



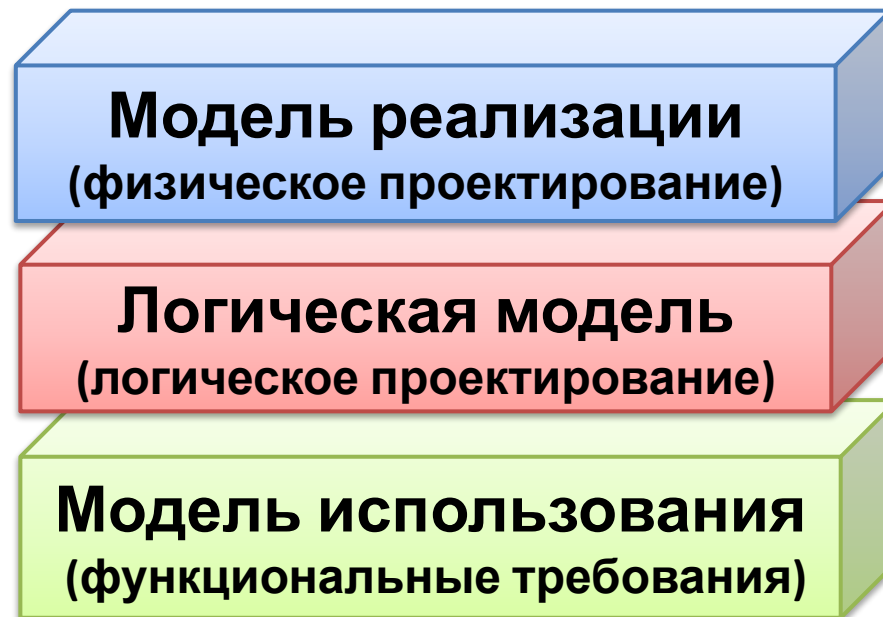
ОБЪЕКТНО- ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ ИС (ЛОГИЧЕСКАЯ МОДЕЛЬ)

