




УНИВЕРСИТЕТ

искусственного интеллекта

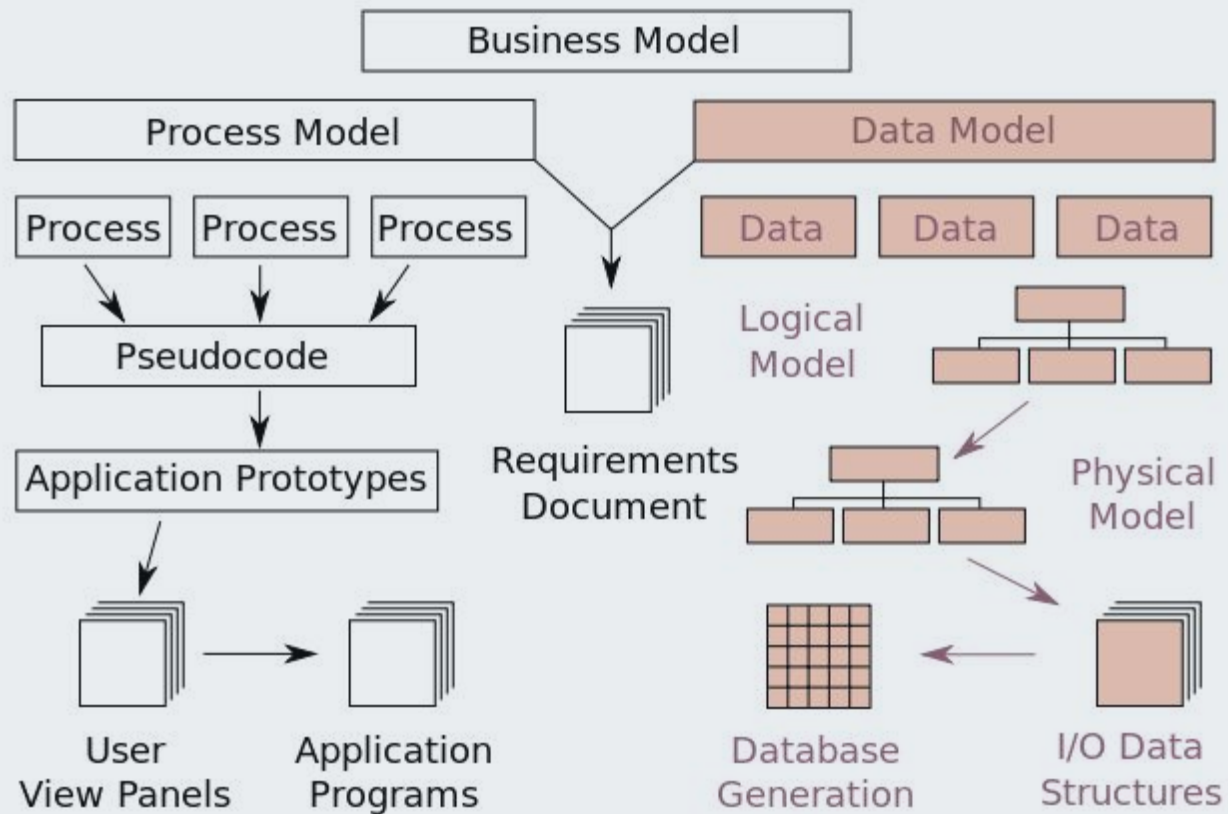
МОДЕЛЬ ДАННЫХ



ПЛАН

- Введение в моделирование данных
 - Entity Relation Diagram
 - 3 Уровня модели данных
 - UML диаграммы
- 

