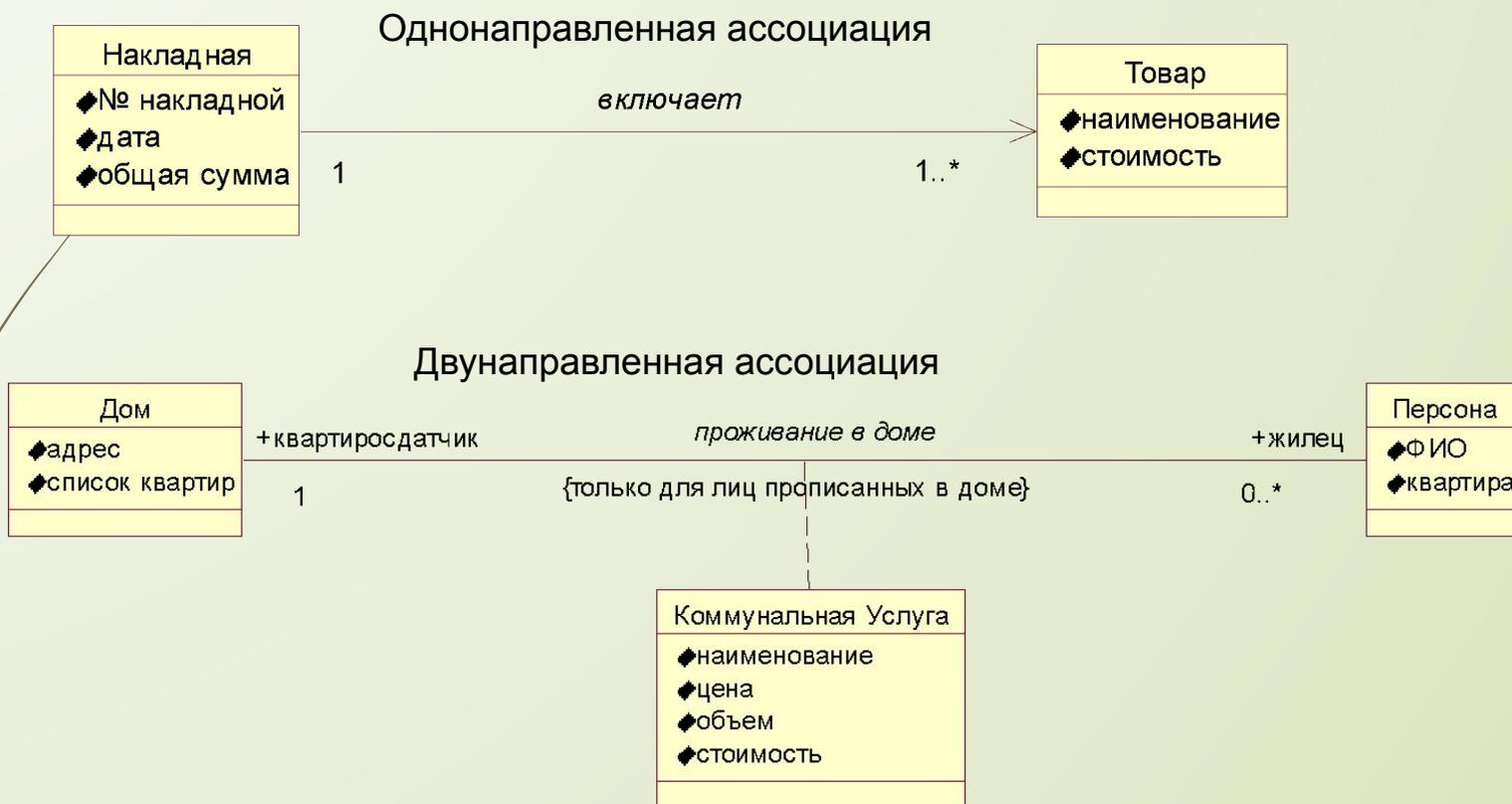
Связи объектов и отношения классов

- Объекты обмениваются сообщениями (message) через соединения (object link), называемые связями [2].
- Связи между объектами являются экземплярами отношений между классами, называемыми ассоциациями. К последним в UML относятся: одно- и двунаправленные ассоциации (association) и агрегации (aggregation & composition).

