

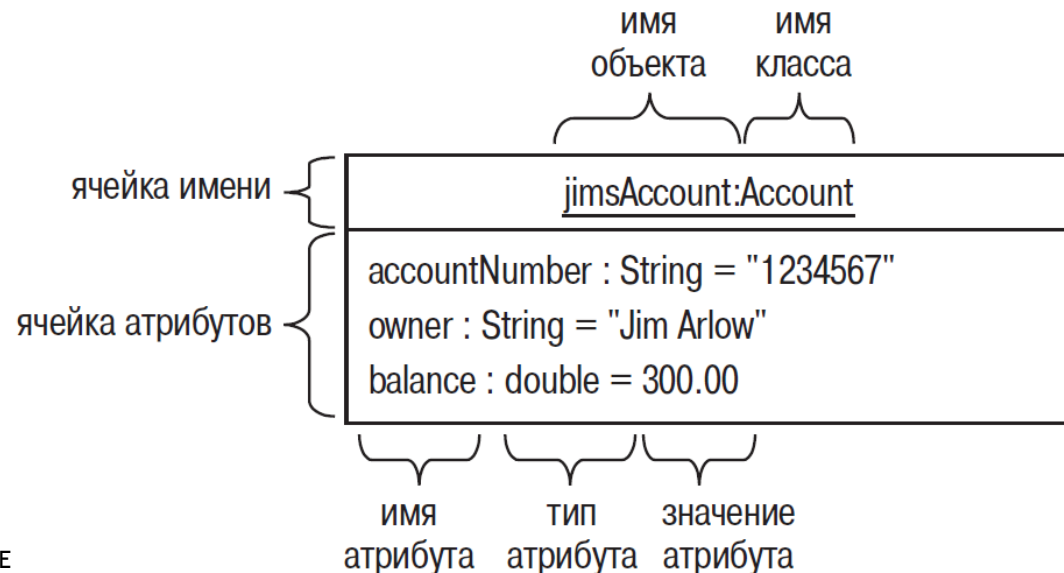
ПРОГРАММНАЯ ИНЖЕНЕРИЯ

Классы анализа

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ ПОСРЕДСТВОМ UML

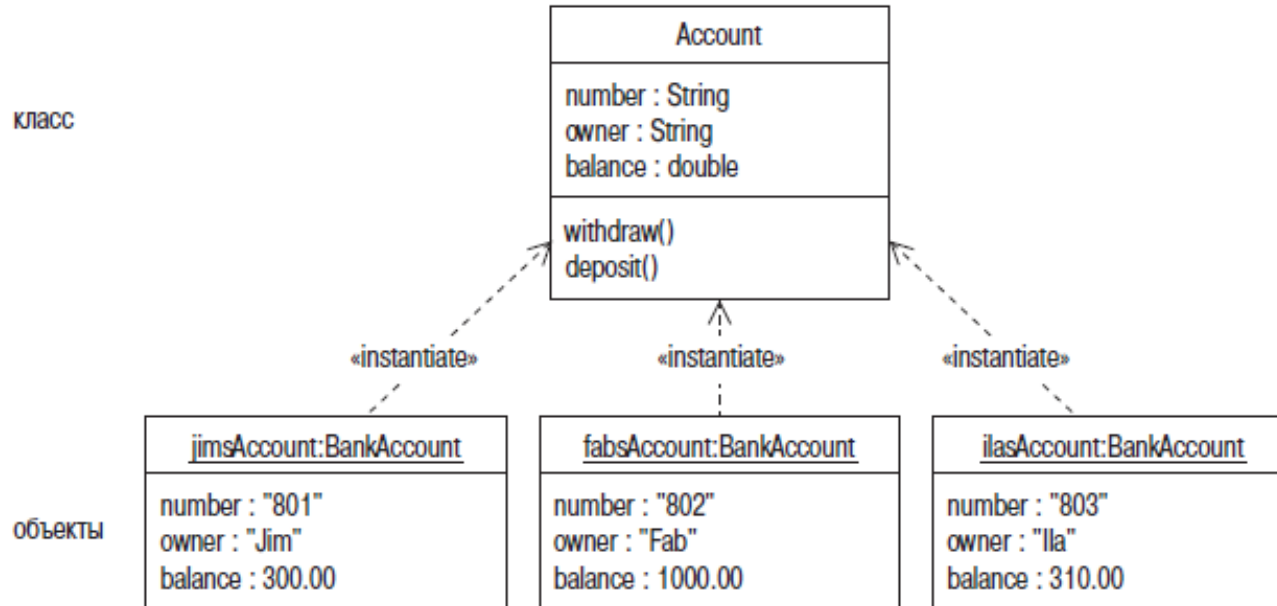
НОТАЦИЯ ОБЪЕКТОВ UML

- ◎ Прямоугольник с двумя ячейками:
 - ◎ Идентификатор объекта (подчеркнутый) и/или имя класса через двоеточие (может быть только имя класса, например «:Account» - анонимный объект)
 - ◎ Блок атрибутов (по выбору, т.к. набор атрибутов определяет класс)