Введение в моделирование данных

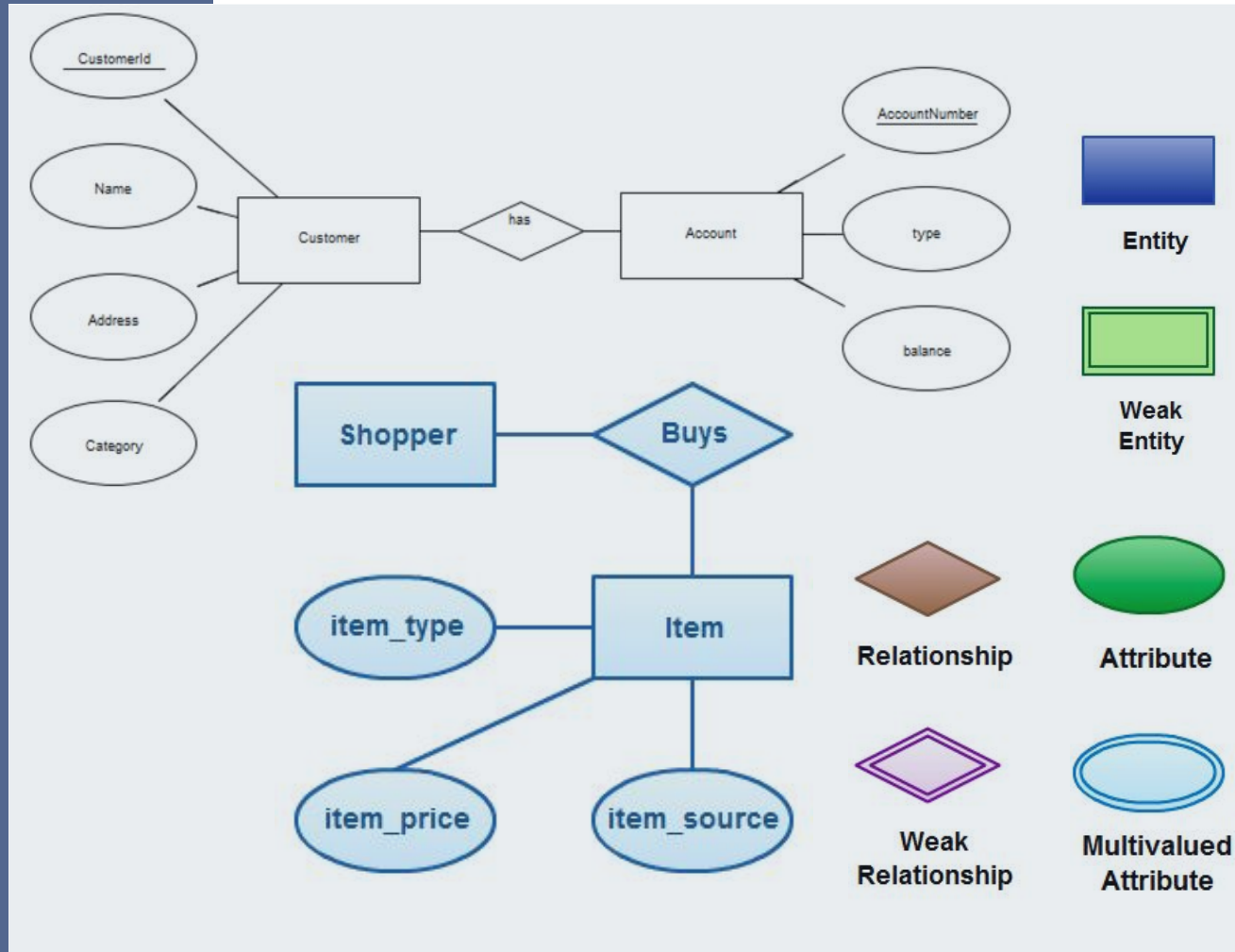


Моделирование данных используется для анализа и определения требований к данным для организации бизнес процесса, в задаче проектирования информационных систем.

Для описания модели данных используются специальные языки:

- Entity-relationship diagram (ERD)
- UML

Entity Relation Diagram



Entity Relationship Diagram (ERD) представляет собой визуальное представление различных сущностей в системе и как они относятся друг к другу.

