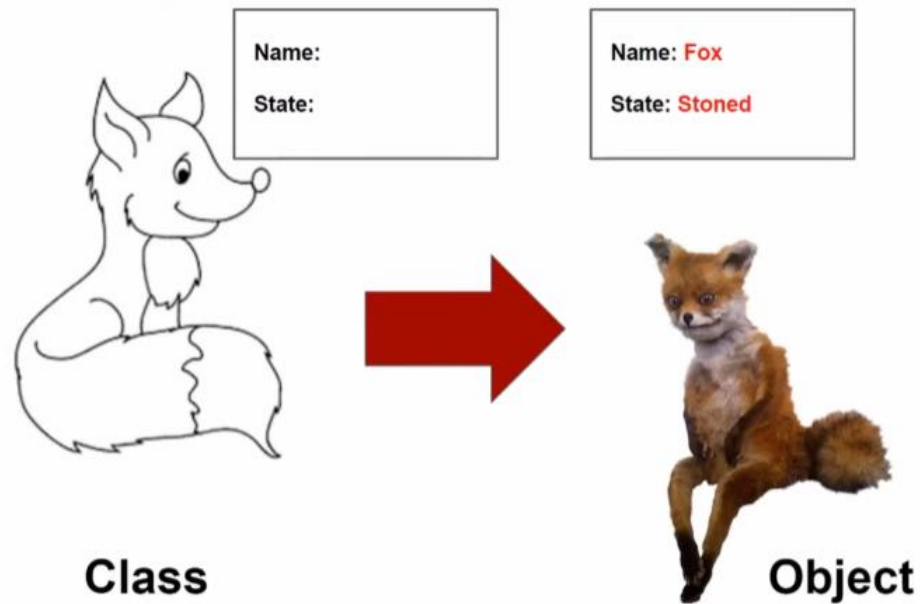


Class Diagram

OOD

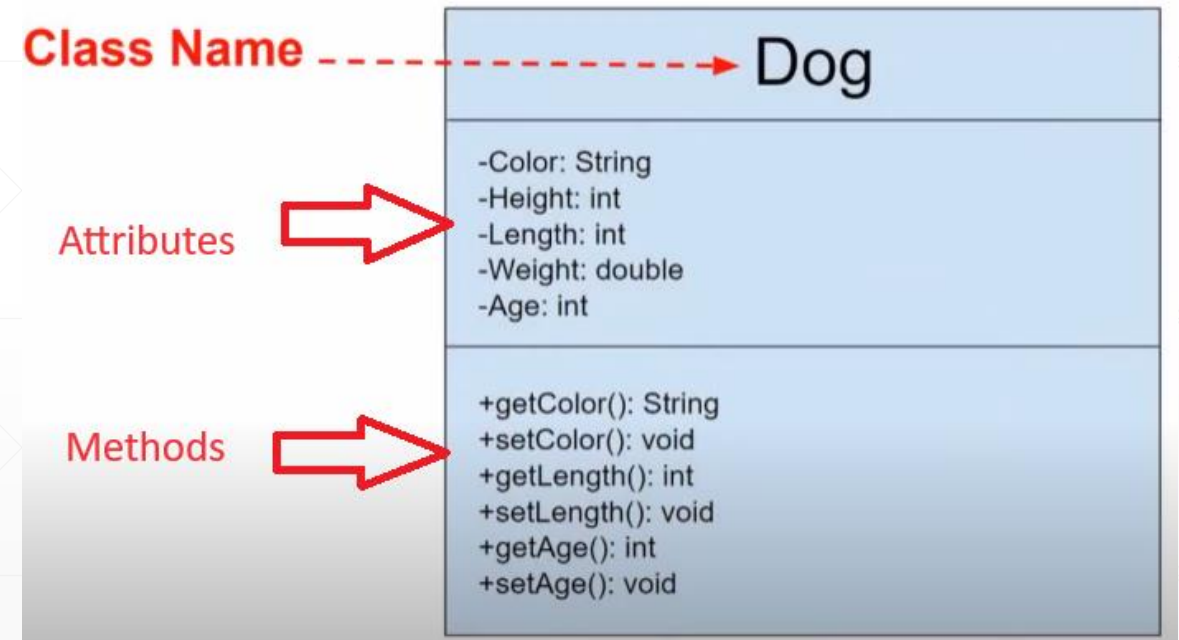
Třída/Class

- Šablona pro vytváření objektů
- Deklaruje počáteční stav, vlastnosti a chování (metody)