КЛАССЫ И ОБЪЕКТЫ UML

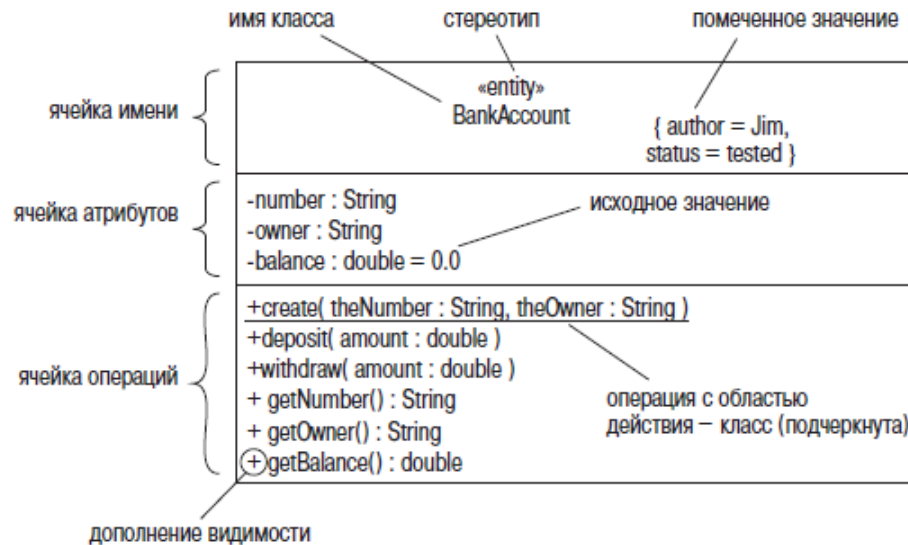
- Между классом и объектами этого класса устанавливается отношение «instantiate» (создать экземпляр)



- Отношение зависимости означает, что изменение сущности поставщика оказывает влияние на сущность клиент

НОТАЦИЯ КЛАССОВ UML

- Обязательной частью в визуальном синтаксисе является только ячейка с именем класса. Все остальные ячейки и дополнения необязательны.



- Если на диаграмме интерес представляют лишь отношения между различными классами, тогда может быть достаточно ячейки с именем.
- Если диаграмма призвана проиллюстрировать поведение классов, тогда, вероятно, в каждый класс будет добавлена ячейка с ключевыми операциями.

НОТАЦИЯ АТТРИБУТОВ КЛАССА

- ⦿ Единственная обязательная часть UML-синтаксиса для атрибута является его имя.

видимость имя : тип [множественность] = начальноеЗначение

/
обязателен

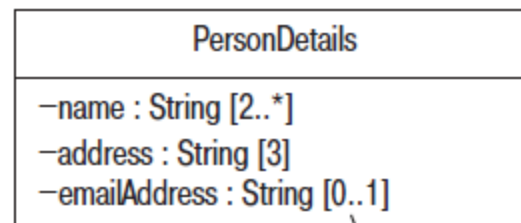
Видимость:

+	Public
-	Private
#	Protected
~	Package

Базовые типы:

Integer
UnlimitedNatural
Boolean
String
Real

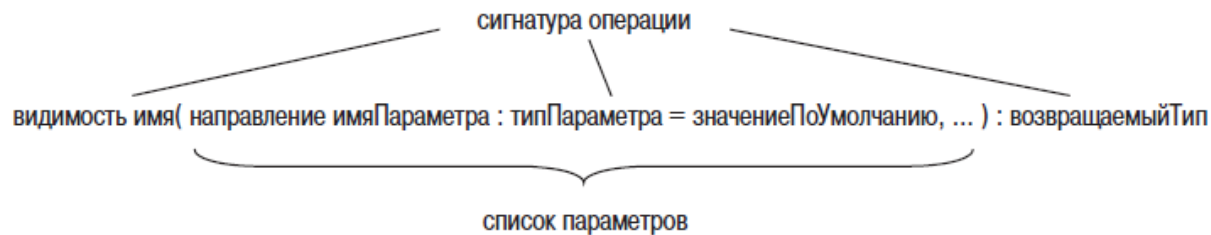
Множественность:



выражение кратности

НОТАЦИЯ ОПЕРАЦИЙ

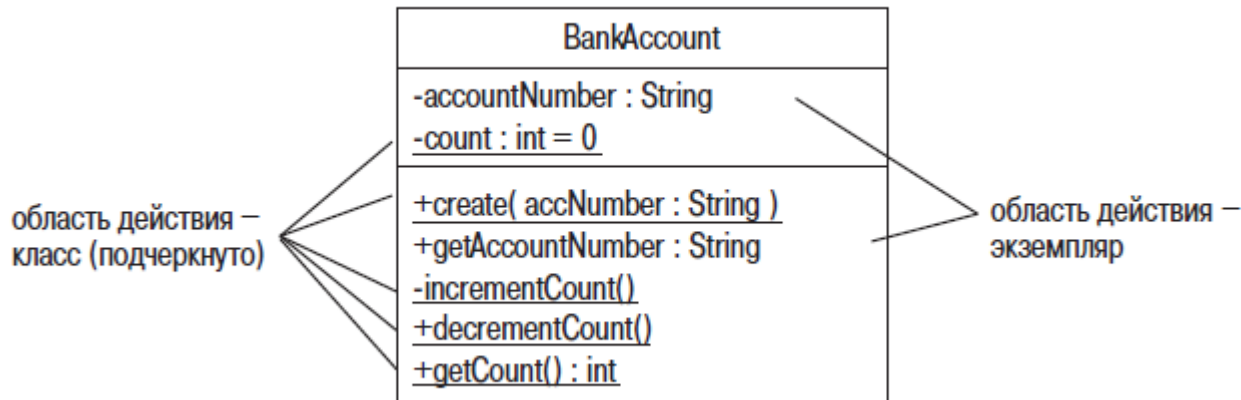
- ◎ Сигнатура операции включает имя, тип всех ее параметров и возвращаемый тип



- ◎ Направление параметра:
 - ◎ in (по умолчанию) – входной параметр
 - ◎ inout – параметр ввода/вывода
 - ◎ out – выходной параметр
 - ◎ return – возвращаемый параметр (UML 2.1 – только 1)

НОТАЦИЯ ОБЛАСТИ ДЕЙСТВИЯ

- Иногда нужно определить атрибуты, которые имеют единственное, общее для всех объектов класса значение. И нужны операции (как операции создания объектов), не относящиеся ни к одному конкретному экземпляру класса. Говорят, что такие атрибуты и операции имеют *область действия – класс*.



- В аналитических моделях обычно не занимаются конструкторами (и тем более деструкторами). Они не имеют ни влияния, ни отношения к бизнес-семантике класса. Если все-таки есть желание обозначить операции создания, можно ввести операцию `create()` без параметров как структурный ноль.

КЛАССЫ АНАЛИЗА

АНАЛИЗ ПРЕЦЕДЕНТА

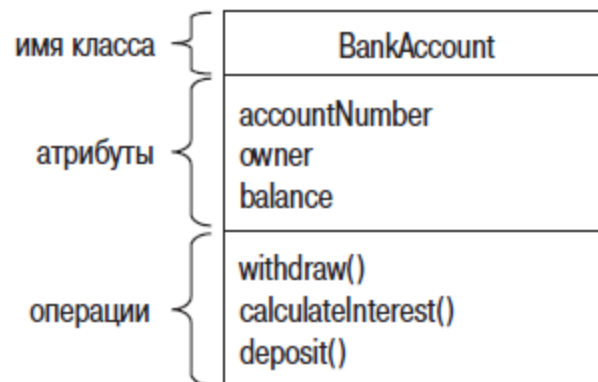
- ◎ Деятельность UP «**Анализ прецедента**» включает:
 - ◎ создание классов анализа
 - ◎ реализации прецедентов
- ◎ **Классы анализа** – это классы, которые представляют четкую абстракцию предметной области и должны проецироваться на реальные бизнес-понятия;
- ◎ **Реализации прецедентов** – это кооперации объектов, показывающие, как системы взаимодействующих объектов могут реализовывать поведение системы, описанное в прецеденте.