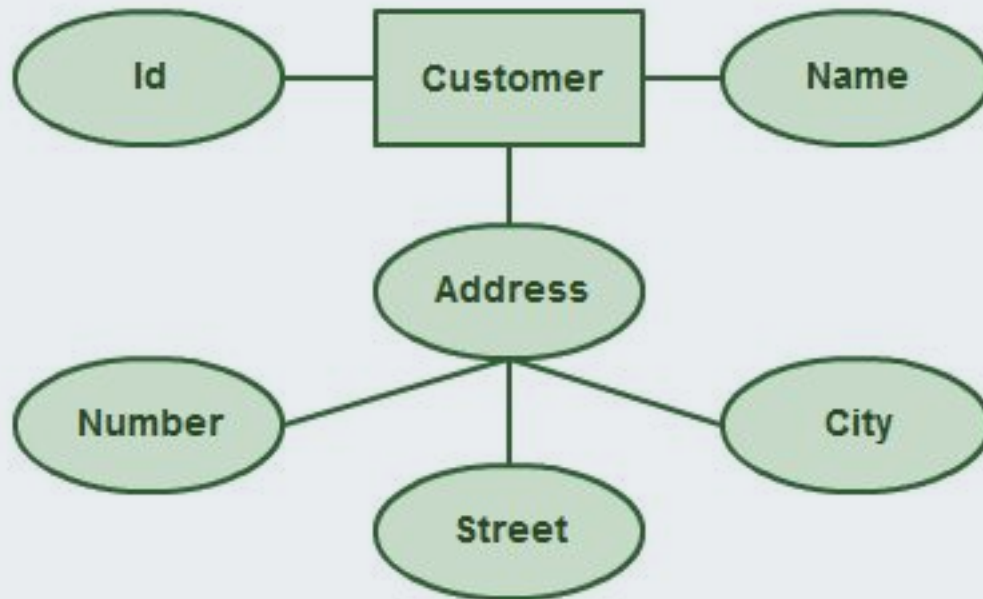
Основные Элементы Диаграммы

- Entity: независимый индивидуальный элемент (сущность) системы выражающий объект (покупатель, учитель, дом) или концепцию (транзакции, отзывы).
- Entity type: категория определяющая одну группу сущностей.
- Relationship: Отражает зависимость между Entity (сущностями) системы.
- Attribute: описывает Entity или Relationship, представляет собой отдельную часть информации

Entity Relation Diagram: Attribute

Атрибут представляет собой свойства, особенность или характеристику Entity, Relationship или другого атрибута.

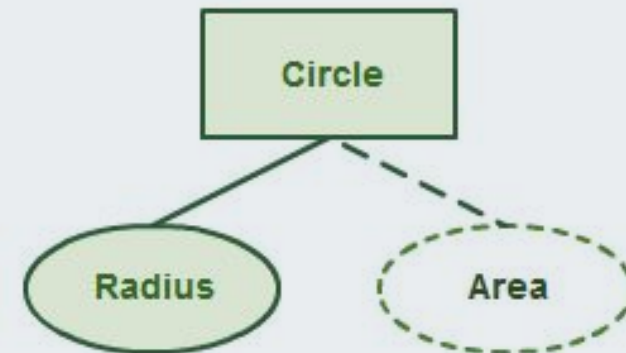
Атрибут так же может иметь вложенные атрибуты (составной атрибут): "customer address" имеет вложенные атрибуты: number, street, city.



Множественные атрибуты:



Производные атрибуты:



Entity Relation Diagram: Relationship

Relationship описывает взаимодействие между Entity, например сущность “Carpenter” (столяр) может относиться к сущности “table” (стол) по средством relationship “makes” (делает). Relationships обозначаются с помощью значка ромб и содержат глагол определяющий это отношение.

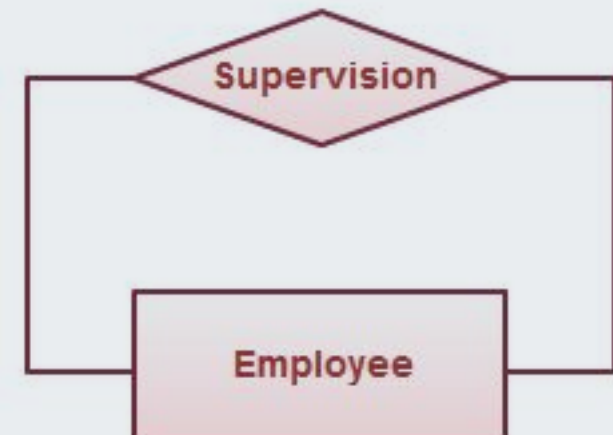


Тип отношения (cardinality):