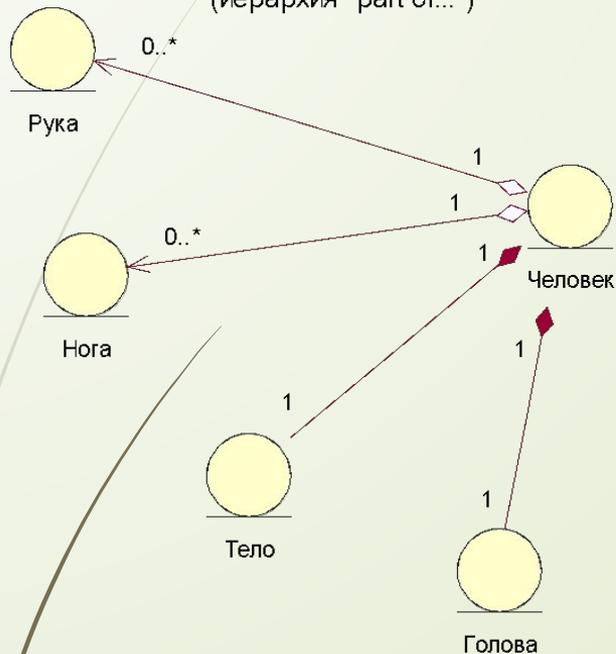


Отношения классов

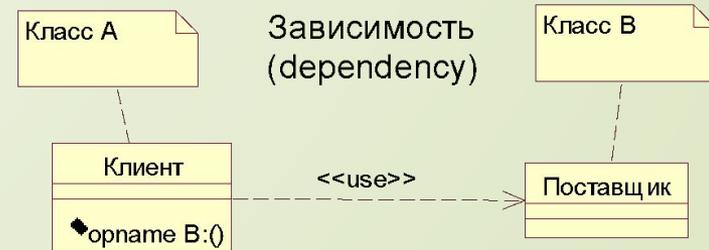
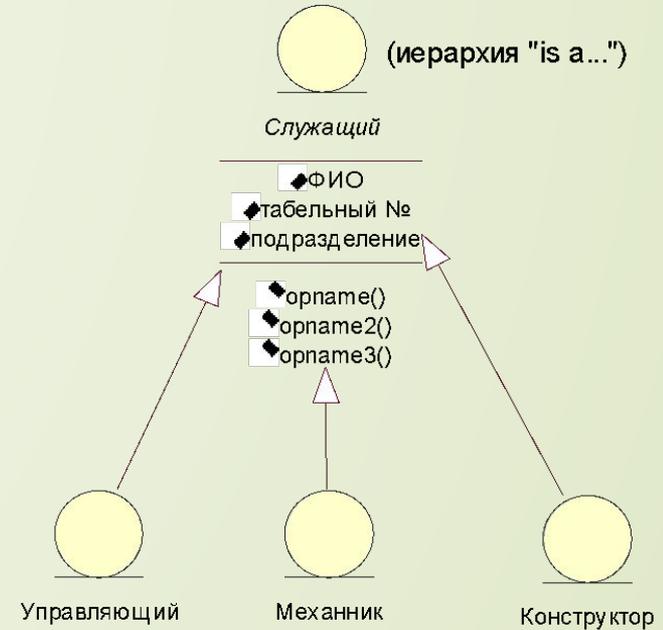
- **Ассоциации**, а также их более сильные типы **агрегации** и **композиции** бывают одно- и двунаправленными. При установлении этого типа отношений исходный класс получает атрибуты целевого класса



Агрегация и композиция одно- и двусторонние
(иерархия "part of...")



Обобщение (Generalization)



Используется если:

1. Операции класса А необходим параметр класса В.
2. Операция класса А возвращает значение класса В.
3. Операция класса А где-то в своей реализации использует объект класса В, но не в качестве атрибута.

