Unified Modeling Language (UML)

Programmering-B

Thomas Birk Abildgaard

October 1, 2025

HTX Frederikshavn

UML - Introduktion

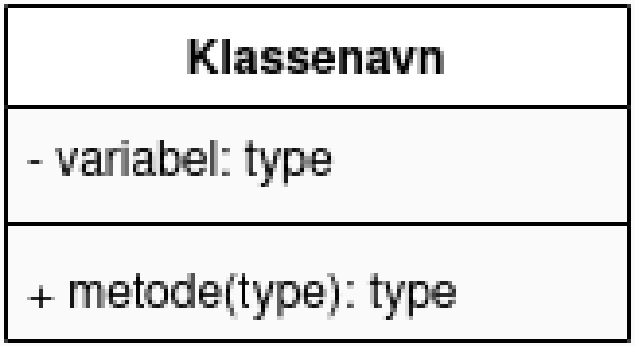
Hvad er UML?

- Standard sprog til at visualisere, specificere, konstruere artifakter i et software system.
- Standard siden 1997 (Object Management Group) UML v1.0
- UML er ikke et programmeringssprog.
- (Der eksisterer dog software der kan generere kode bga. en UML spec.)
- Mål : Et simpelt og general-purpose modeleringssprog
- OO Analyse og Design & UML er "gode venner".
- UML modelering af: Klasser, objekter, abstraktioner, nedarvning, komposition og polymorfi m.m.

Klassediagrammet

Klassen

- Navn
- Attributter (variabler)
- Metod navn
- Argumenter & Retur type
- Modifiers
  - + (public)
  - - (private)
  - # (protected)