one2many, one2one, many2many



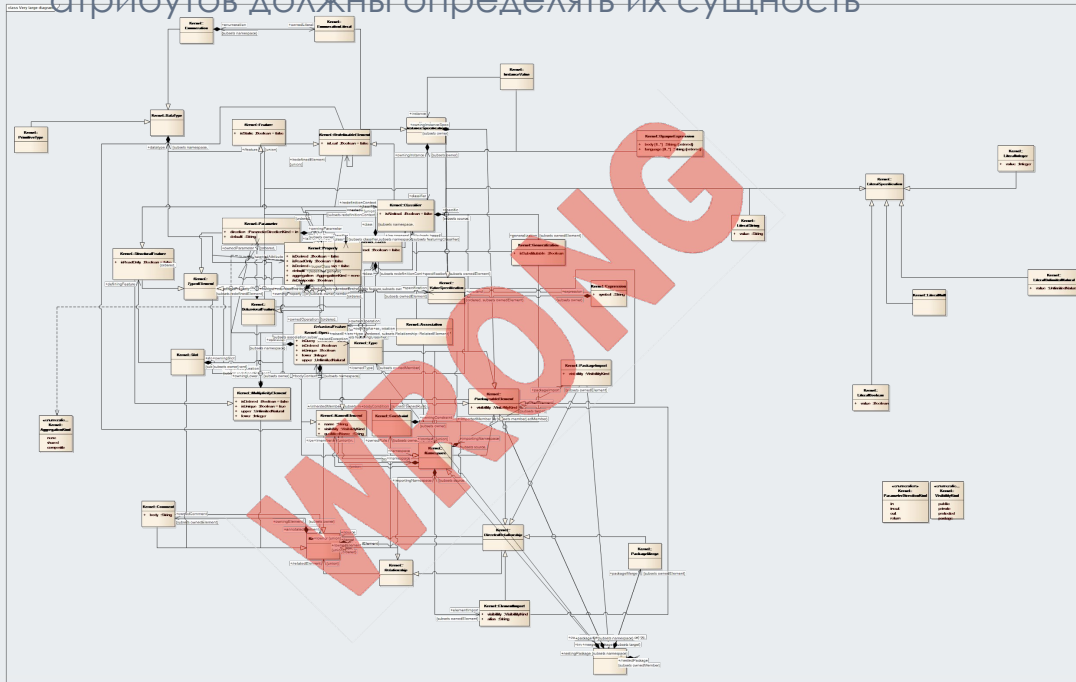
Рекурсивные отношения:



Entity Relation Diagram: Relationship

Пререквизиты для создания диаграммы:

- Определить все entities в информационной системе. Entity должно быть представлено единожды в одной диаграмме (быть уникальным)
- Связать entity друг с другом с помощью relationships .
- Добавить attribute для каждого Entity. Имена атрибутов должны определять их сущность



Советы:

- Указывайте четкие и корректные имена для каждой сущности (простые и понятные всем термины предпочтительней технических терминов).
- Имена должны представлять собой имена существительные.
- Используйте прилагательные для разделения Entity входящих в один класс (part-time employee и full-time employee).
- Имена attributes должны быть понятные, уникальные, не привязаны к конкретной системе.
- Удаляйте все нечеткие, избыточные или ненужные relationship между entities.
- НИКОГДА не соединяйте один relationship с другим напрямую.
- Используйте цвета в диаграмме чтобы объединить entity относящиеся к одному классу или подчеркнуть ключевые области.