1. Буч и д.р.«Объектно-ориентированный анализ и проектирование с примерами приложений», Москва, «Вильямс», 2008 г., 3-е издание.
2. Дж. Арлоу, А. Нейштадт, «UML-2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование». Санкт-Петербург, «Символ-Плюс», 2007 г., 2-е издание.



IT-Студия WebMax.BY, г. Минск

Аналитическая модель

Курс online-тренингов

**Практический анализ и
моделирование на UML**

Тема 2

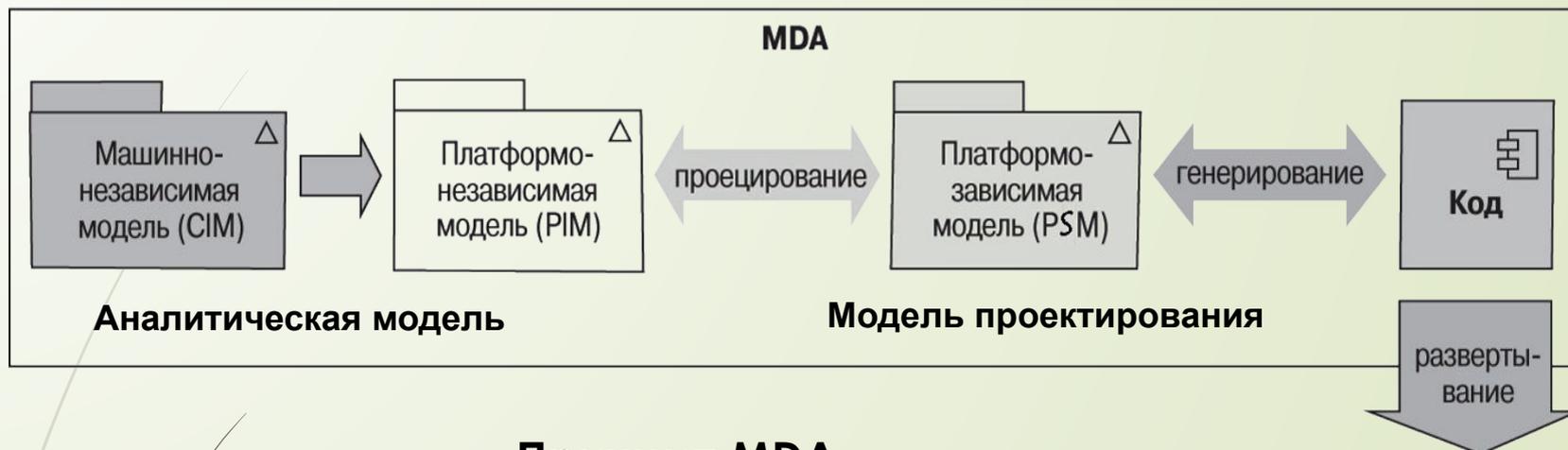
Николай Киреев

Минск – 2015

Тема 2. Аналитическая модель

- Аналитическая модель
- Моделирование сущностей предметной области

Для чего нужны модели?



Принцип MDA

Архитектура, управляемая моделью
(Model Driven Architecture, MDA)

- Программное обеспечение создаётся в результате ряда трансформаций модели при поддержке инструмента моделирования.
- Аналитическая модель описывает предметную область и программную систему с разных точек зрения (представлений).
- Все представления модели направлены на единое целое – проект.

Что такое аналитическая модель

Аналитическая модель – это точное, четкое представление задачи, позволяющее отвечать на вопросы и строить решения.

На этапе проектирования должна использоваться аналитическая модель, а не исходная формулировка задачи.

Аналитическая модель включает две части:

- Модель предметной области (*domain model*);
- Модель программной системы (*application model*).

