

# Metamodelis. MOF. UML klašu diagrammas elementu metamodelis.

3. Lekcija

Ē. Asņina, 2011. m.g.

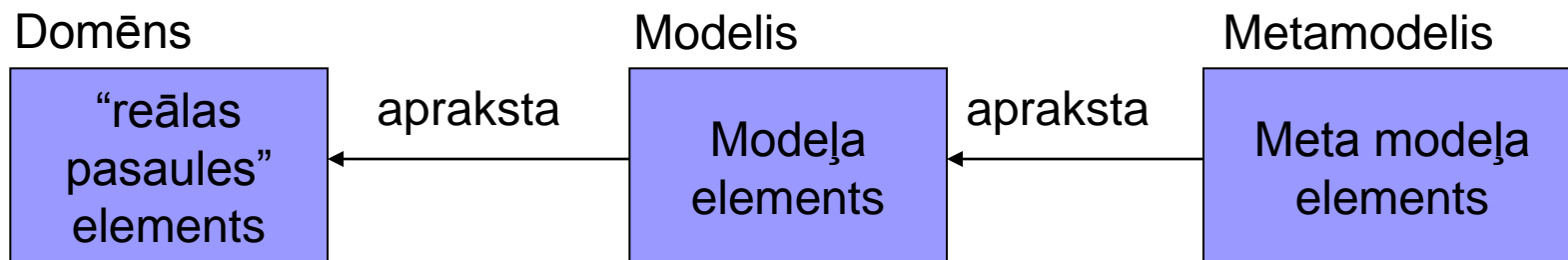


# Kas ir metamodelēšana

- Metamodelēšanas zināšanas ir nepieciešamas:
  - Domēnam specifiskās modelēšanas valodas (DSL) konstruēšanai
  - Modeļa validācijai
  - No modeļa uz modeli transformācijām
  - Koda ģenerācijai
  - Rīku integrācijai

# Kas ir metamodelēšana

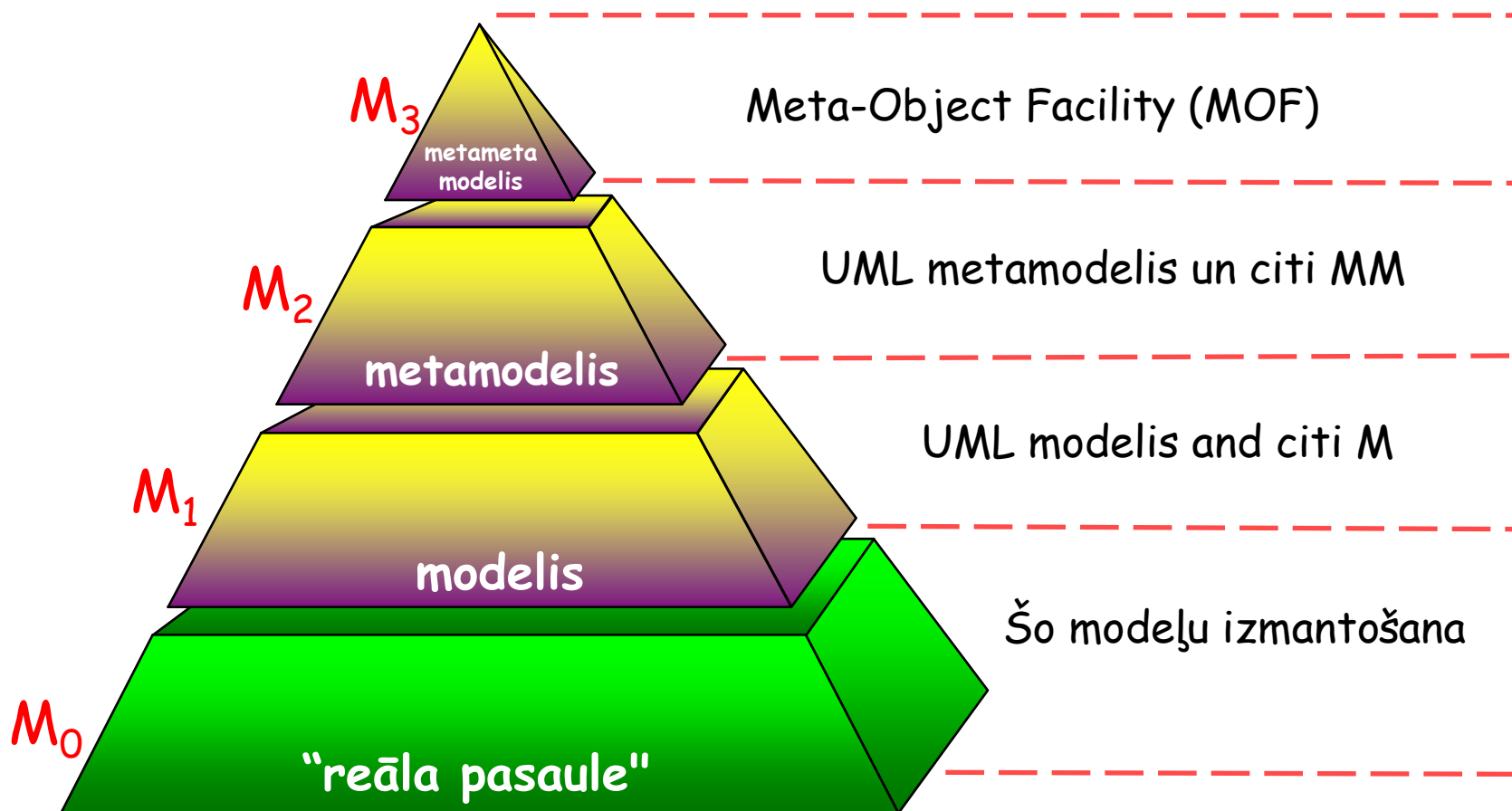
- Metamodelis ir modelis, kas definē modelēšanas valodas *abstrakto sintaksi* un *statisku semantiku*
- Modelis ir metamodelļa eksemplārs