д.т.н. Емельянов Виталий Александрович

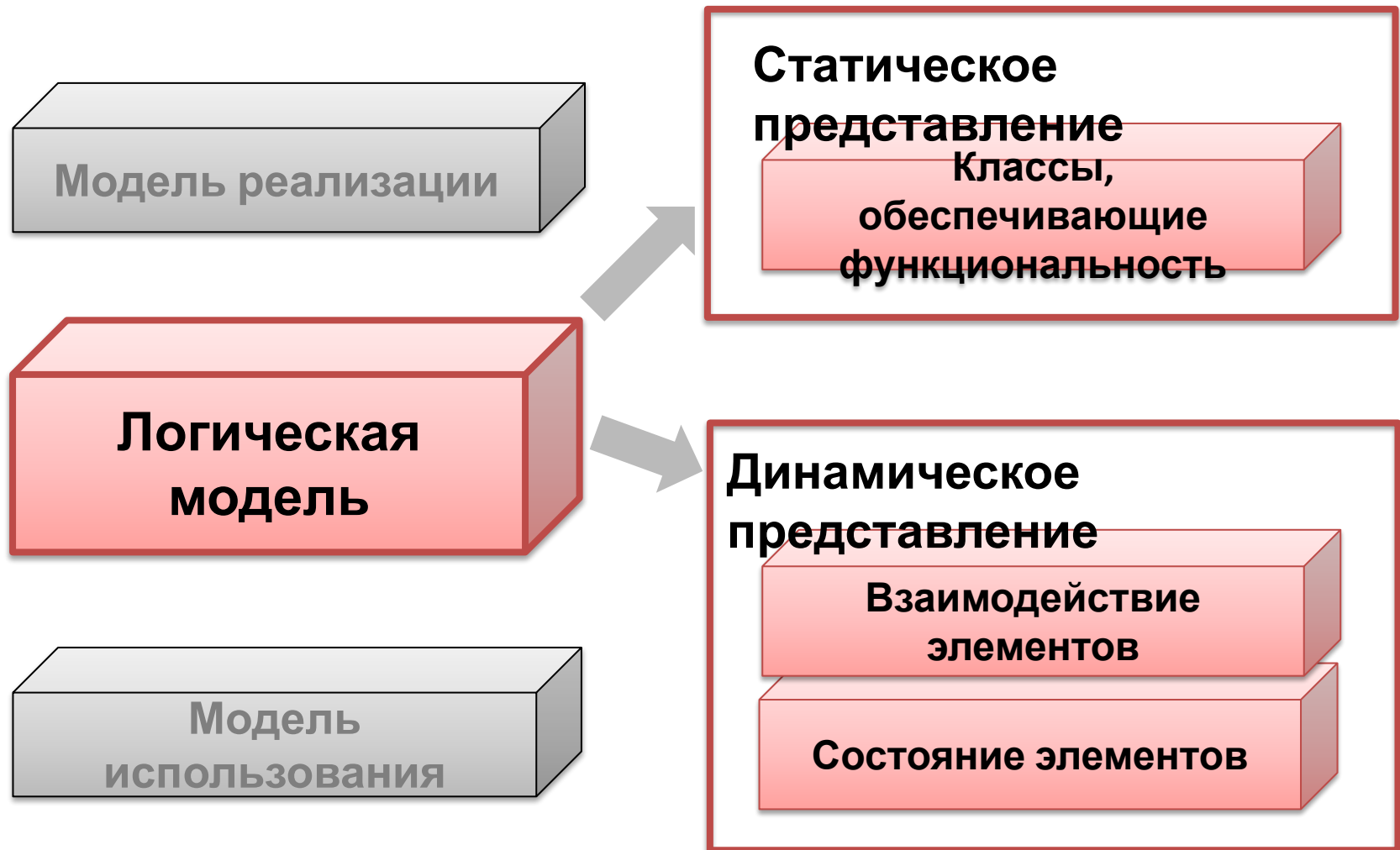


: v.yemelyanov@gmail.com

Объектно-ориентированное проектирование ИС



Объектно-ориентированное проектирование ИС



Логическая модель

Статическое представление

(Логическая структура ИС или ПО)

Классы,
обеспечивающие
функциональность

Диаграмма классов
(Class diagram)

Динамическое представление

(Описание поведения программы, взаимодействие объектов между собой и с актерами с целью достижения заданной функциональности)

Взаимодействие элементов

Диаграмма кооперации
(Collaboration diagram)

Диаграмма последовательности
(Sequence diagram)

Диаграмма деятельности
(Activity diagram)

Состояние элементов

Диаграмма состояний
(Statechart diagram)

Логическая модель (статическое представление)

Статическое представление

(Логическая структура ИС или ПО)

Классы,
обеспечивающие
функциональность

Диаграмма классов
(Class diagram)

Динамическое представление

(Описание поведения программы, взаимодействие объектов между собой и с актерами с целью достижения заданной функциональности)

Взаимодействие элементов

Состояние элементов

Диаграмма кооперации
(Collaboration diagram)

Диаграмма последовательности
(Sequence diagram)

Диаграмма деятельности
(Activity diagram)

Диаграмма состояний
(Statechart diagram)

Диаграммы классов (Class diagram)

Применение:

- ❑ **на этапе анализа** — для указания ролей и обязанностей сущностей, которые обеспечивают поведение системы
- ❑ **на этапе проектирования** — для фиксации структуры классов, которые формируют системную архитектуру и реализуют функции системы

Диаграммы классов (Class diagram)

Сущности, отражаемые на диаграмме:

- ❑ «Пакеты» или «подсистемы»
- ❑ «Классы»
- ❑ «Интерфейсы»
- ❑ «Отношения»: зависимости, обобщения и ассоциации

Диаграммы классов (Class diagram)

Декомпозиция системы на подсистемы:

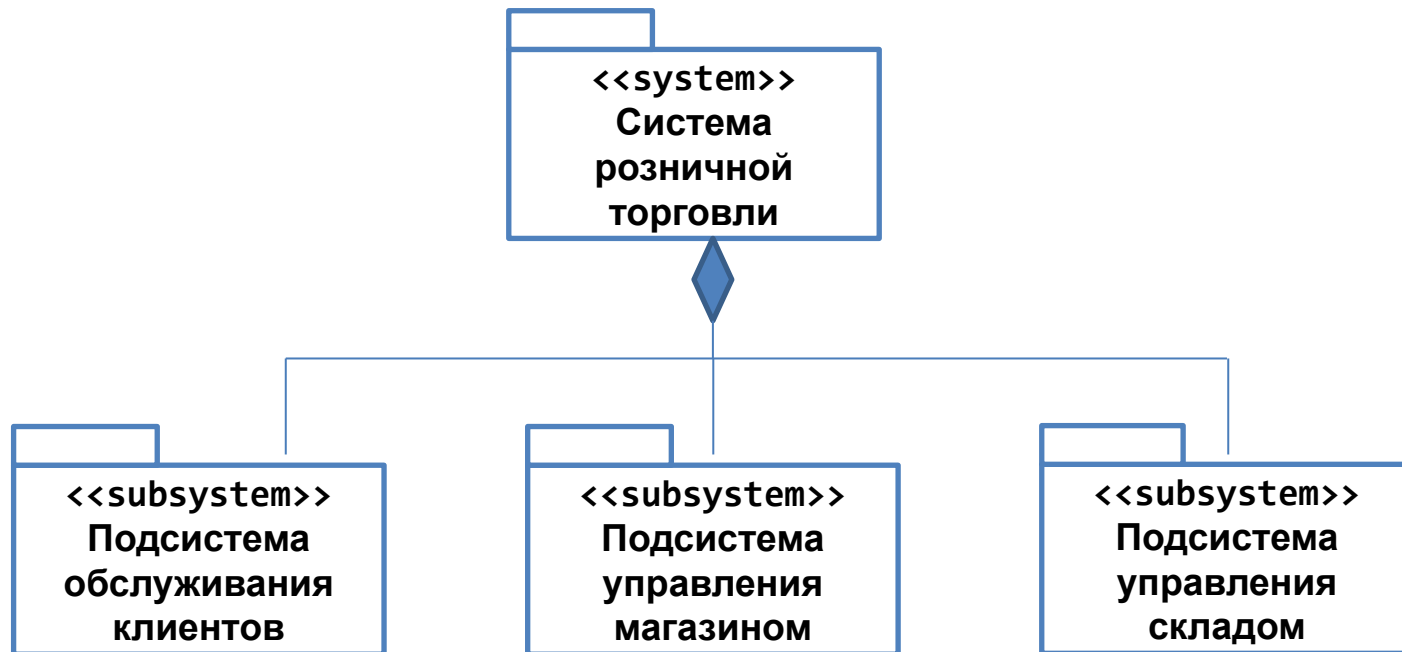
- Изображаются система и подсистема в виде пиктограммы стереотипного пакета (Package) - представляет собой общий механизм организации элементов в группы:



- Основное отношение между системами и подсистемами - это агрегирование - отношение типа "часть/целое", в котором один из классов имеет более высокий ранг (целое) и состоит из нескольких меньших по рангу (частей):



Диаграммы классов (Class diagram)

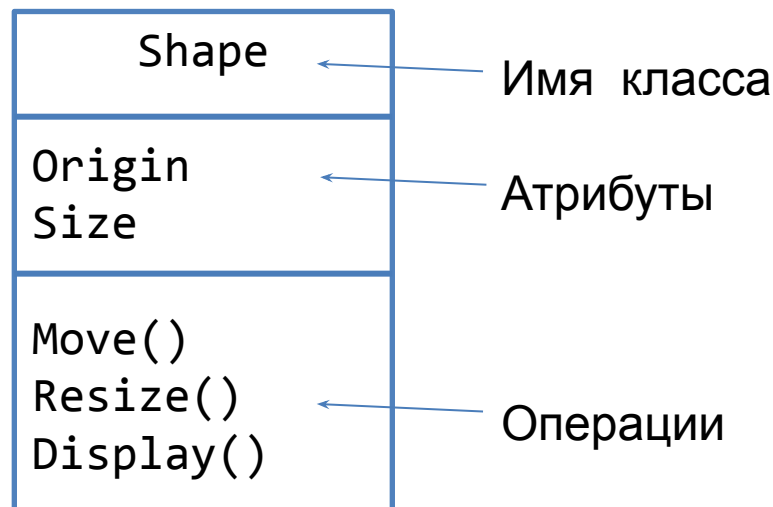


Диаграммы классов (Class diagram)

Сущность «Класс»:

- описание совокупности объектов с общими атрибутами, операциями, отношениями и семантикой. Графически класс изображается в виде прямоугольника.

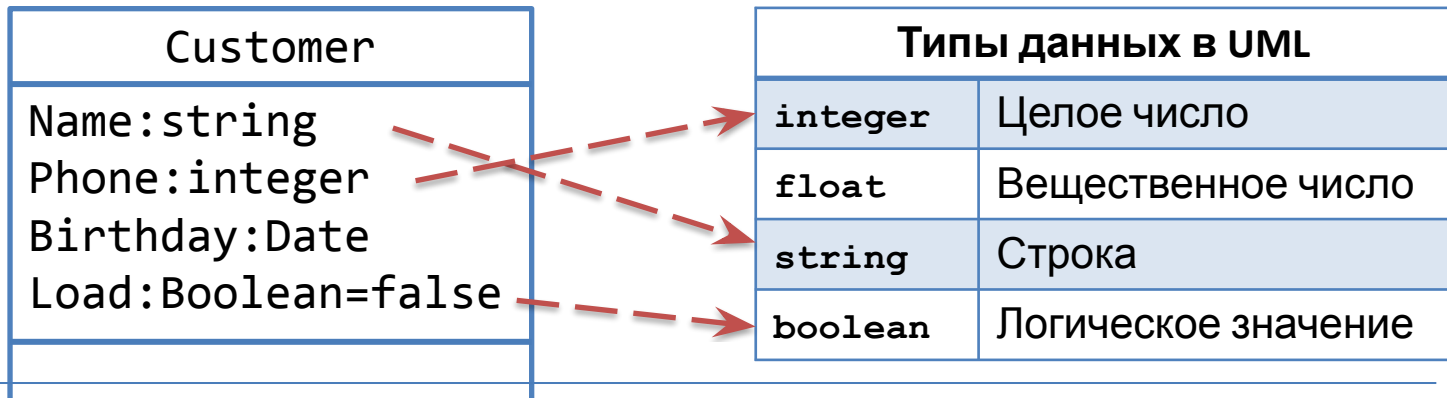
Условное графические обозначение:



Диаграммы классов (Class diagram)

Сущность «Класс»:

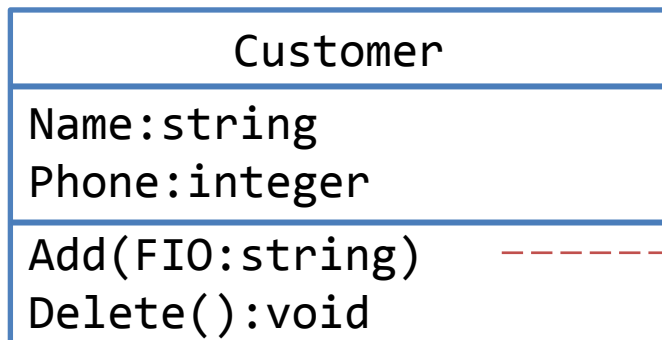
- **Атрибуты** соответствуют характеристикам объектов. **Значение атрибута определяет состояние объекта.**
- Класс может иметь любое число атрибутов или не иметь их вовсе.
- При описании атрибута можно явным образом указывать его класс и начальное значение, принимаемое по умолчанию



Диаграммы классов (Class diagram)

Сущность «Класс»:

- ❑ **Операция** - реализация функции (действия), которую можно запросить у любого объекта класса для воздействия на поведение.
- ❑ Операцию можно описать более подробно, указав ее сигнатуру, в которую входят имена и типы всех параметров, их значения, принятые по умолчанию, а применительно к функциям - тип возвращаемого значения. **ПРИМЕР:**

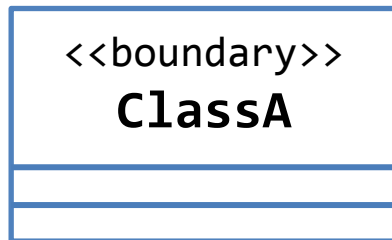


Логика выполнения операции
раскрывается через диаграммы
деятельности или состояний

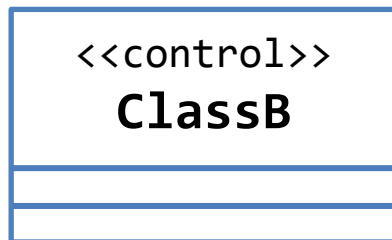
Диаграммы классов (Class diagram)

Сущность «Класс»:

- **Стереотип класса** - отражает назначение класса относительно других классов или внешней среды



Граничный класс. Располагается на границе системы и внешней среды. Является частью системы и отвечает за взаимодействие с внешней средой (актерами)



Управляющий класс. Отвечает за координацию действий других классов. Это активный класс, выполняющий рассылку сообщений другим классам. Диаграмма классов должна содержать как минимум один управляющий класс.

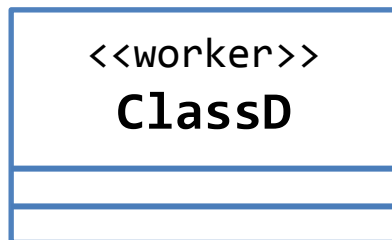
Диаграммы классов (Class diagram)

Сущность «Класс»:

- **Стереотип класса** - отражает назначение класса относительно других классов или внешней среды



Класс-сущность. Определяет информацию, которая должна храниться постоянно и не должна уничтожаться при выключении системы (**пример – база данных**). Это пассивный класс, принимающий сообщения от других классов. Обычно содержит только атрибуты.

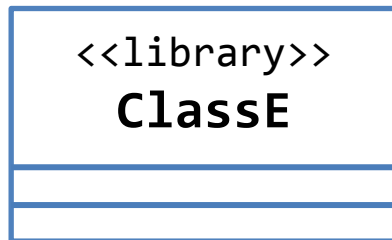


Класс-сотрудник. Определяет сотрудника, который является элементом системы и взаимодействует с другими сотрудниками при реализации вариантов использования

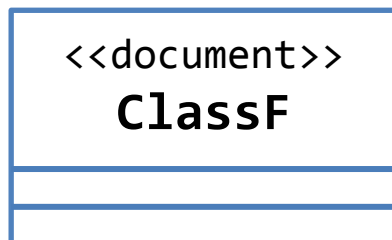
Диаграммы классов (Class diagram)

Сущность «Класс»:

- **Стереотип класса** - отражает назначение класса относительно других классов или внешней среды



Библиотека классов. Определяет классы, динамически компонуемые с основной программой в процессе ее выполнения.



Класс-документ. Определяет класс, не содержащий программного кода. **Примеры:**

- файл с исходными данными;
- файл с результатами работы программы;
- файл справочной информации по программе и т.п.

