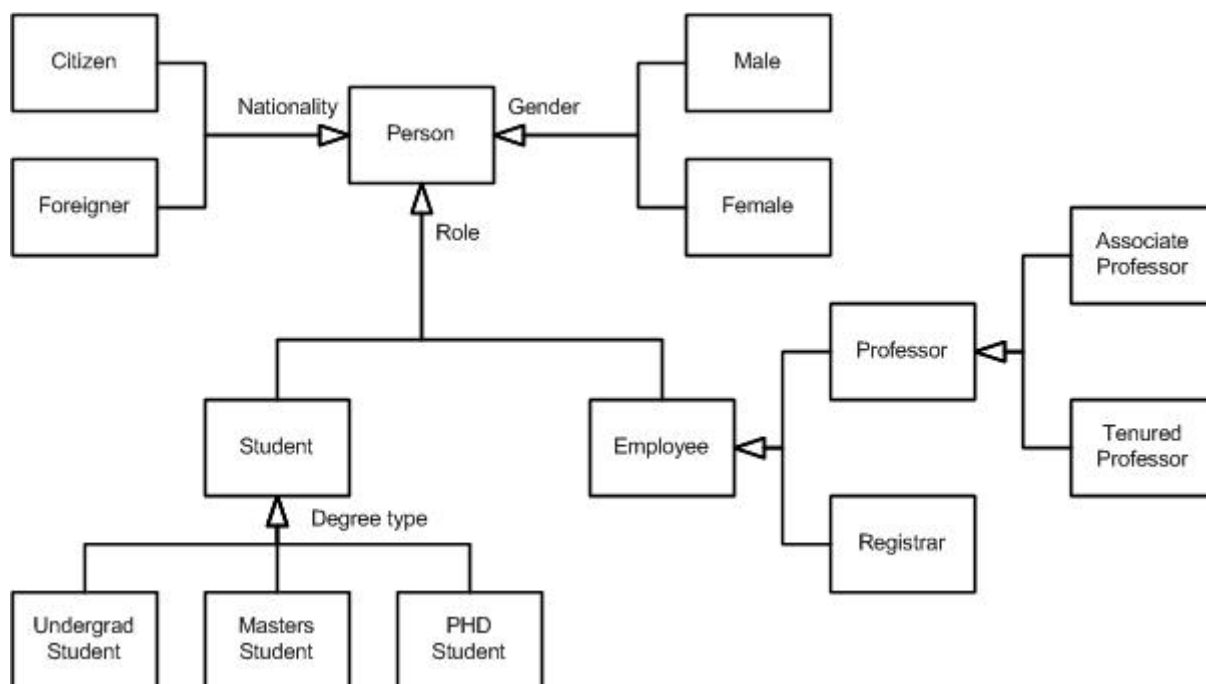


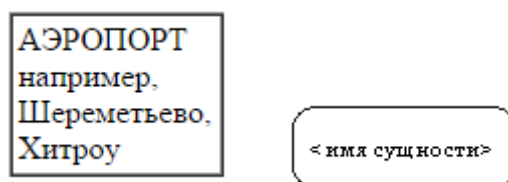
## Концептуальная модель (Conceptual Model, Kontseptuaalmodel)

**Концептуальная модель** – это систематизированное содержательное описание моделируемой системы (или проблемной ситуации). Описание разрабатываемой имитационной модели включает определение основных элементов моделируемой системы, их характеристики и взаимодействие между элементами.



**Сущность (Entity)** - это реальный или представляемый объект, информация о котором должна сохраняться и быть доступна. В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности - это имя типа, а не некоторого конкретного экземпляра этого типа.

Ниже изображена сущность АЭРОПОРТ с примерными объектами Шереметьево и Хитроу:



Графическое изображение сущности

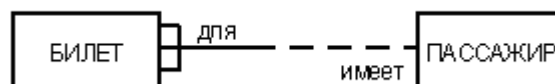


**Связь (Relationship)** - это графически изображаемая ассоциация, устанавливаемая между двумя сущностями. Эта ассоциация всегда является бинарной и может существовать между двумя разными сущностями или между сущностью и ей же самой (рекурсивная связь). В любой связи выделяются два конца (в соответствии с существующей парой связываемых сущностей), на каждом из которых указывается имя конца связи, степень конца связи (сколько экземпляров данной сущности связывается), обязательность связи (т.е. любой ли экземпляр данной сущности должен участвовать в данной связи).