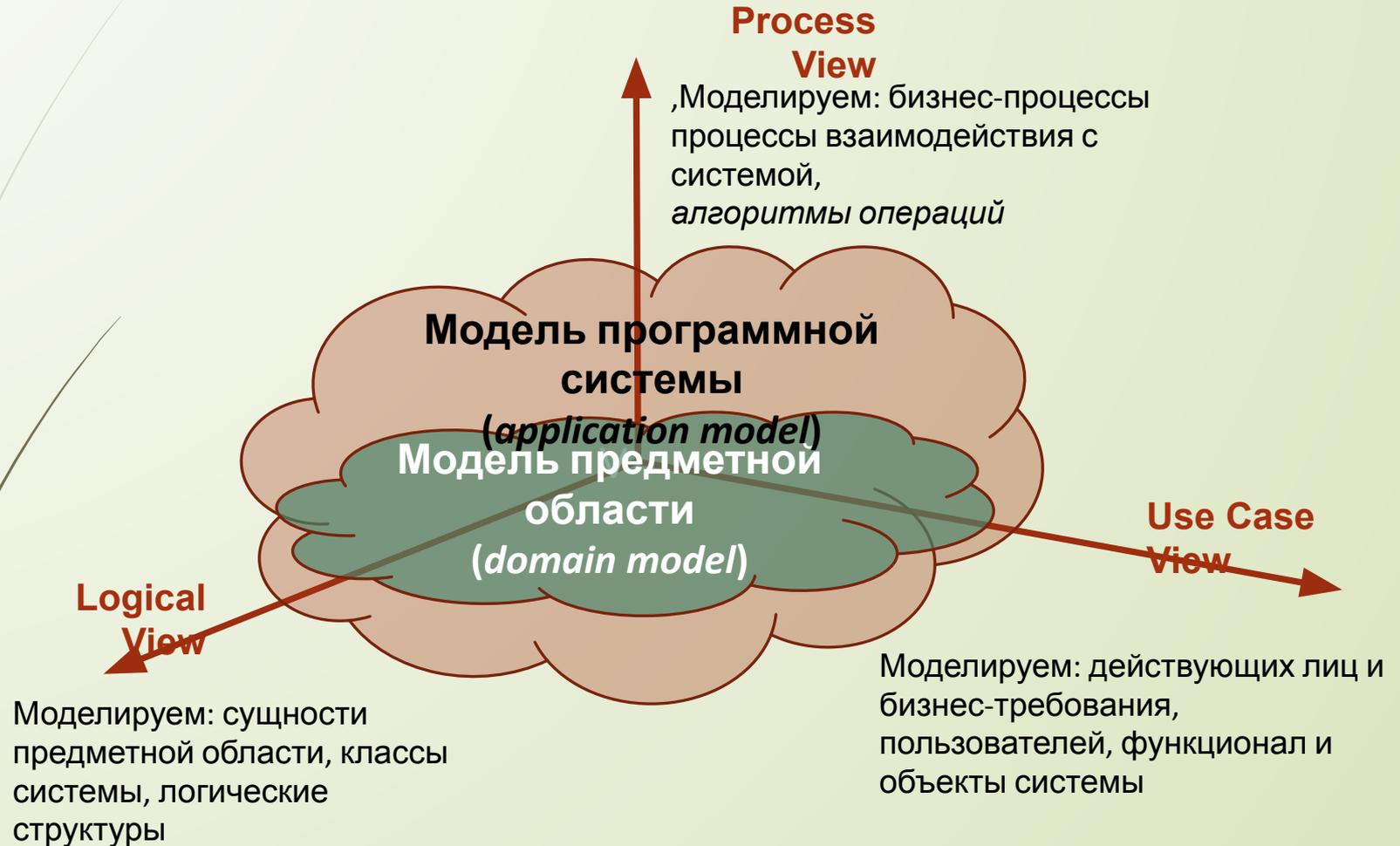
Представление – это точка зрения с которой рассматривается предметная область или программная система.

Представления аналитической модели (в RUP)

1. **Представление классов (Logical View).** Моделируем: сущности предметной области (*business entity*), классы анализа (*boundary, entity, controll*), логическую структуру данных;
2. **Представление прецедентов (Use Case View).** Моделируем: варианты использования (*use case*), пользователей (*actor*), объекты классов анализа, их связи и взаимодействие;
3. **Представление процессов & состояний (Process View).** Моделируем: бизнес-процессы, последовательности действий в вариантах использования, алгоритмы операций.