Диаграммы классов (Class diagram)

Сущность «Класс»:

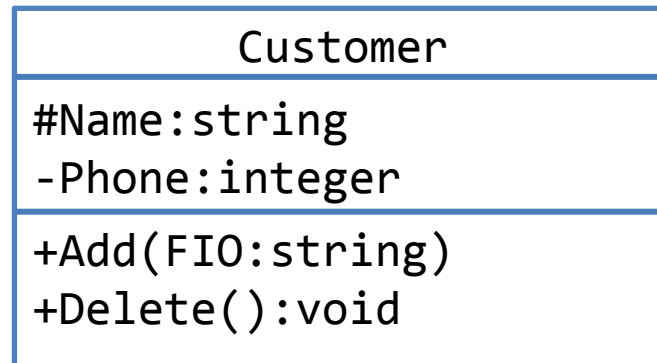
- Уровни видимости атрибутов и операций:

Обозначение квантора	Подпись	Смысл
+	public	Доступен для прямого доступа из любых классов
–	private	Не доступен для прямого доступа из других классов
#	protected	Доступен для прямого доступа только из классов-наследников
~	package	Доступен для прямого доступа для любых классов, входящих в пакет

Диаграммы классов (Class diagram)

Сущность «Класс»:

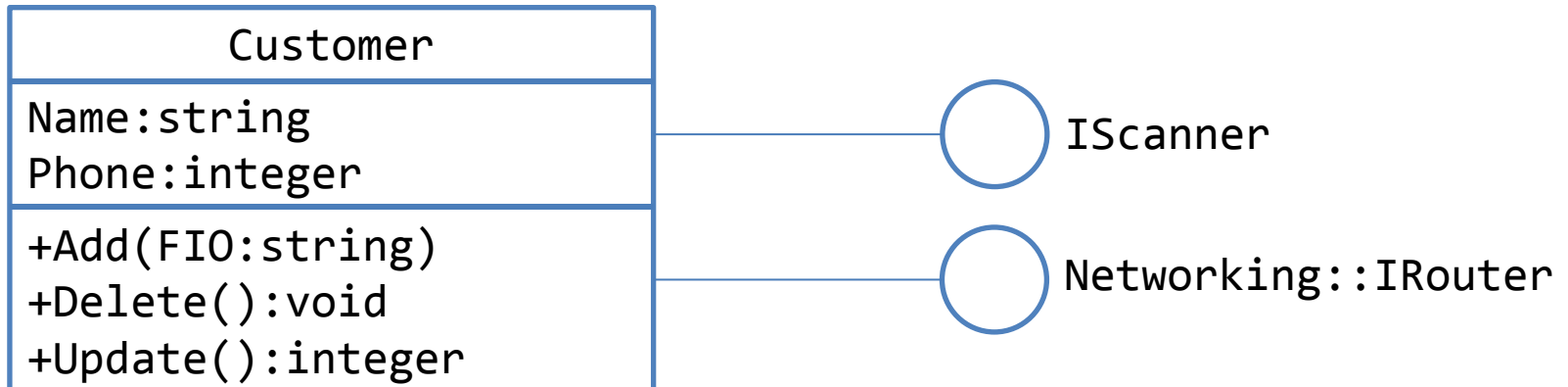
- **Уровни видимости атрибутов и операций:**
 - Отсутствие квантора видимости на диаграмме классов означает, что видимость атрибута не указана
 - Отсутствие квантора видимости в программе означает квантор видимости, принятый по умолчанию (в языке C# это квантор `private`)



Диаграммы классов (Class diagram)

Сущность «Интерфейс»:

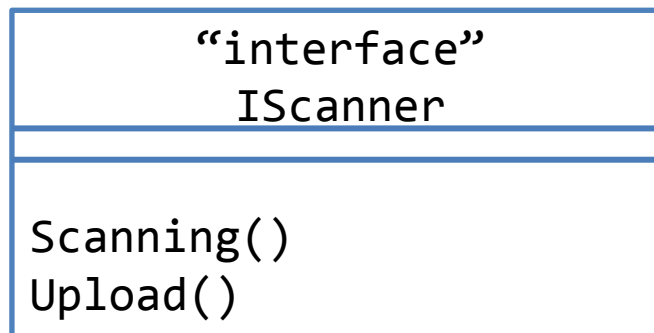
- ❑ **Интерфейс** - это набор операций, используемых для специфицирования услуг, предоставляемых классом или компонентом
- ❑ Интерфейсы используются для визуализации, специфицирования, конструирования и документирования стыковочных узлов между составными частями системы.
- ❑ Имя интерфейса представляет собой текстовую строку. Взятое само по себе, оно называется простым. Составное имя образуется путем добавления в его начало имени пакета, в который входит данный интерфейс. **ПРИМЕР:**



Диаграммы классов (Class diagram)

Сущность «Интерфейс»:

- В отличие от классов, интерфейсы не описывают структуры (поэтому не могут содержать атрибуты) и реализации (а значит, не содержат реализующих операций методов).
- Если это необходимо для понимания модели, можно изобразить интерфейс как стереотипный класс, перечислив операции в соответствующем разделе. **ПРИМЕР:**










Диаграммы классов (Class diagram)

Сущность «Интерфейс»:

- Интерфейс специфицирует контракт класса или компонента, но не накладывает никаких ограничений на реализацию.
- Класс или компонент могут реализовывать несколько интерфейсов. При этом они обязуются выполнить все свои контракты, то есть должны содержать методы, корректно реализующие объявленные интерфейсом операции.