3 уровня модели данных

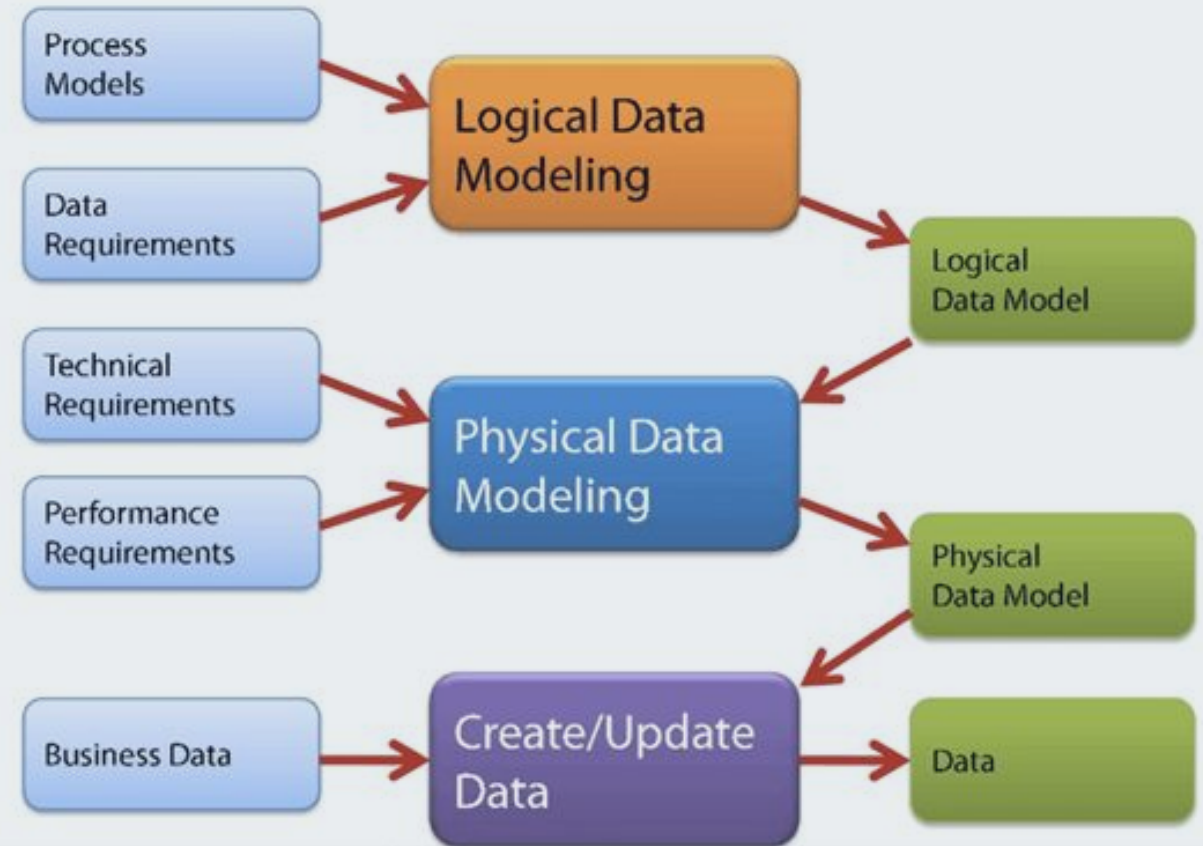
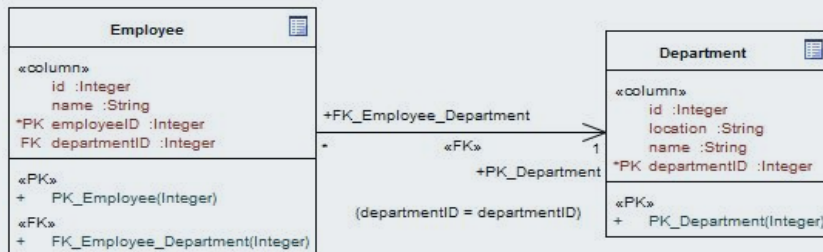


Концептуальный: Этот уровень определяет **ЧТО** будет содержать модель данных.
(Организация определение объема и бизнес концепта/правил для модели данных).