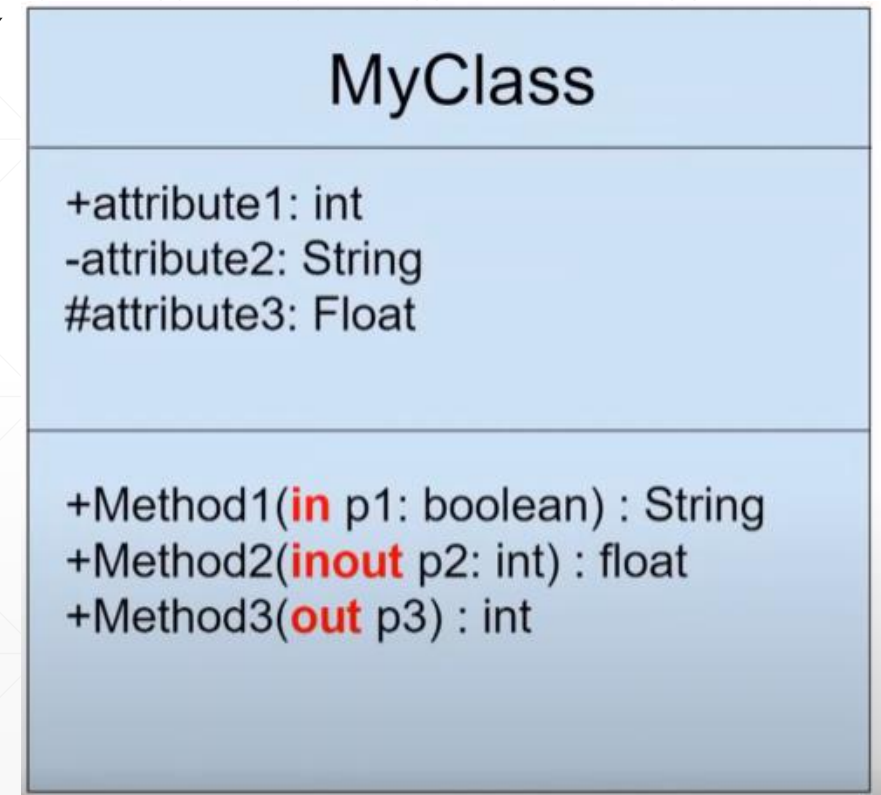
Definování třídy/Class

- V horní části vypisujeme název třídy
- Následně zapisujeme vlastnosti
- Jako poslední zapisujeme metody
- Znaky u vlastností a method:
 - + Public
 - - Private
 - # Protected
 - Package Local



In VS Inout VS Out

- In = Parametr vstupuje do metody, ale nijak se nemění
- Inout = Parametr vstupuje do metody a může se měnit
- Out = Parametr vystupuje jako návratová hodnota



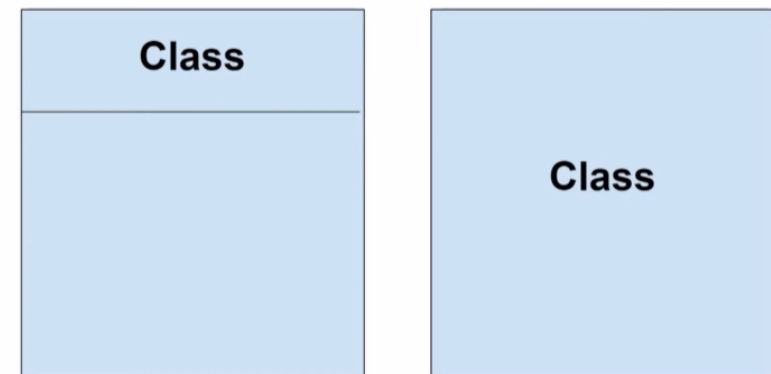
Perspektivity

- Koncepční/Concept
 - Specifikační/Specification
 - Implementační/Implementation
-

Koncepční/Concept

- Diagramy jsou popisovány jako objekty reálného světa
- Diagram reprezentuje koncepty v doméně
- Tyto koncepty jsou vázané na třídy, které je implementují
- Koncepční perspektiva je považována jako jazykově nezávislá