Связь представляется в виде линии, связывающей две сущности или ведущей от сущности к ней же самой. При этом в месте "стыковки" связи с сущностью используются трехточечный вход в прямоугольник сущности, если для этой сущности в связи могут использоваться много (many) экземпляров сущности, и одноточечный вход, если в связи может участвовать только один экземпляр сущности. Обязательный конец связи изображается сплошной линией, а необязательный - прерывистой линией.

Как и сущность, связь - это типовое понятие, все экземпляры обеих пар связываемых сущностей подчиняются правилам связывания.

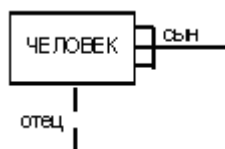
В изображенном ниже примере связь между сущностями БИЛЕТ и ПАССАЖИР связывает билеты и пассажиров. При том конец сущности с именем "для" позволяет связывать с одним пассажиром более одного билета, причем каждый билет должен быть связан с каким-либо пассажиром. Конец сущности с именем "имеет" означает, что каждый билет может принадлежать только одному пассажиру, причем пассажир не обязан иметь хотя бы один билет.



Трактовкой изображенной диаграммы является:

- Каждый БИЛЕТ предназначен для одного и только одного ПАССАЖИРА;
- Каждый ПАССАЖИР может иметь один или более БИЛЕТОВ.

На следующем примере изображена рекурсивная связь, связывающая сущность ЧЕЛОВЕК с ней же самой. Конец связи с именем "сын" определяет тот факт, что у одного отца может быть более чем один сын. Конец связи с именем "отец" означает, что не у каждого человека могут быть сыновья.



Трактовкой изображенной диаграммы является:

- Каждый ЧЕЛОВЕК является сыном одного и только одного ЧЕЛОВЕКА;
- Каждый ЧЕЛОВЕК может являться отцом для одного или более ЛЮДЕЙ ("ЧЕЛОВЕКОВ").

Связи может даваться имя, выражаемое грамматическим оборотом глагола и помещаемое возле линии связи. Имя каждой связи между двумя данными сущностями должно быть уникальным, но имена связей в модели не обязаны быть уникальными. Имя связи всегда формируется с точки зрения родителя, так что предложение может быть образовано соединением имени сущности-родителя, имени связи, выражения степени и имени сущности-потомка.

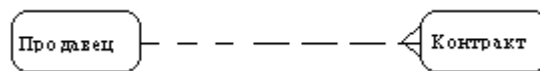
Например, связь продавца с контрактом может быть выражена следующим образом:

- продавец может получить вознаграждение за 1 или более контрактов;
- контракт должен быть инициирован ровно одним продавцом.

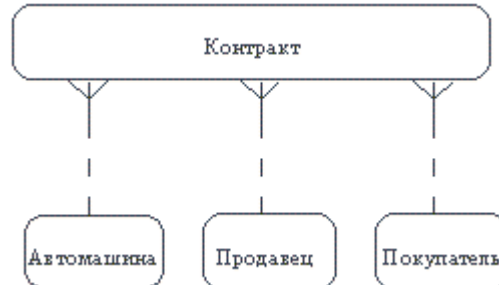
Степень связи и обязательность графически изображаются следующим образом:



Таким образом, 2 предложения, описывающие связь продавца с контрактом, графически будут выражены следующим образом:



Описав также связи остальных сущностей, получим следующую схему:



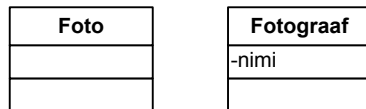
Связь может дополнительно определяться с помощью указания степени или мощности (количества экземпляров сущности-потомка, которое может существовать для каждого экземпляра сущности-родителя):

- каждый экземпляр сущности-родителя может иметь ноль, один или более связанных с ним экземпляров сущности-потомка;
- каждый экземпляр сущности-родителя должен иметь не менее одного связанного с ним экземпляра сущности-потомка;
- каждый экземпляр сущности-родителя должен иметь не более одного связанного с ним экземпляра сущности-потомка;
- каждый экземпляр сущности-родителя связан с некоторым фиксированным числом экземпляров сущности-потомка.