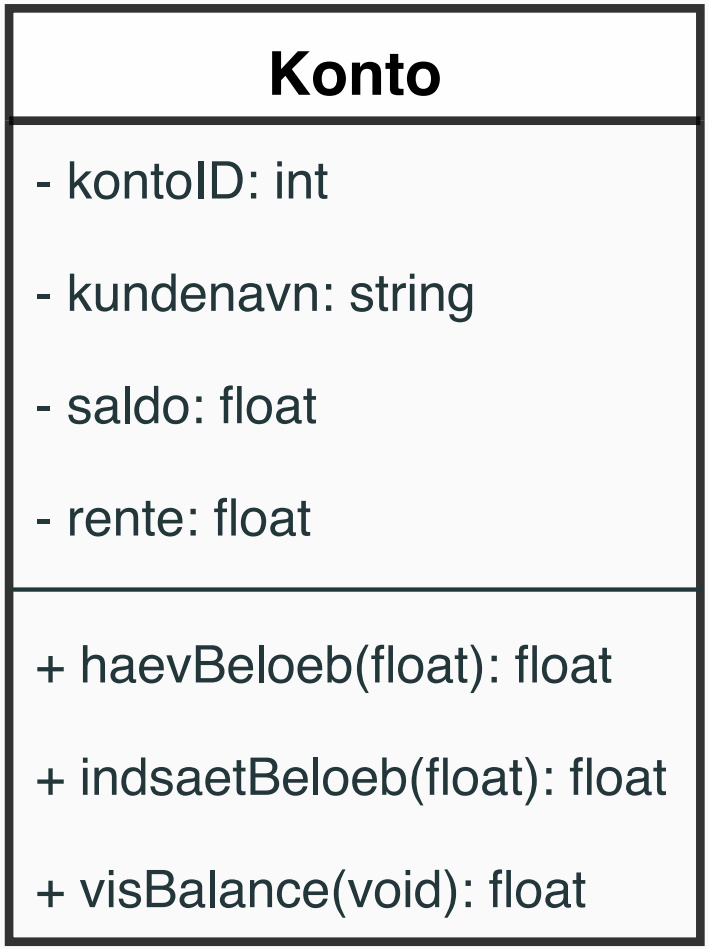


Default

Attributter $\rightarrow$ private

Metoder $\rightarrow$ public

Eksempel : Klasse



Relationer

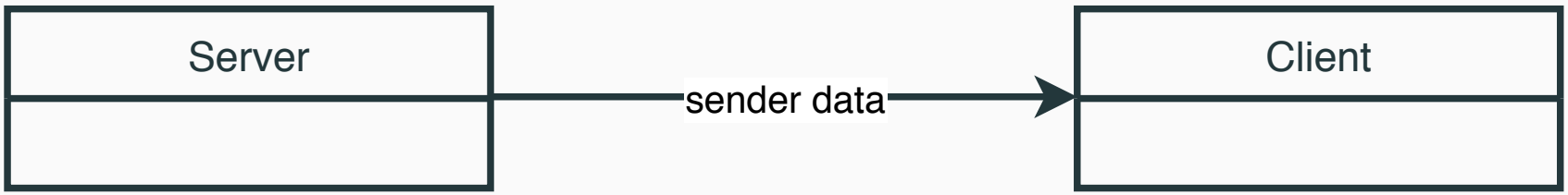
Klasser & Relationer

- Association
- Aggregering
- Composition
- Nedarvning (Generalisering & Specificering)

Association

- Angiver to klassers kommunikation
- Pil angiver kommunikations retning
- Label angiver "forhold"
- Løs binding