Диаграммы классов (Class diagram)

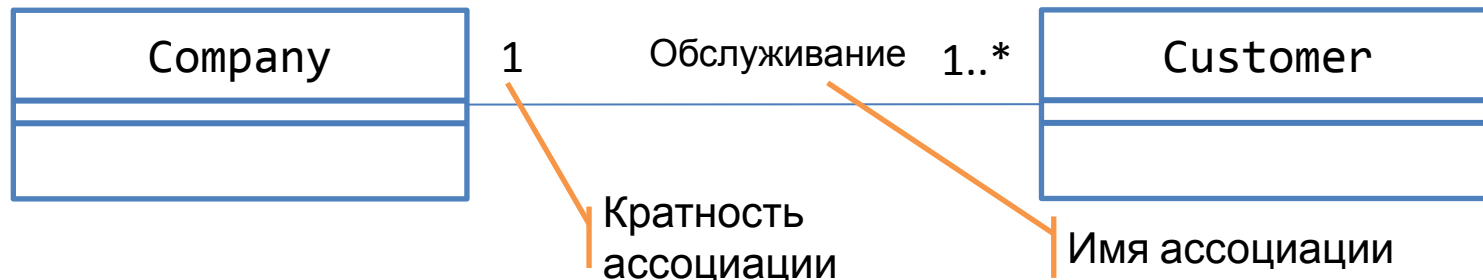
Сущность «Отношения»:

- ❑ **Ассоциация:**  или 
- ❑ **Обобщение:**  Подкласс —————> Суперкласс
- ❑ **Зависимость:**  Зависимый элемент —————> Независимый элемент
- ❑ **Реализация:**  Приемник —————> Источник
- ❑ **Агрегация:**  Целое — Часть
- ❑ **Композиция:**  Целое — Часть
(Физическое включение)

Диаграммы классов (Class diagram)

Отношение «Ассоциация»:

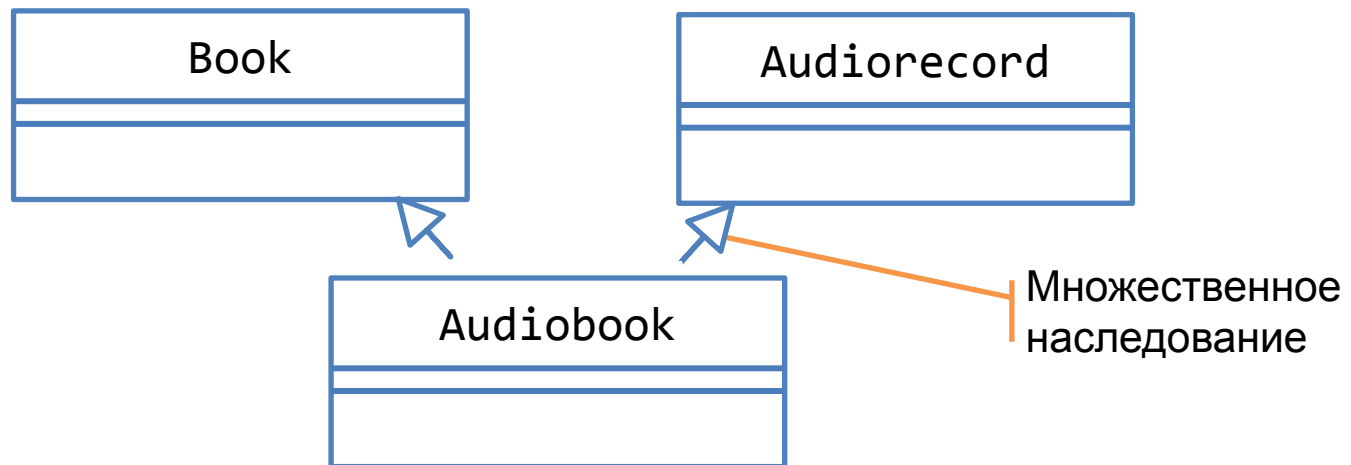
- **Ассоциация означает**, что классы или объекты классов взаимодействуют между собой при реализации варианта/вариантов использования
- **Имя ассоциации** (Name) должно отражать вариант использования или совокупность вариантов использования
- **Кратность ассоциации** определяет количество объектов каждого класса, участвующих в ассоциации.



Диаграммы классов (Class diagram)

Отношение «Обобщение (наследование)»:

- Раскрывает структуру предметной области по принципу “предок-потомок”
- Класс определяется путем указания отличий от базового класса. Определяемый класс может отличаться от базового класса дополнительными атрибутами и/или операциями.