Основные представления аналитической модели

Проект – это единый объем, рассматриваемый с разных точек зрения



Что поясняет domain model?

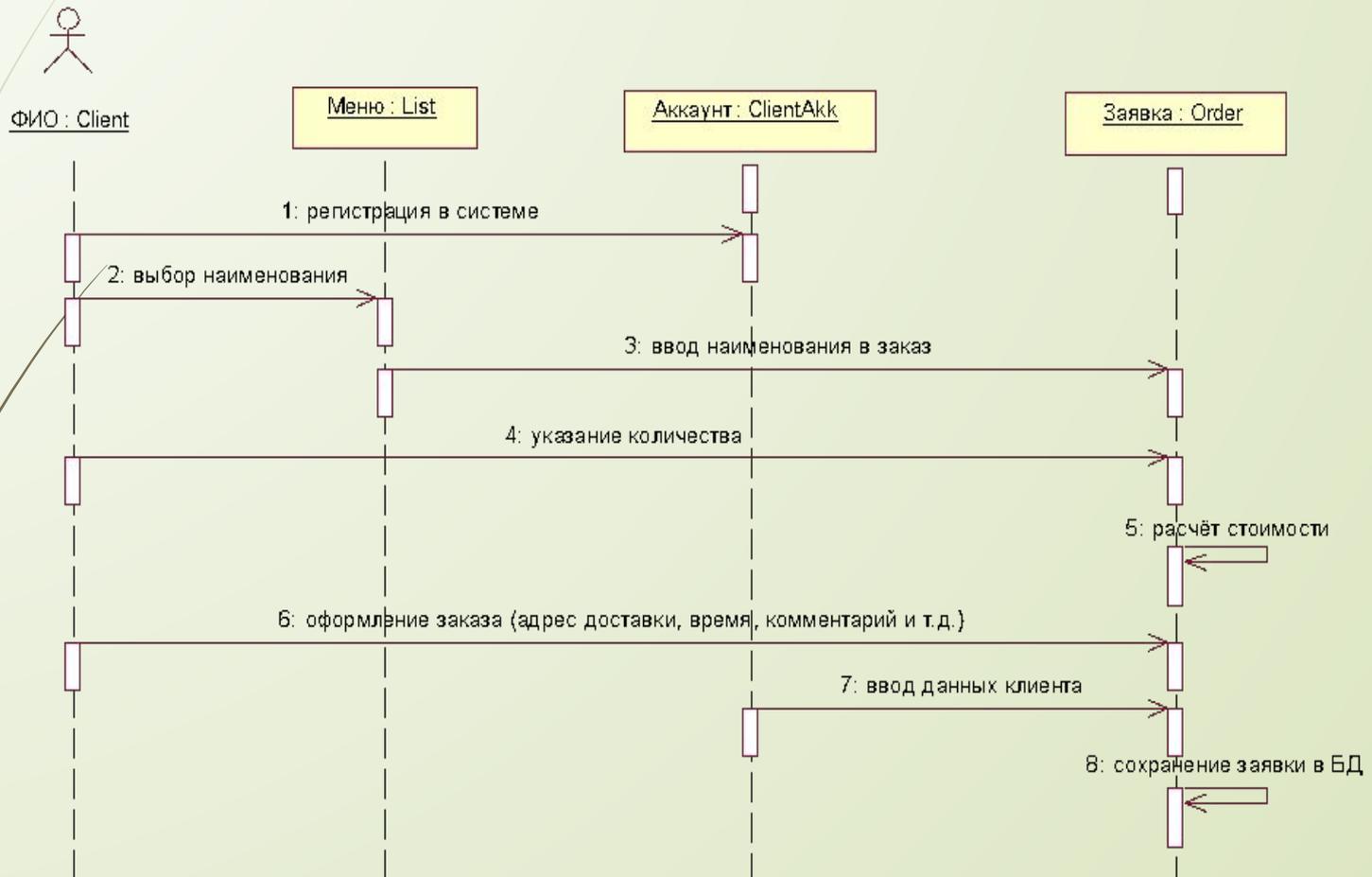
- как работает бизнес заказчика
- проблемы, которые нужно решить, и цели, которые планируется достигнуть с помощью программной системы
- пользователей и требования, которым должна удовлетворять программная система

Что поясняет domain model?