Association eks.1

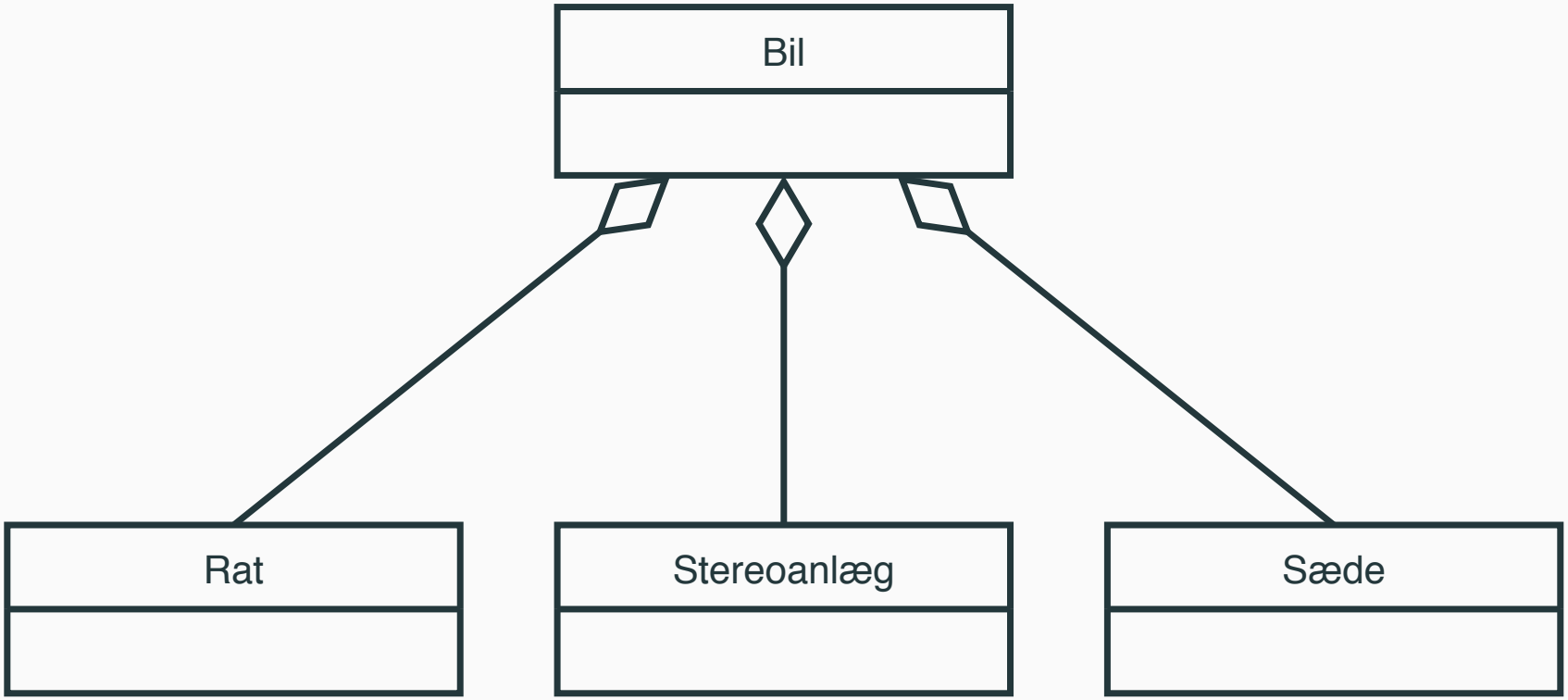


Aggregering & Composition

Aggregering

- Aggregering viser et forhold hvor "barn" kan eksistere uden "forældre"
- Svag forening
- Forhold : "Har en"
- Slettelse af samling påvirker ikke dens dele.

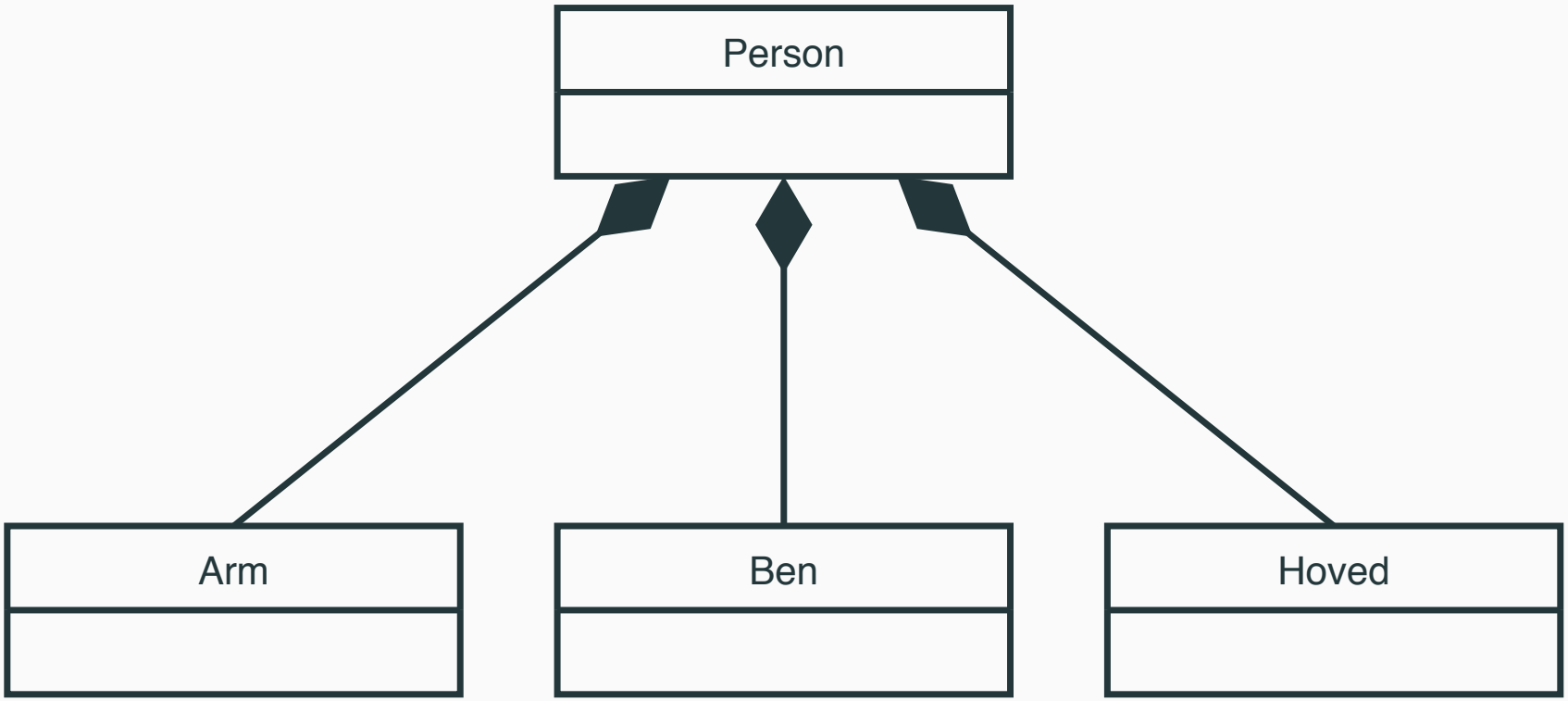
Aggregering



Composition

- I sammensætning kan den ikke eksistere uafhængigt af forældrene.
- Stærk tilknytning
- Forhold : "Er en del af"
- Hvis ejer objekt slettes, slettes indeholdende klasse objekt.