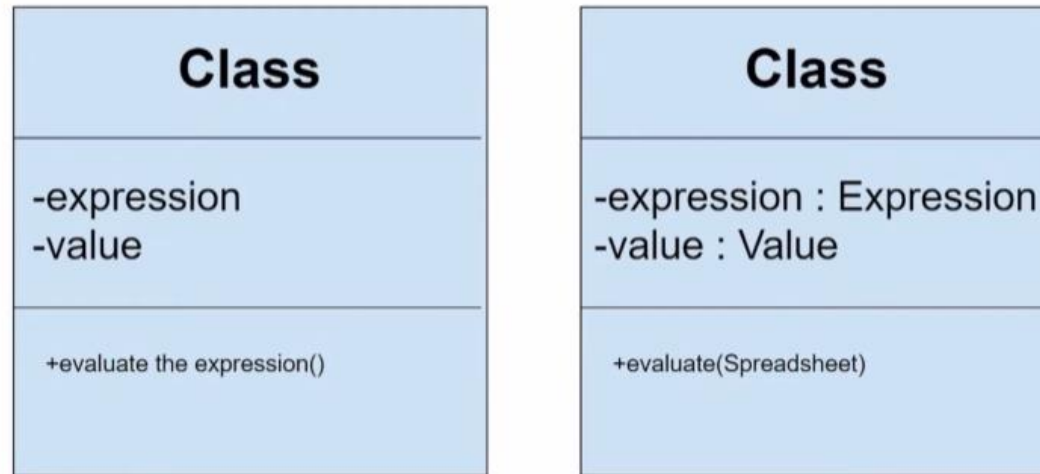
Conceptual Perspective



Specifická/Specification

- Diagramy popisují abstrakci softwaru nebo komponent se specifikací a interface, ale bez reference na implementaci

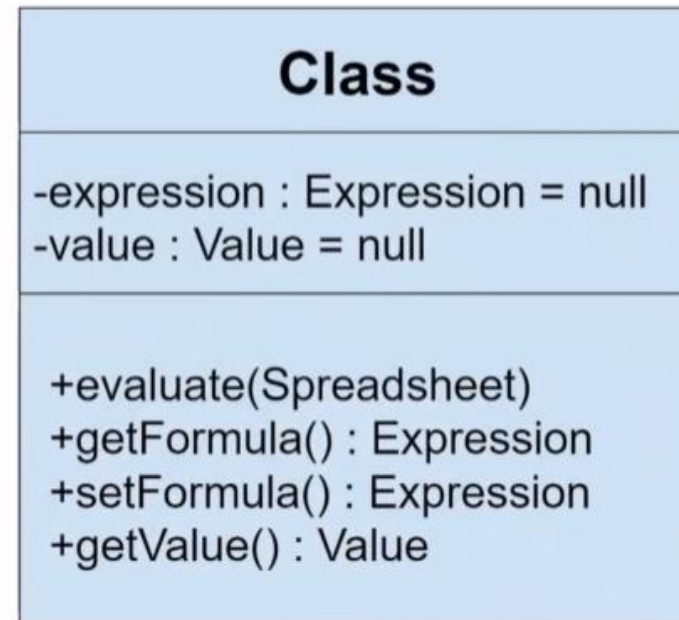
Specification Perspective



Implementační/Implementation

- Diagramy popisují softwarovou implementaci dle nějaké technologie nebo programovacího jazyku

Implementation Perspective



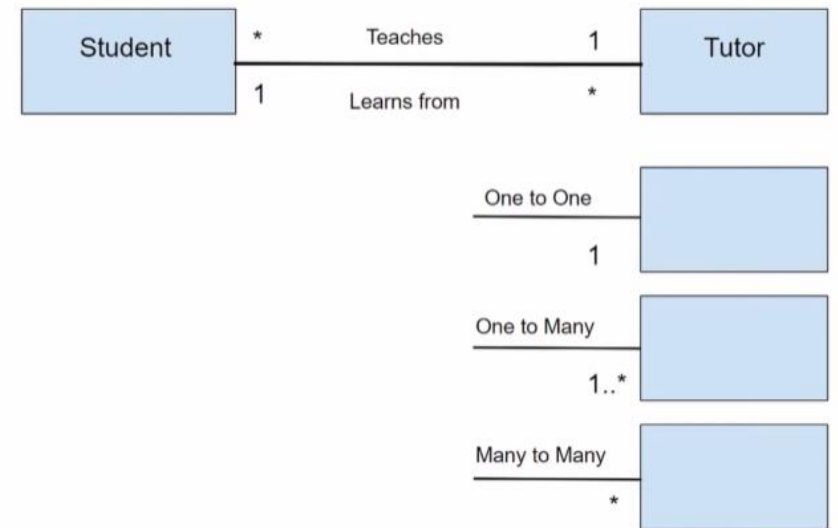
Relace mezi třídami v UML

- Asociace/Association
 - Dědičnost/Inheritance
 - Realizace/Realisation
 - Závislost/Dependency
 - Agregace/Aggregation
 - Kompozice/Composition
-