Диаграммы классов (Class diagram)

Отношение «Реализация»:

- Отношение реализации соответствует ситуации, при которой класс-интерфейс объявляет спецификацию операций, а классы общего назначения могут взять на себя обязанность реализовать операции, объявленные в интерфейсе

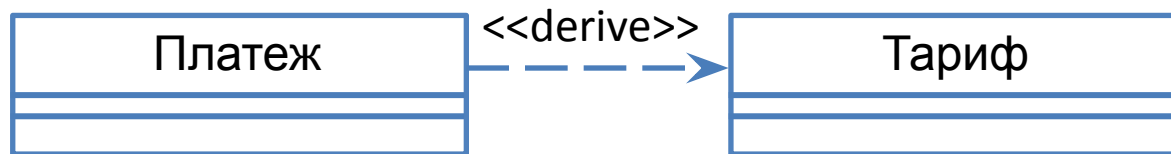


Класс **Catalog** обязан реализовать операции **Search()** и **Update()**

Диаграммы классов (Class diagram)

Отношение «Зависимость»:

- Определяет зависимость одного класса от другого класса.
- Характер зависимости указывается в виде стереотипа зависимости



Независимый класс (Тариф) определяет данные, на основании которых вычисляется значение атрибутов в зависимом классе (Платеж)

Диаграммы классов (Class diagram)

Подходы к выявлению классов предметной области:

Классический

Классы в модели соответствуют реально существующим объектам

Экспертный

Классы в модели назначаются экспертом

**Неформальное
описание**

Описание ПрО подвергается анализу:

1. Существительные рассматриваются как кандидаты на объект или атрибут.
2. Глаголы - кандидаты на операции.
3. Условия - кандидаты на сценарий использования

Поведение

Класс образуют объекты с общим поведением (все, что плавает, все, что летает и т.д.)

**Варианты
использования**

Классы выявляются путем анализа функций, выполняемых в интересах пользователя.

Диаграммы классов (Class diagram)

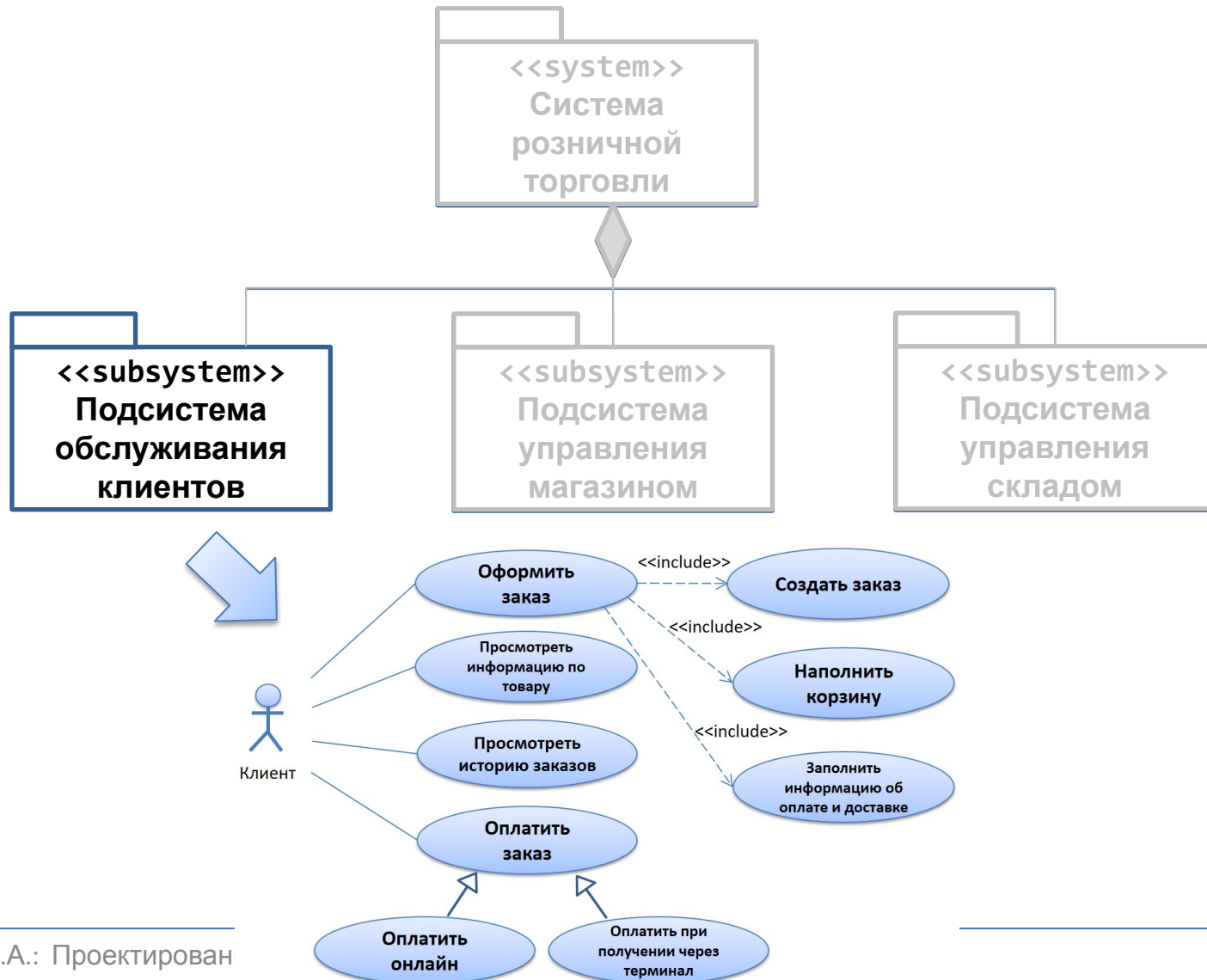
Описание ПрО:

Компания «СУПЕР МАГАЗИН» является сетью магазинов розничной торговли (7 магазинов). У компании есть свой сайт, где представлена информация по продаваемым товарам. Но сайт является только информационным и не позволяет проводить покупку товаров онлайн. В компании отсутствует возможность оперативного отслеживания изменения количества товаров в наличии в магазинах и на складе.