Composition eks.1



Multiplicitet

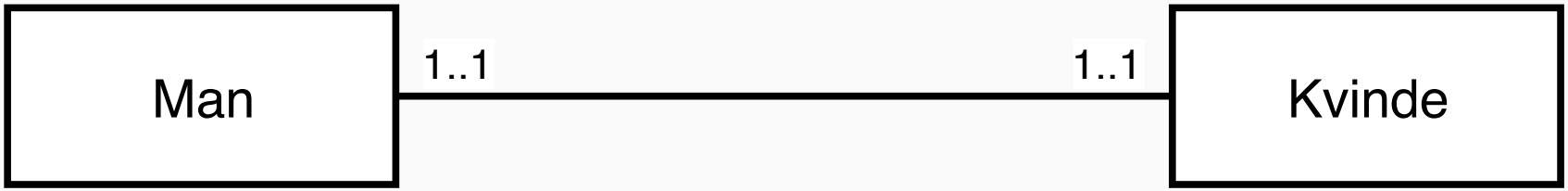
Multiplicitet

Nedre og øvre grænse for en relation. (ell størrelse på collection)
[nedre..øvre]

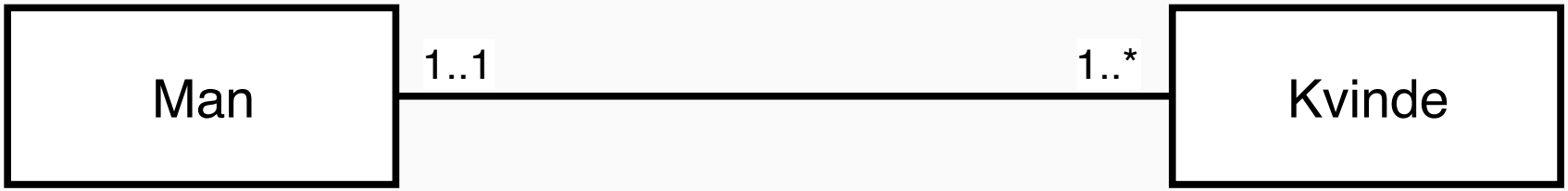
Multiplicity	Option	Cardinality
0..0	0	Collection must be empty
0..1		No instances or one instance
1..1	1	Exactly one instance
0..*	*	Zero or more instances
1..*		At least one instance
5..5	5	Exactly 5 instances
m..n		At least m but no more than n instances

kan bruges på klasse attributter(collection) og klasse relationer.