Логический: Определяет **КАК** информационная система будет внедрена без оглядки на конкретные требования баз данных или языка программирования. (основная цель это определить базовые типы данных и связей между ними)

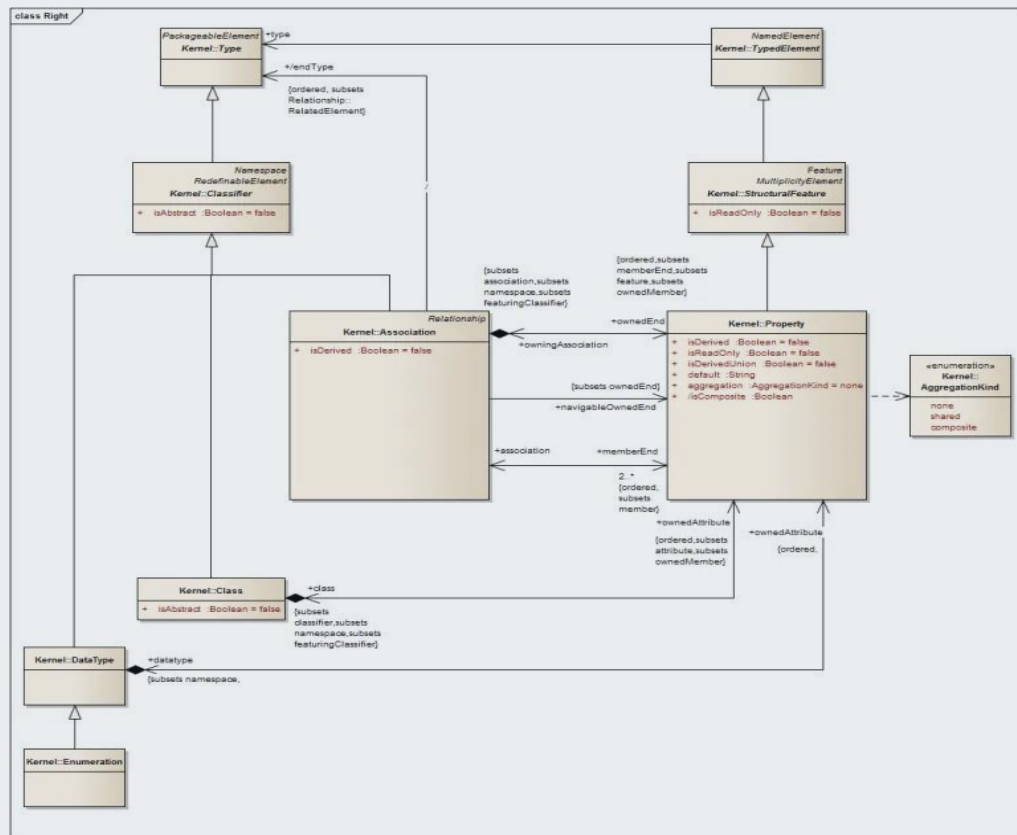


Физический: Определяет **КАК** информационная система будет реализована с использованием конкретной БД или языка программирования.



UML

UML (Unified Modeling Language), специализированный язык моделирования, служащий для визуализации и отражения архитектуры, дизайна и плана внедрения сложных информационных систем, ПО и процессов.



Два основных типа UML диаграмм:

Структурные UML диаграммы:

- Class diagram
- Package diagram
- Object diagram
- Component diagram
- Composite structure diagram
- Deployment diagram

Поведенческие UML диаграммы:

- Activity diagram
- Sequence diagram
- Use case diagram
- State diagram
- Communication diagram
- Interaction overview diagram
- Timing diagram

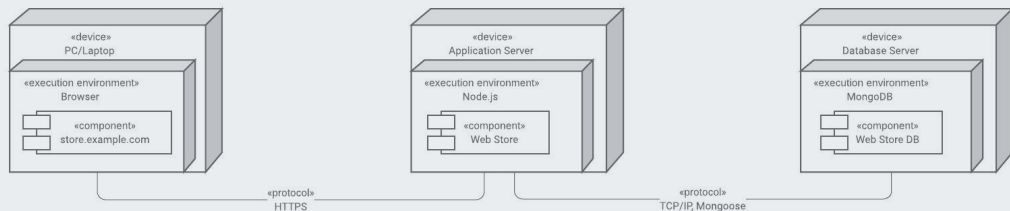
UML: Структурные диаграммы

Структурная UML диаграмма показывает как информационная система или ПО структурирована и имеет Объектно-ориентированную структуру.