Asociace/Association

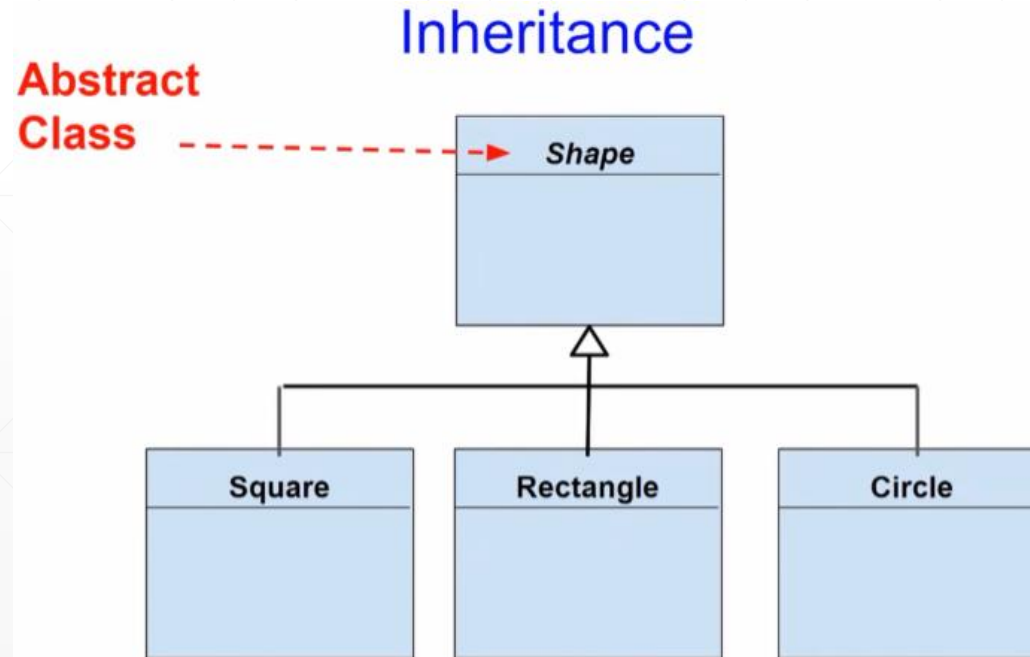
- Dvě třídy spolu nějak komunikují
- Můžeme specifikovat mnohočetnost
- 1:1= Jedna položka odpovídá jedné položce
- 1:N= Jedna položka odpovídá několika položkám
- N:N= Několik položek odpovídá několika položkám