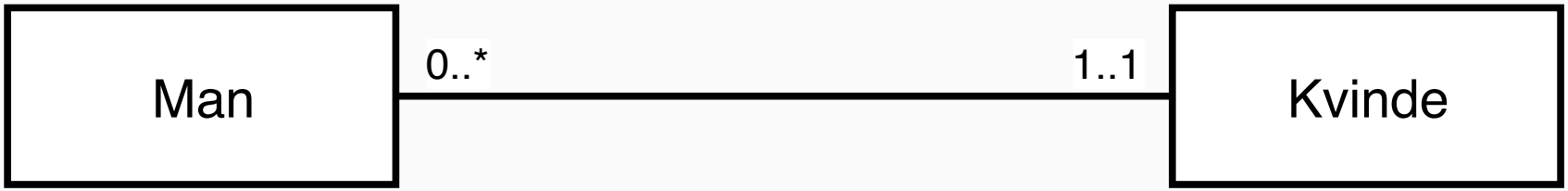
Multiplicitet eks.1



Multiplicitet eks.2



Multiplicitet eks.3



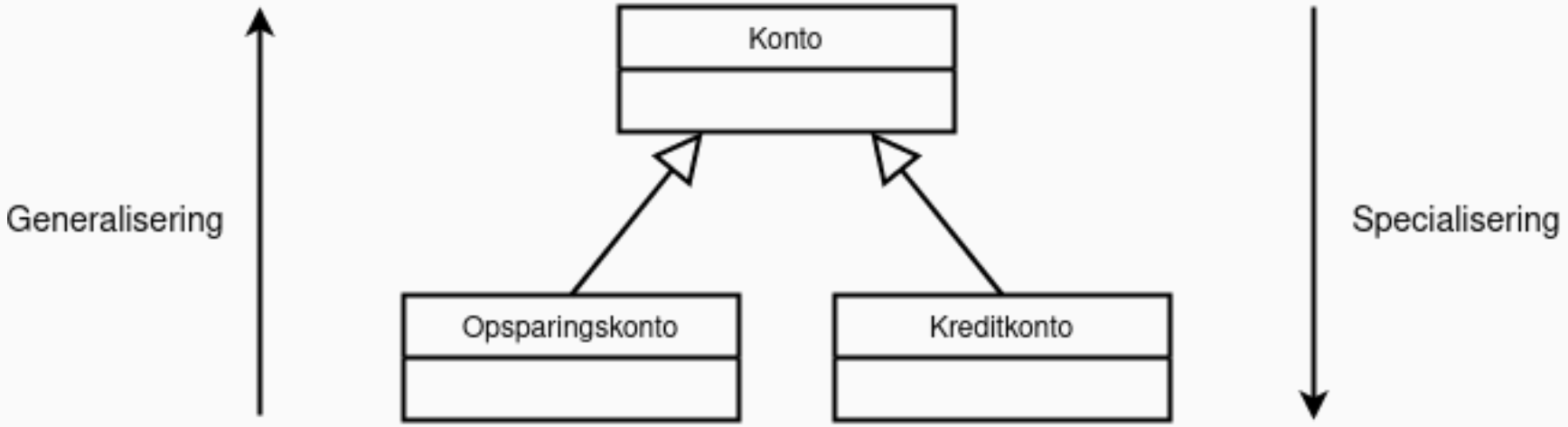
Multiplicitet eks.4

En HTX ansat må maksimalt have én laptop

Hvad skriver angives dette?