Чтобы понять как работает бизнес заказчика нужно:

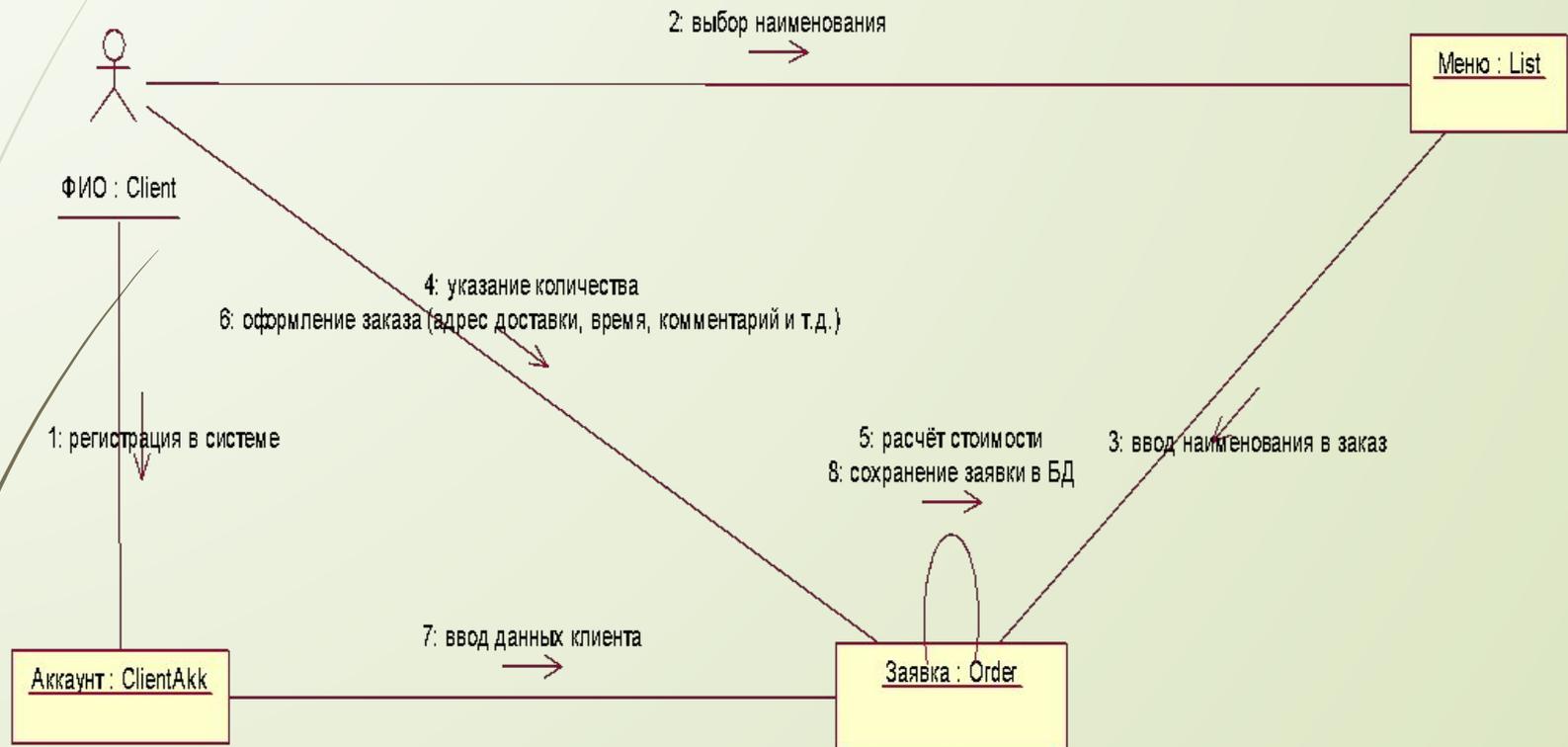
- Определить понятия и сущности предметной области (*моделирование сущностей предметной области, их атрибутов и при необходимости их взаимосвязей*)
- Выявить заинтересованных лиц и определить бизнес-процессы (обязанности, потребности или сервисы), которые они инициируют (*моделирование действующих лиц и бизнес-прецедентов*)
- Описать бизнес-процессы (*моделирование бизнес-Сущности предметной области (Business Entity) – элемент модели, представляющий объект (группу объектов), имеющий атрибуты и, относящийся к глоссарию предметной области*)