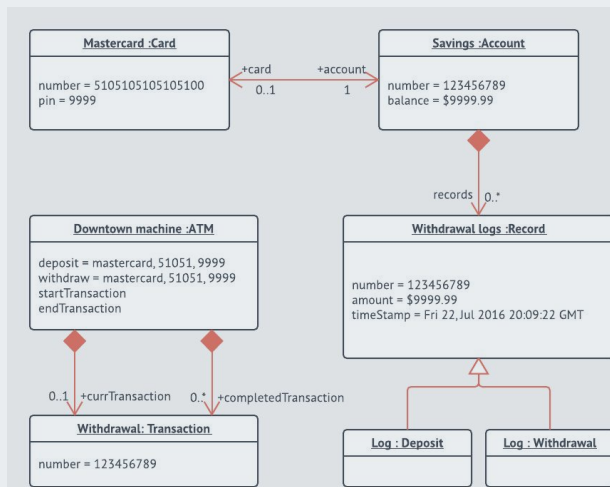
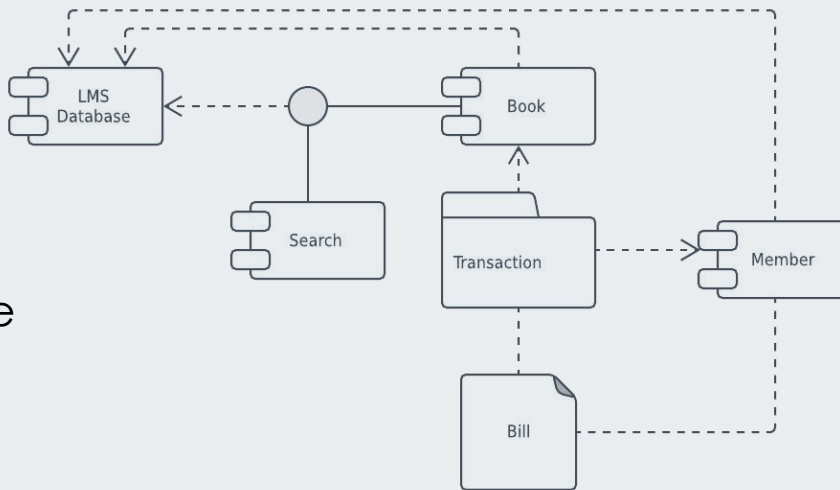
Класс представлен прямоугольником, содержащим 3 компонента следующие в вертикальном порядке

- **Имя класса (обязательно)**
- **Атрибуты класса (опционально)**
- **Операции класса (опционально)**