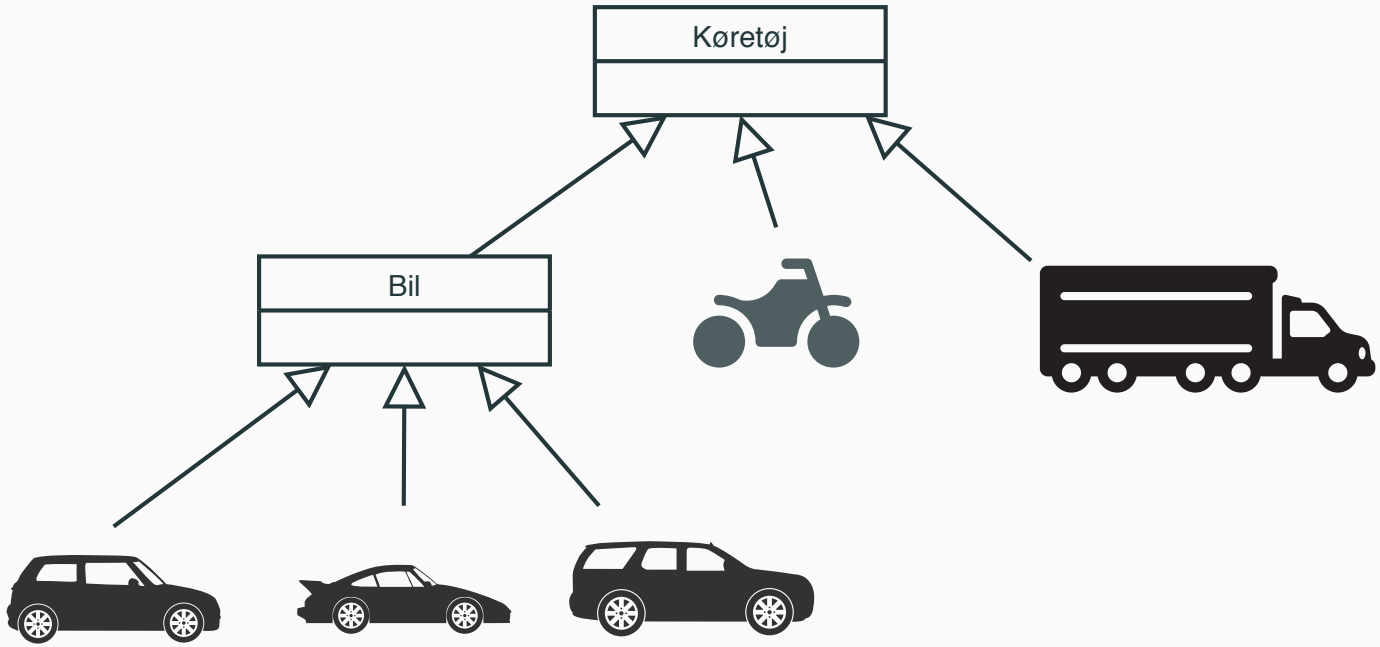
Generalisering & Specialisering

Generalisering & Specialisering (Nedarvning)

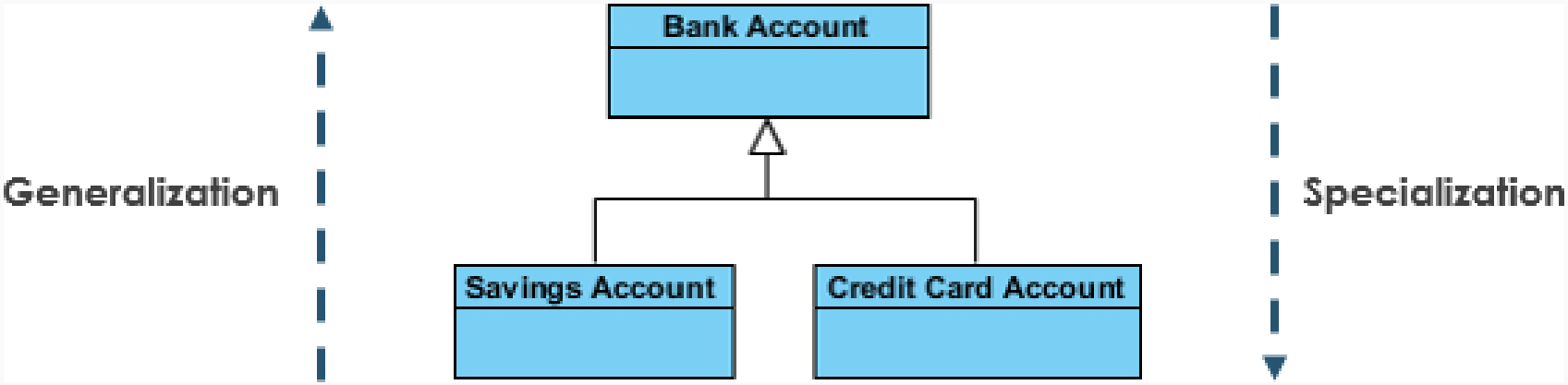


Super- vs Subtype

Nedarvning



Nedarvning eks.2



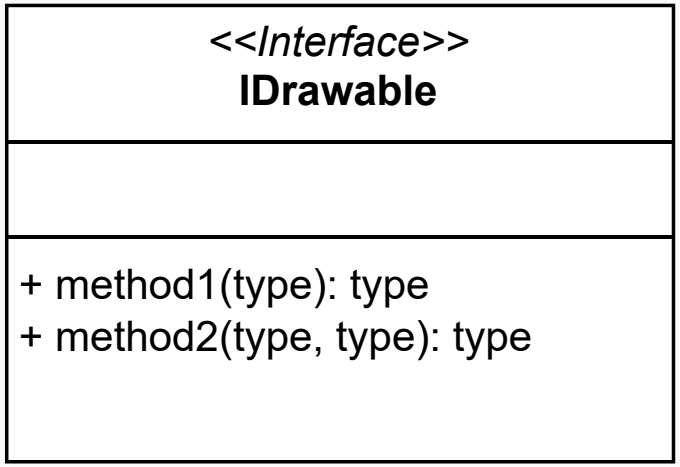
Opgave

Lav et klassediagram over (en del af) SuperAdventureRPG:
Inddrag nedenstående klasser:

- Monster
- Location
- LivingCreature

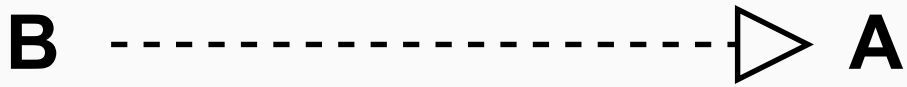
Interfaces

Interface



- Markeres altid med «Interface»
- Interface navn prefixes med bogstavet 'I'. (C# specifikt)
- Ingen attributter eller felter, **kun operationer**.(klassisk UML)
  - *obs: C# .NET8 eller nyere tillader **properties** i interfaces. (En property i virkeligheden en kontrakt for getters/setters, dvs. kontrolleret adgang til et bagvedliggende felt.*
- Bruges typisk til at modellere **roller**, **kontrakter** og **API'er** i

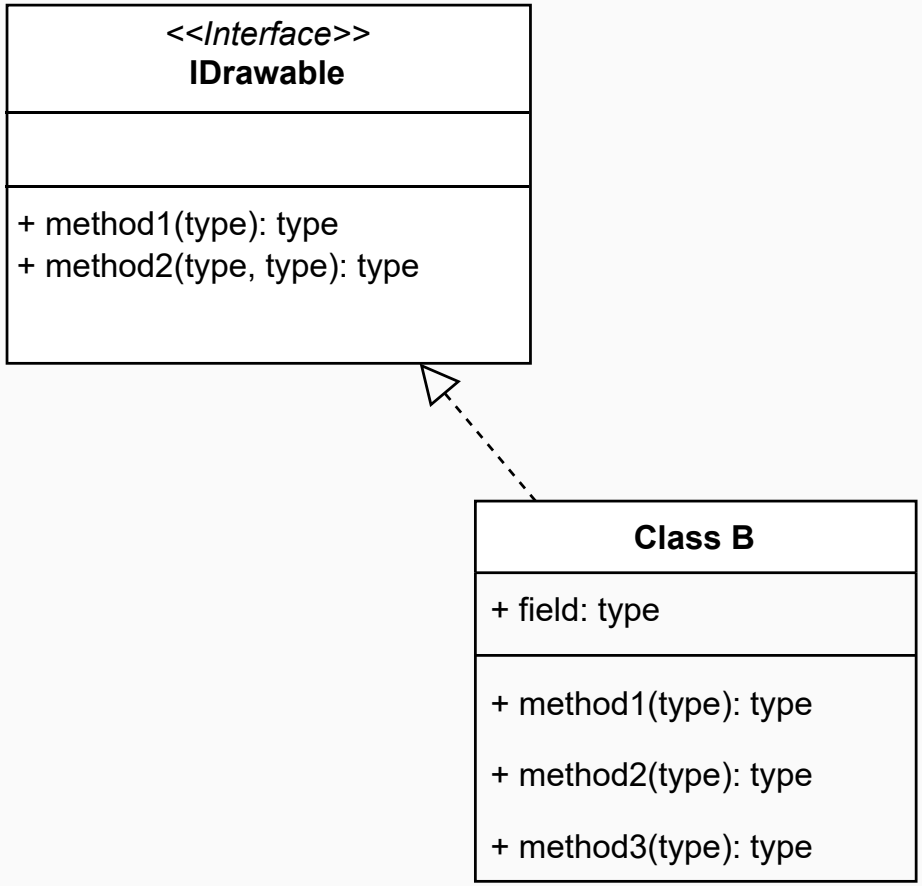
Interface - implements relationen



Læses: B implements A

B implementerer interface A, dvs **B** skal opfylde *kontrakten* og implementere **alle** metoder i interface **A**.

Interface - implements relationen



Klasse B, implementerer interface *IDrawable*