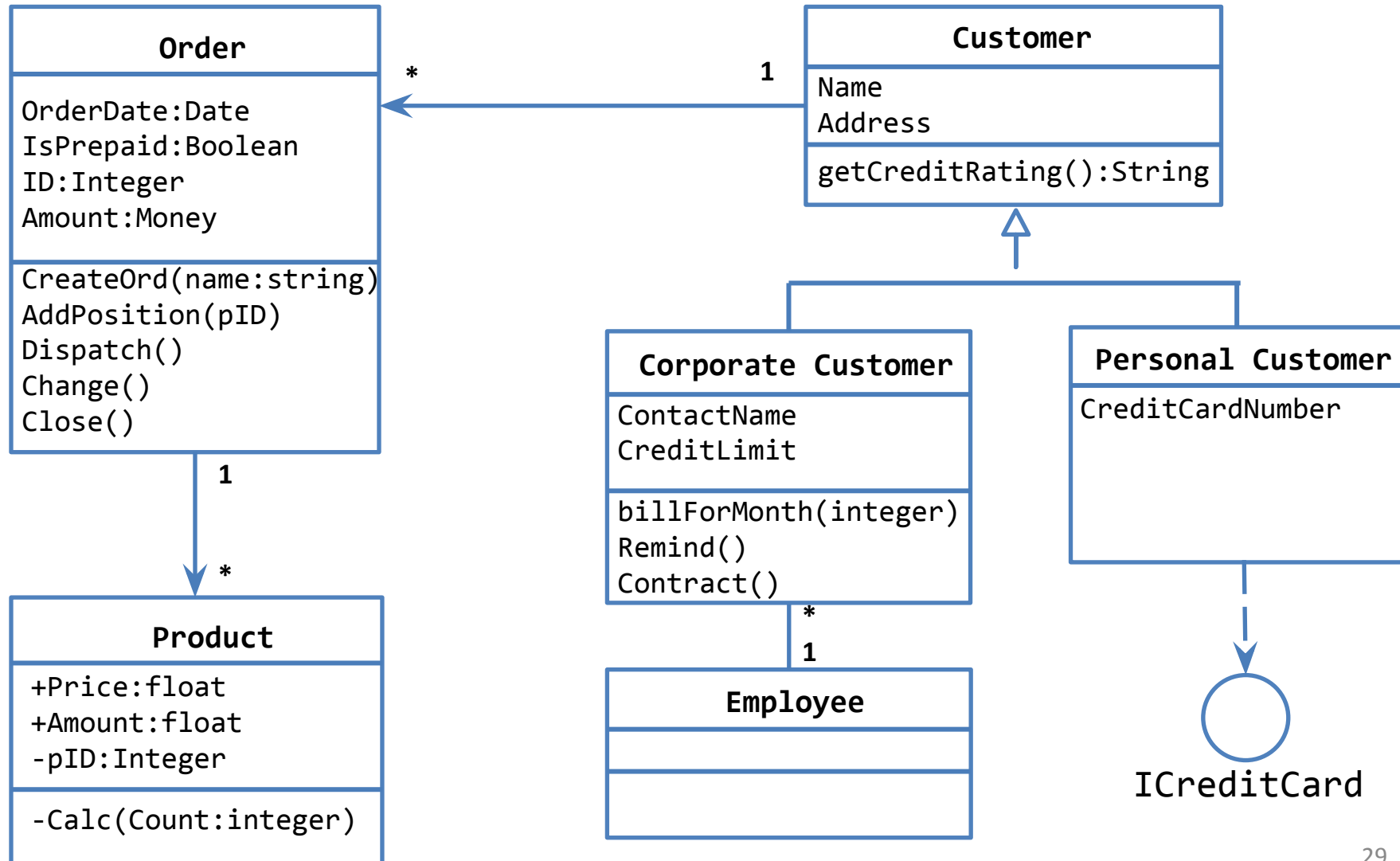
Компании «СУПЕР МАГАЗИН» требуется информационная система розничной торговли, которая должна автоматизировать следующие процессы:

- 1. Обслуживание клиентов, включая оформление и оплату заказов для юридических и физических лиц онлайн;**
2. Управление сетью ее магазинов;
3. Управление запасами и складом для сети магазинов.

Диаграммы классов (Class diagram)



Диаграммы классов (Class diagram)



Диаграммы классов (Class diagram)