□ Объектная модель Диаграмма последовательности (sequence diagram)



Модели сущностей предметной области

- **Объектная модель** Диаграмма коммуникации (collaboration diagram)



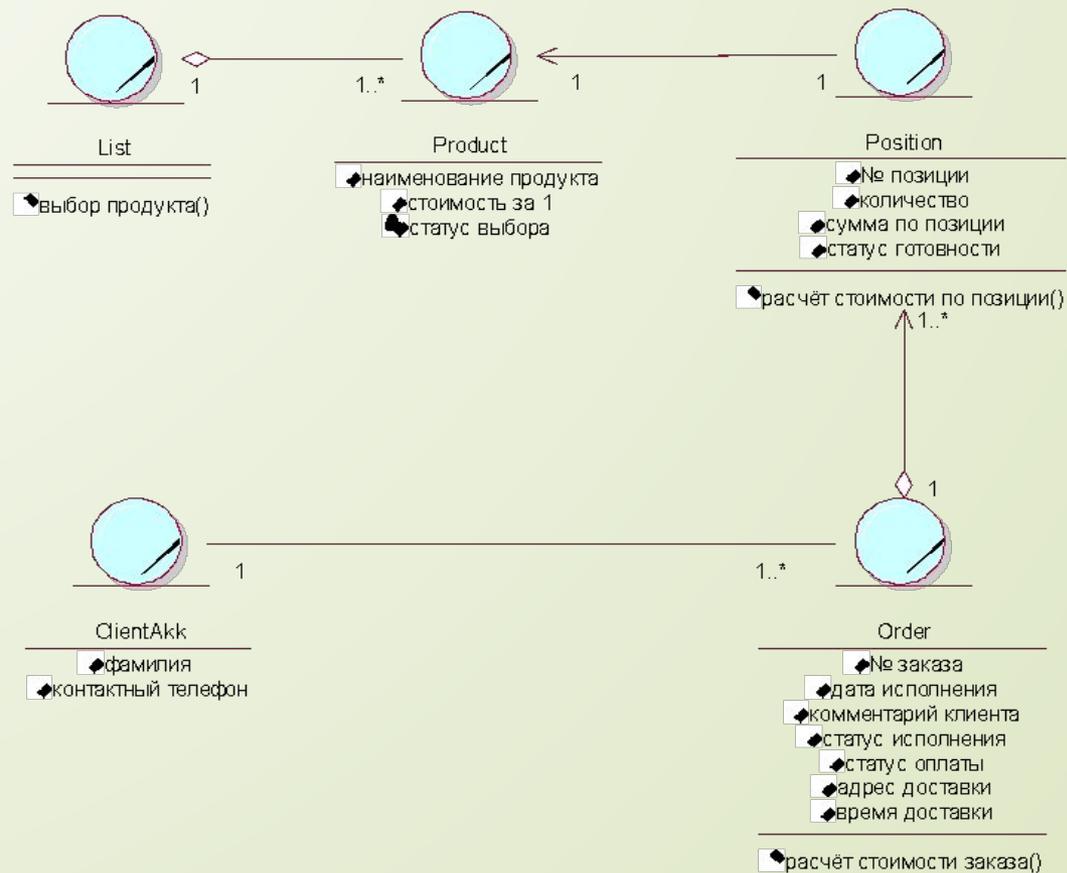
□ Модель сущность-связь (ERD)