**Kontsept** kuulub reaalsesse maailma, on osa sellest (näiteks: õpilane, õpetaja, tudeng, klient, töötaja, dokument, leping jne) ja seda on vaja süsteemis meeles pidada.

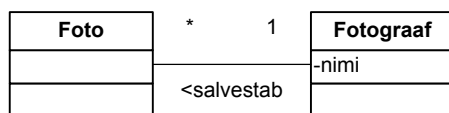
- igal kontseptil on 1 või mitu omadust (atribuuti), mis teda kirjeldab, iseloomustab
- iga kontsept peab olema eristatav teistest kontseptidest

Kontsepte kujutatakse diagrammil kastina, milles on kontsepti nimi ja vahel ka mõned olulised omadused.



**Kontseptid on omavahel seotud ehk suhtes.** Suhe esitab seost kontseptide vahel – mitu ühe kontsepti eksemplari on seotud teise kontseptiga. Kontseptide vahel võib seoseid olla ka mitu.

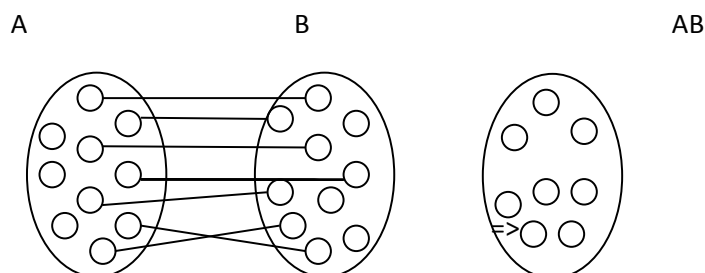
**Seosed on suunamata jooned**, millele märgitakse, mida seos väljendab koos teksti suunaga, et oleks arusaadav, mida üks kontsept teise suhtes „teeb“.



Seoses olevate kontseptidega võib olla seotud ka mahuline näitaja, mis näitab, mitu kontsepti antud seosega seotud on. Tähistus võib olla konkreetne suurus (0, 4, 1) või vahemik (1-5, \*-arv ei ole teada, 6-\*).