КЛАСС АНАЛИЗА

- ⊙ Класс анализа должен четко и однозначно проецироваться в реальное прикладное понятие.
- ⊙ Следовательно, задача ОО аналитика – попытаться прояснить беспорядочные или несоответствующие прикладные понятия и превратить их в то, что может стать основой для класса анализа.
- ⊙ Первый шаг в построении ОО программного обеспечения – прояснить предметную область. Если она содержит четко определенные бизнес-понятия и имеет простую функциональную структуру, решение практически лежит на поверхности.
- ⊙ Важно, чтобы *все классы аналитической модели являлись классами* анализа, а не классами, вытекающими из проектных соображений (области решения). Когда дело дойдет до детального проекта, может случиться так, что классы анализа будут в конце концов переработаны в один или более проектных классов. (Пример с курсовой работой и разницей описания систем).

СОСТАВ КЛАССА АНАЛИЗА

- ⊙ В классах анализа содержатся только ключевые атрибуты и обязанности, определенные на очень высоком уровне абстракции. *Указывают атрибуты, которые, возможно, будут присутствовать в проектных классах.*
- ⊙ Однако одна операция уровня анализа очень часто разбивается на несколько операций уровня проекта.



ПРИЗНАКИ ХОРОШЕГО КЛАССА АНАЛИЗА

- ◎ его имя отражает его назначение
- ◎ он является четкой абстракцией, моделирующей один конкретный элемент предметной области
- ◎ у него небольшой четко определенный набор обязанностей
- ◎ у него высокая внутренняя связность (cohesion)
- ◎ у него низкая связанность с другими классами (coupling)

ПОДХОДЫ К СОЗДАНИЮ КЛАССОВ АНАЛИЗА

- ◎ 3-5 методов в классе – оптимальное количество
- ◎ Классы не должны быть изолированными
- ◎ Не создавать много мелких классов (1-2 метода)
- ◎ Не объединять все в крупные классы (> 6 методов)
- ◎ Избегать всемогущих классов («...System»
«...Controller»)
- ◎ Избегать глубокой иерархии наследования

ВЫЯВЛЕНИЕ КЛАССОВ АНАЛИЗА

Нет универсального подхода, дающего 100% результат, но существуют проверенные методы:

- ◎ Метод «Существительное/Глагол»
- ◎ Метод CRC-анализа
- ◎ Применение стереотипов RUP
- ◎ Применение готовых шаблонов классов анализа

СУЩЕСТВИТЕЛЬНОЕ / ГЛАГОЛ

- ⊙ Анализируются: модель требований; модель прецедентов; документы с постановкой задачи.
- ⊙ Существительные и именные группы указывают на классы или атрибуты. Глаголы и глагольные группы служат признаком обязанностей или операций.
- ⊙ Результат – список с потенциальными классами, атрибутами и обязанностями (методами). После чего – распределение атрибутов и методов по классам.




CRC-АНАЛИЗ

- ◎ CRC – class-responsibilities-collaborators (класс – обязанности – участники)
- ◎ CRC-анализ должен всегда использоваться в сочетании с анализом существительное/глагол прецедентов, требований, глоссария и другой относящейся к делу документации. Основная идея – рассортировать данные, *поступающие в результате анализа информации.*
- ◎ CRC – это техника мозгового штурма, при которой важные моменты предметной области записываются на стикерах.

Имя класса: BankAccount	
Обязанности: поддерживать остаток	Участники: Bank

СТЕРЕОТИПЫ RUP

- ◎ Согласно RUP считается полезным поискать классы, которые можно обозначить стереотипами «**boundary**» (граница), «**control**» (управление) и «**entity**» (сущность).

Стереотип	Пиктограмма	Семантика
«boundary»		Класс, который является посредником во взаимодействии между системой и ее окружением.
«control»		Класс, инкапсулирующий характерное для прецедента поведение.
«entity»		Класс, используемый для моделирования постоянной информации о чем-то.

ШАБЛОНЫ КЛАССОВ АНАЛИЗА

- ◎ Базовые шаблоны (паттерны) могут предоставить готовые компоненты аналитической модели.
- ◎ Есть несколько книг, в которых приводятся примеры таких паттернов:
 - ◎ «Enterprise Patterns and MDA: Building Better Software with Archetype Patterns and UML», Jim Arlow, Ila Neustadt, AddisonWesley, 2004
 - ◎ Архитектура корпоративных программных приложений. Мартин Фаулер. Вильямс, 2007 г.
- ◎ Шаблоны Order, Party, Product, Rule ...