Модели сущностей предметной области

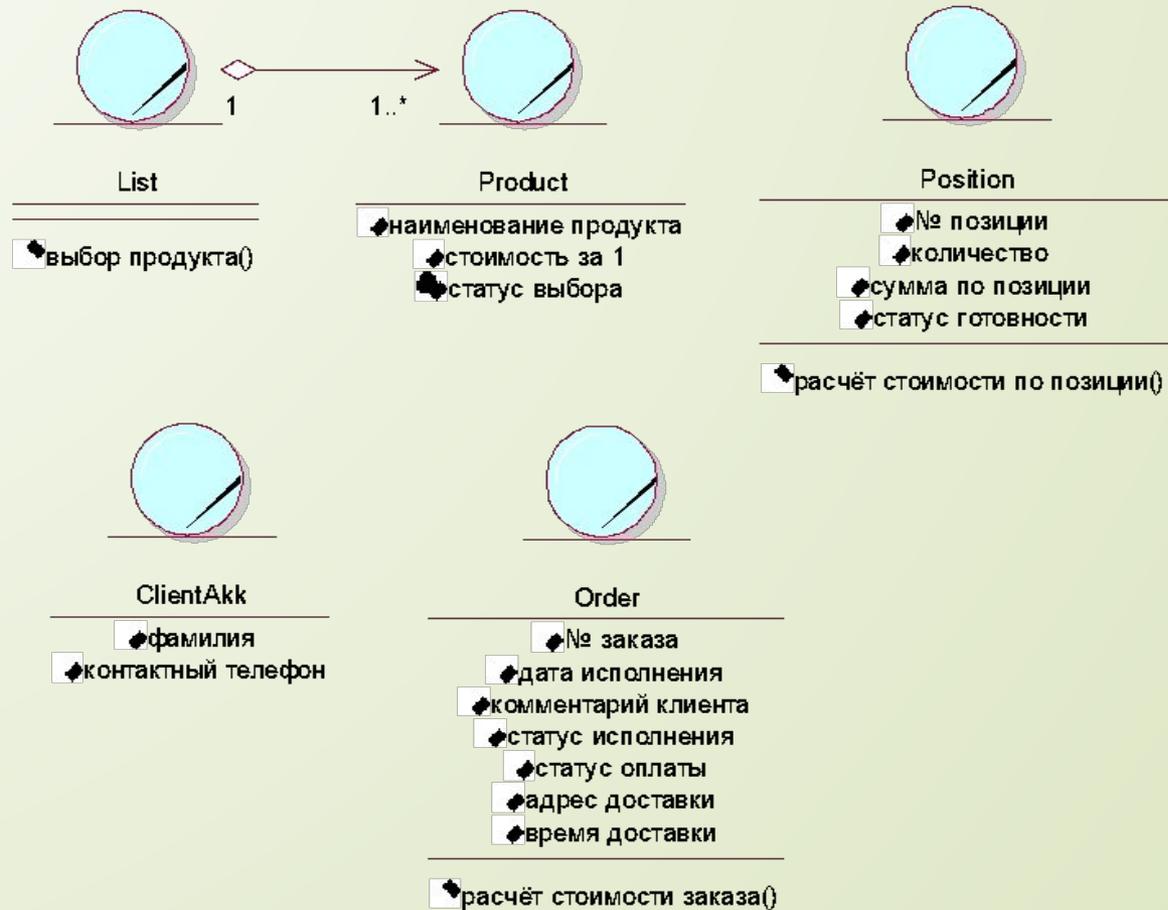
□ Модель классов

Диаграмма классов (class diagram)



Модели сущностей предметной области

□ Модель классов (без связей)



!СПАСИБО ЗА ВНИМАНИЕ

□ Контакты:

е-Mail: info@webmax.by

Skype: nousy123

Тел.: +375 (25) 633-76-78

Сайт: www.webmax.by