Association



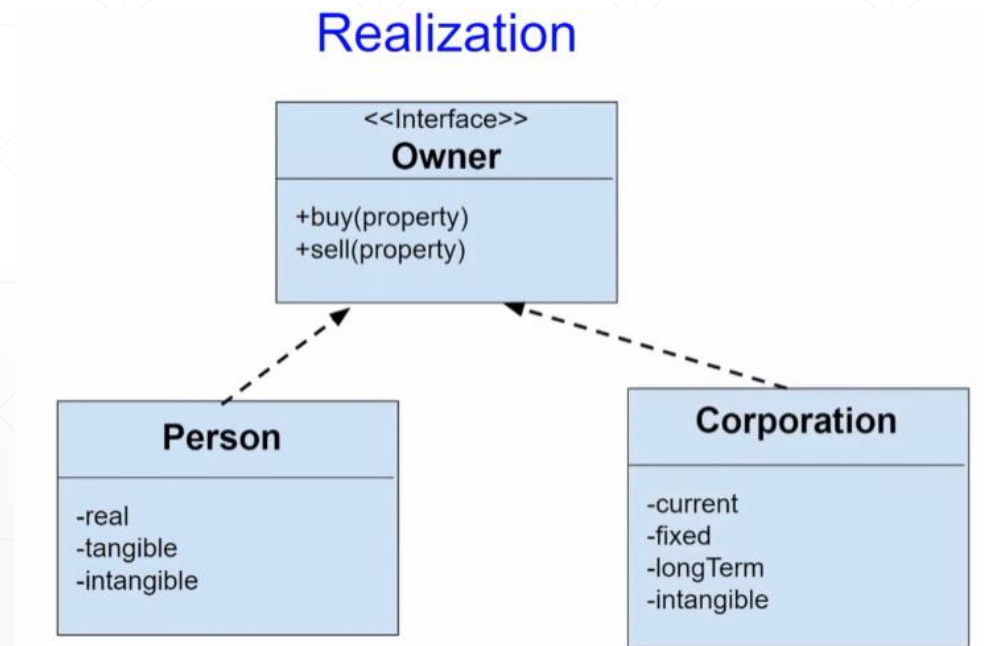
Dědičnost/Inheritance

- Schématické zobrazení relací mezi rodičovskou třídou a potomkem
- Pokud je název třídy napsán kurzivou, je třída abstraktní



Realizace/Realization

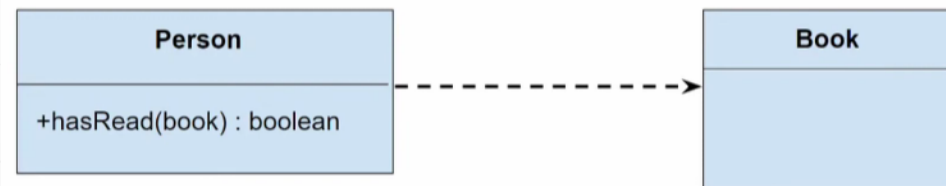
- Běžně referuje relace mezi interface a objekty, které tento interface implementují



Závislost/Dependency

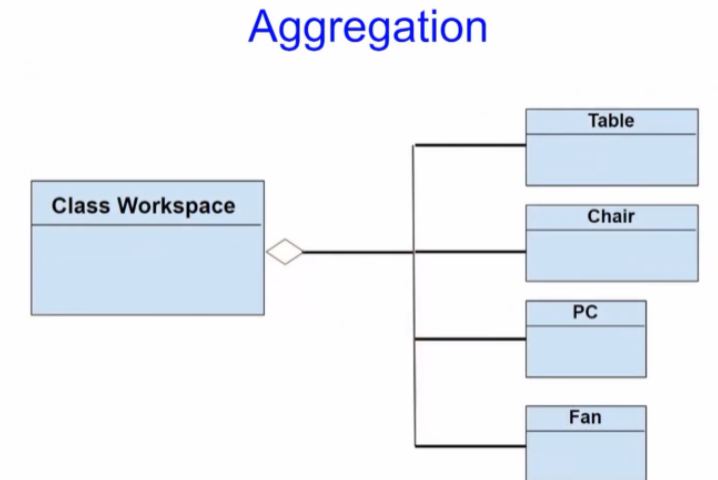
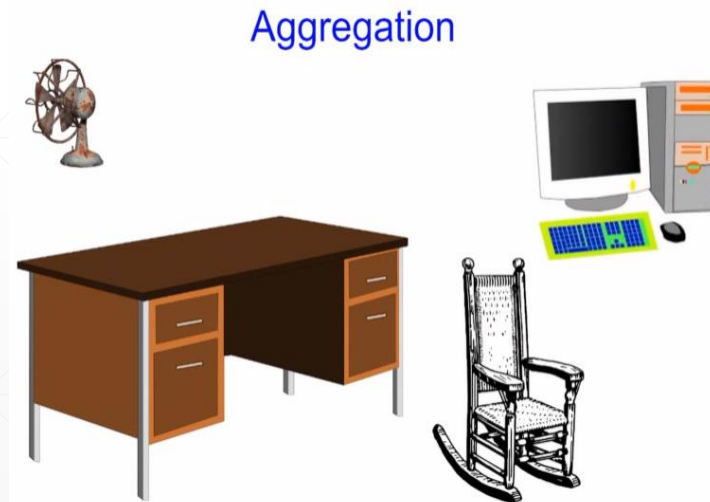
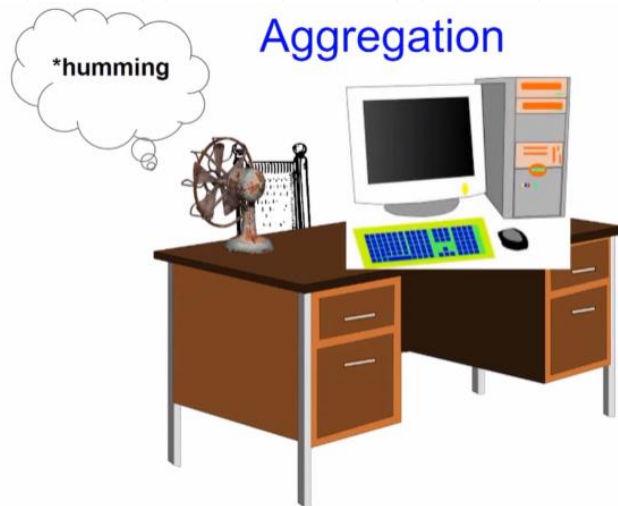
- Objekt jedné třídy používá objekt jiné třídy ve svých metodách
- Speciální případ asociace dvou tříd
- Změna v jedné třídě ovlivní druhou třídu