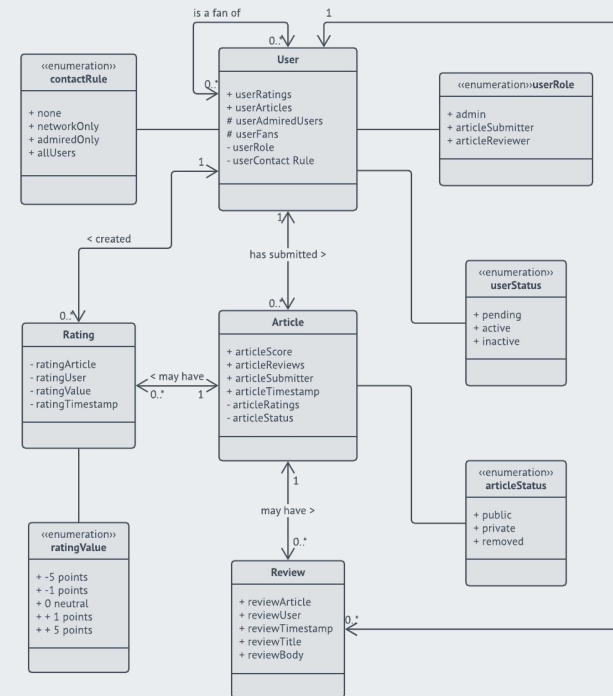
Class UML



Component UML



Class UML

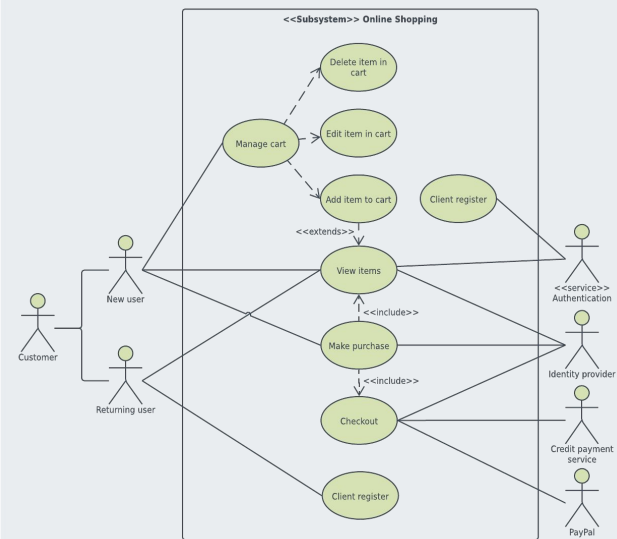


Object UML

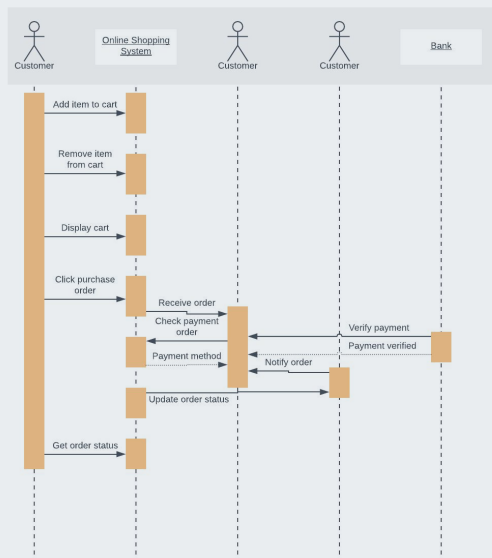
UML: Поведенческие диаграммы

Поведенческая UML диаграмма показывает как система себя ведет и взаимодействует внутри себя, с пользователями и внешними системами.

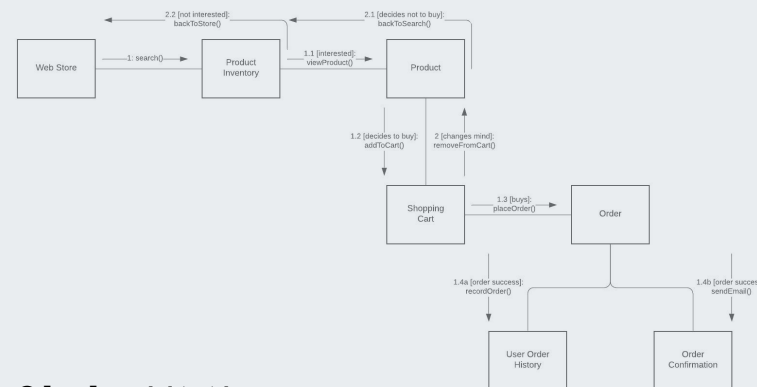
Use Case UML



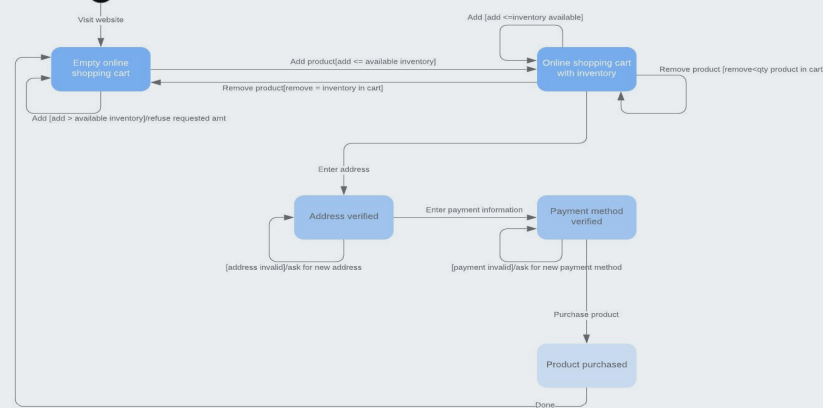
Sequence UML



Communication UML



State UML



Activity UML