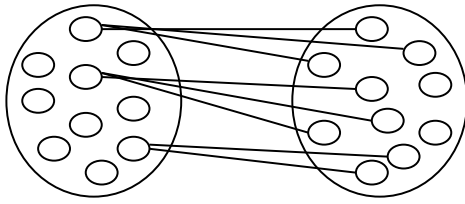
## SUHTED

### 1:1 (1:0?)

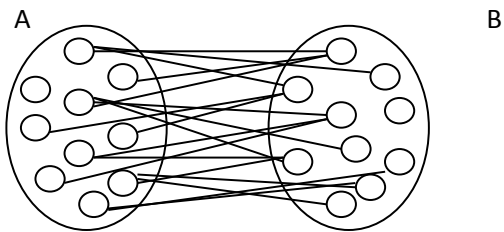


### 1:0..M





**0:M..0:N**

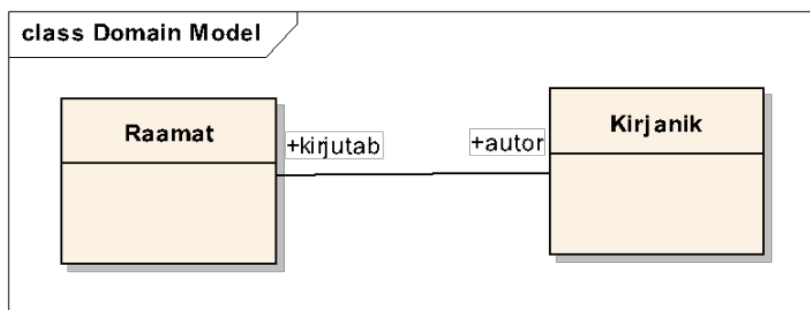


### Suhted kontseptuaalmudelis

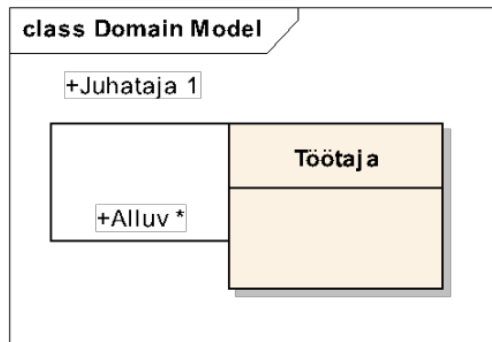
Üldjuhul piisab kontseptuaalmudelis suunamata seosest, millele on lisatud seost iseloomustav nimi ja seose mahud.

Kui soovitakse anda edasi detailsemat infot, võib kasutada erinevaid klassidiagrammi seoseid.

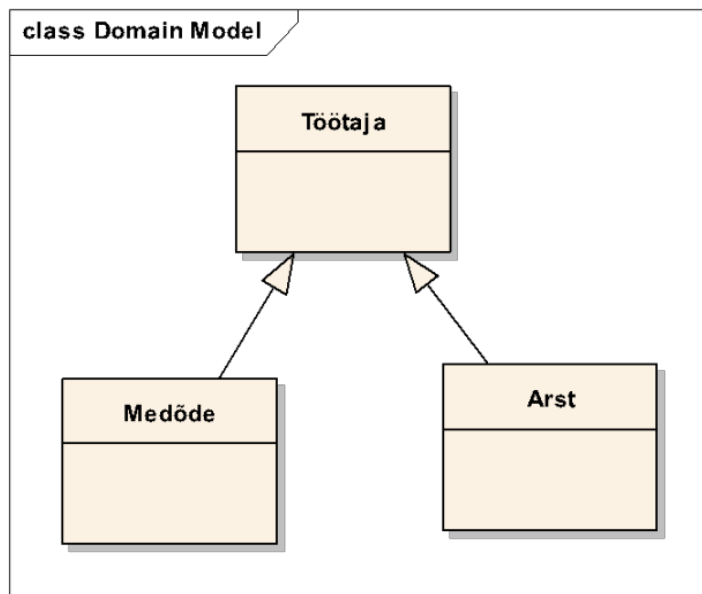
- Sageli kirjutatakse seose nimetuse asemel seose otstes seotud elementide rollid



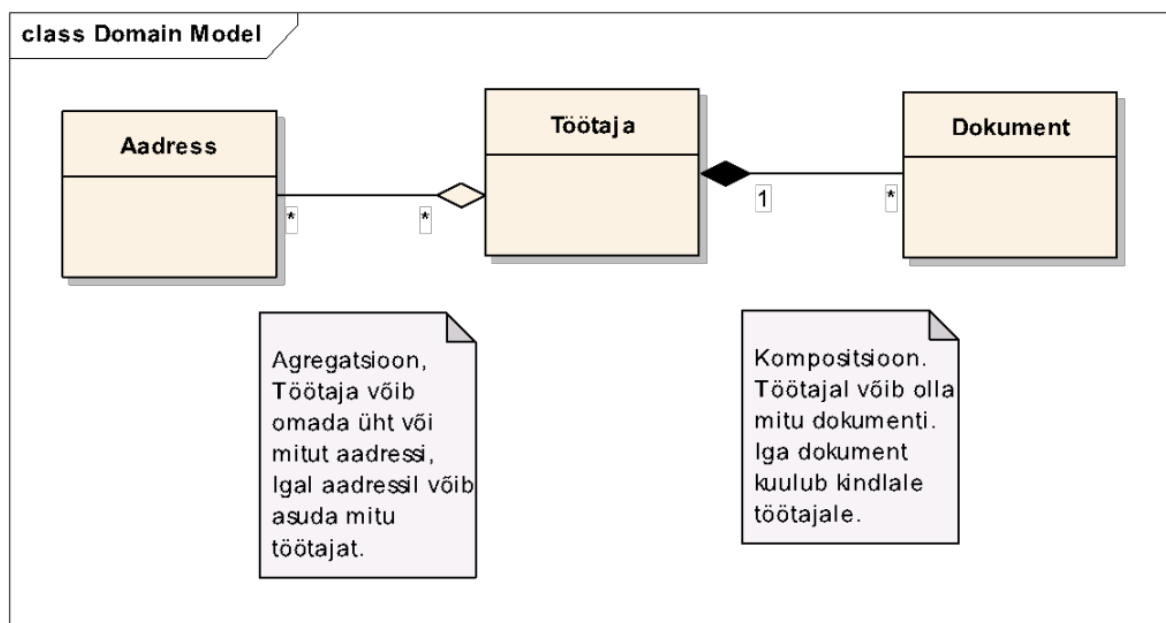
- Kontsept võib omada suhet ka iseendaga:

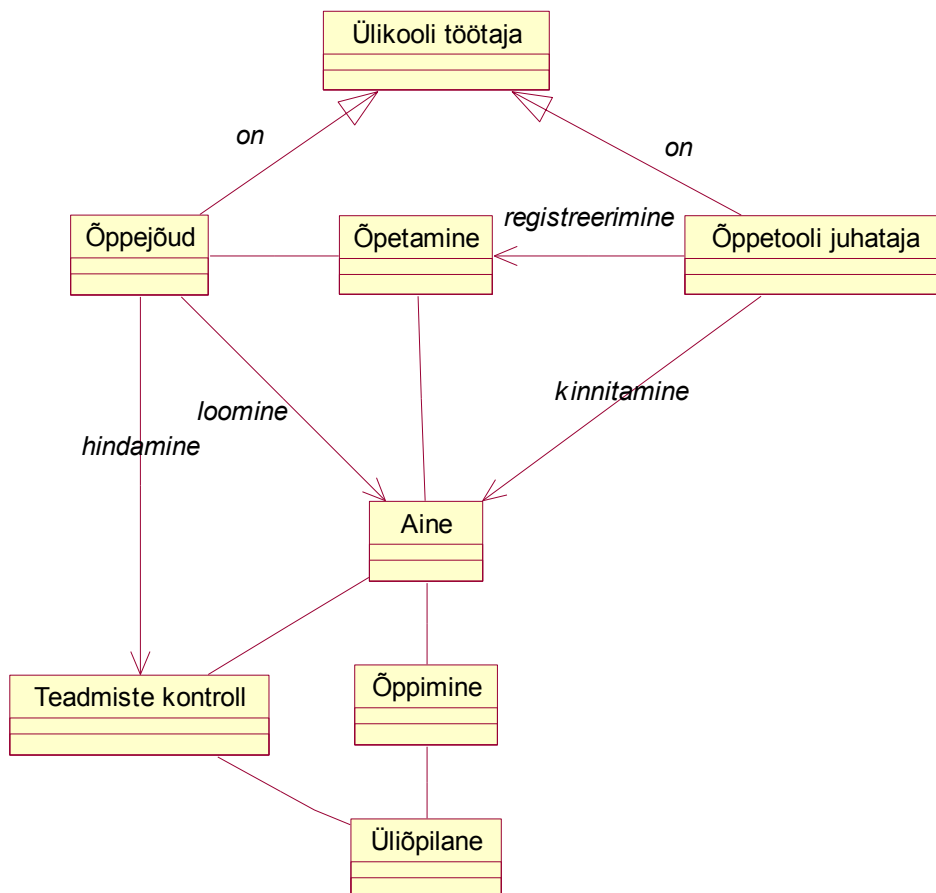


### Обобщение (Generalization, Üldistus):



- Liitseosed – agregatsioon (Aggregation) ja kompositsioon (Composition)





Enterprise Architect: New View -> Class View -> New Diagram ->UML Structural -> Class.

ArgoUML-s: Create New Class Diagram

**Задание: Концептуальная модель (Conceptual Model)**

Построить концептуальную модель инфосистемы.