Dependency



Agregace/Aggregation

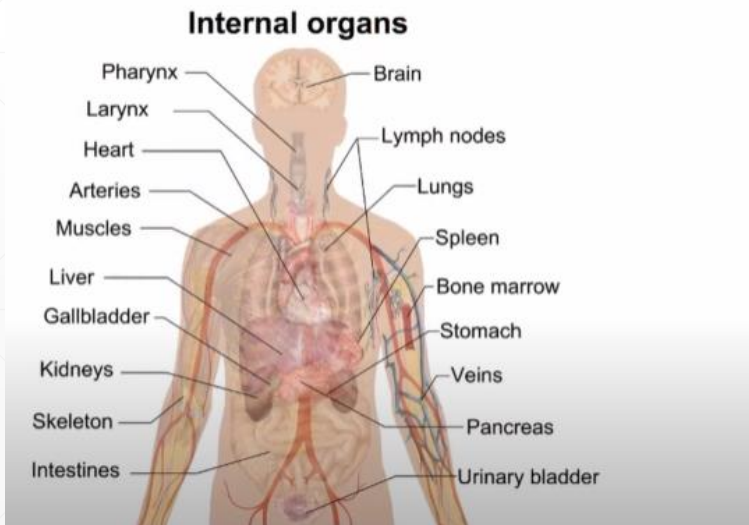
- Speciální typ relace mezi třídami, kdy jedna třída je kompoziční část jiné třídy
- Objekty tvoří nějaký agregovaný objekt, ale mohou existovat i samostatně



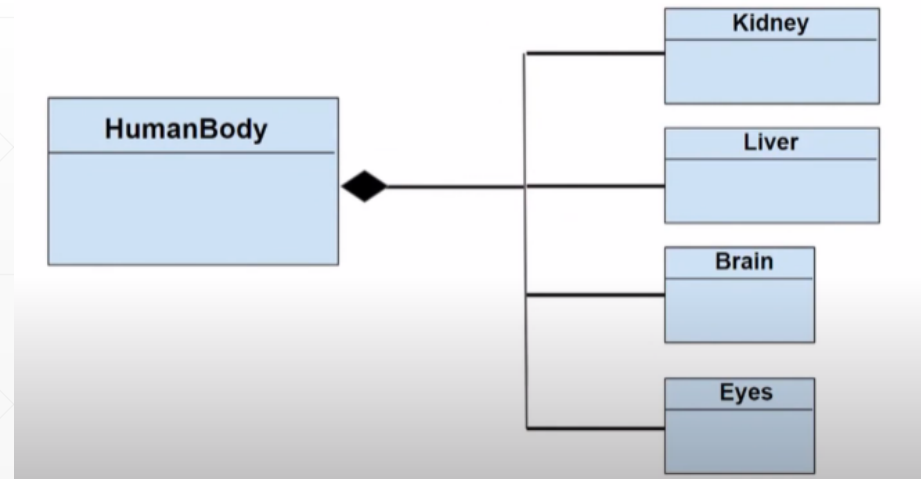
Kompozice/Composition

- Typ agregace, ve kterém všechny objekty zanikají, když zanikne agregovaná třída
- Lidské tělo je tvořeno orgány, ale nemohou fungovat samostatně

Composition



Composition



Zdroje

- <https://www.youtube.com/watch?v=ocgD68K-gbo&t=13s>