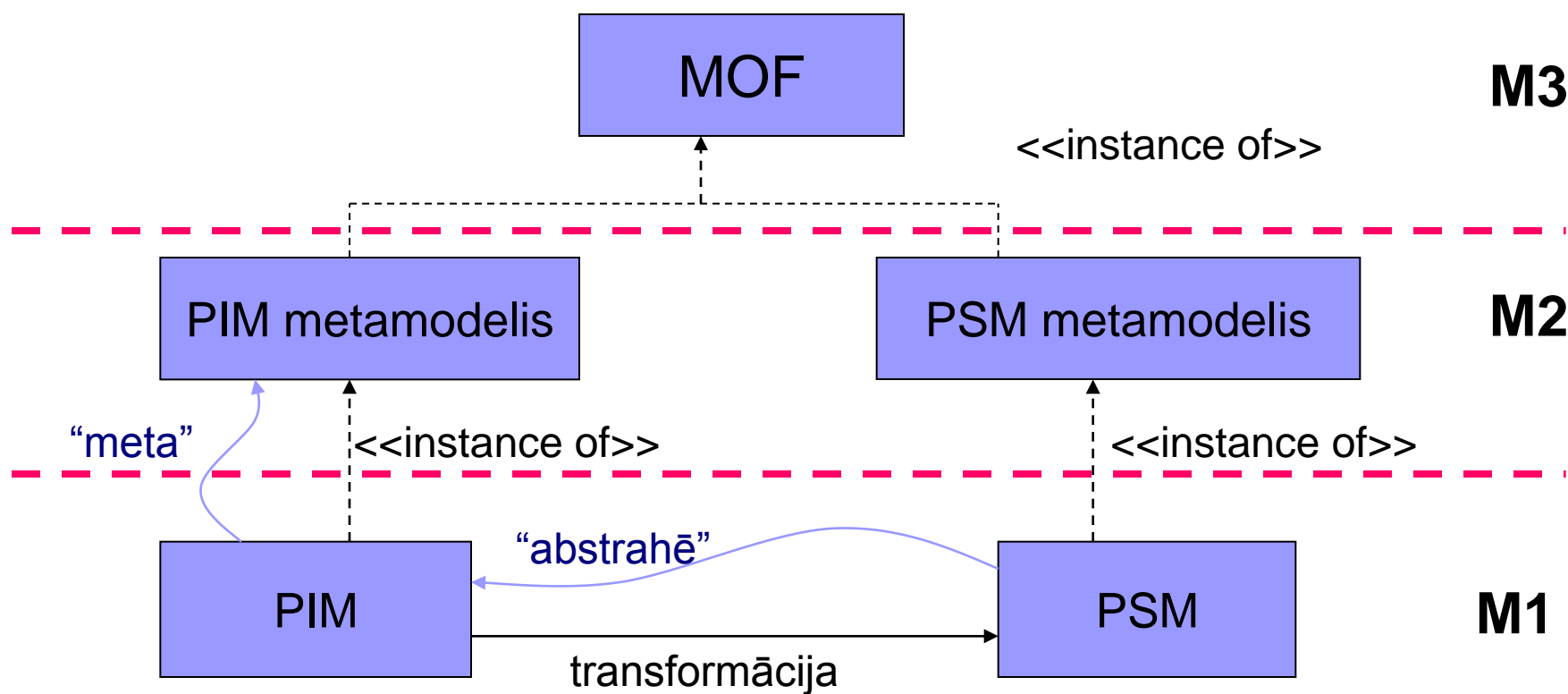
# MDA metamodelēšanas valoda

- Daudzos gadījumos UML valodas nespecificē visus sistēmas aspektus, un
- MDA neprasa, lai būtu izmantota tikai UML
- MDA prasa, lai modelēšanas valodas metamodelis būtu definēts MOF (Meta Object Facility, Metaobjektu vide/spējas) valodas terminos

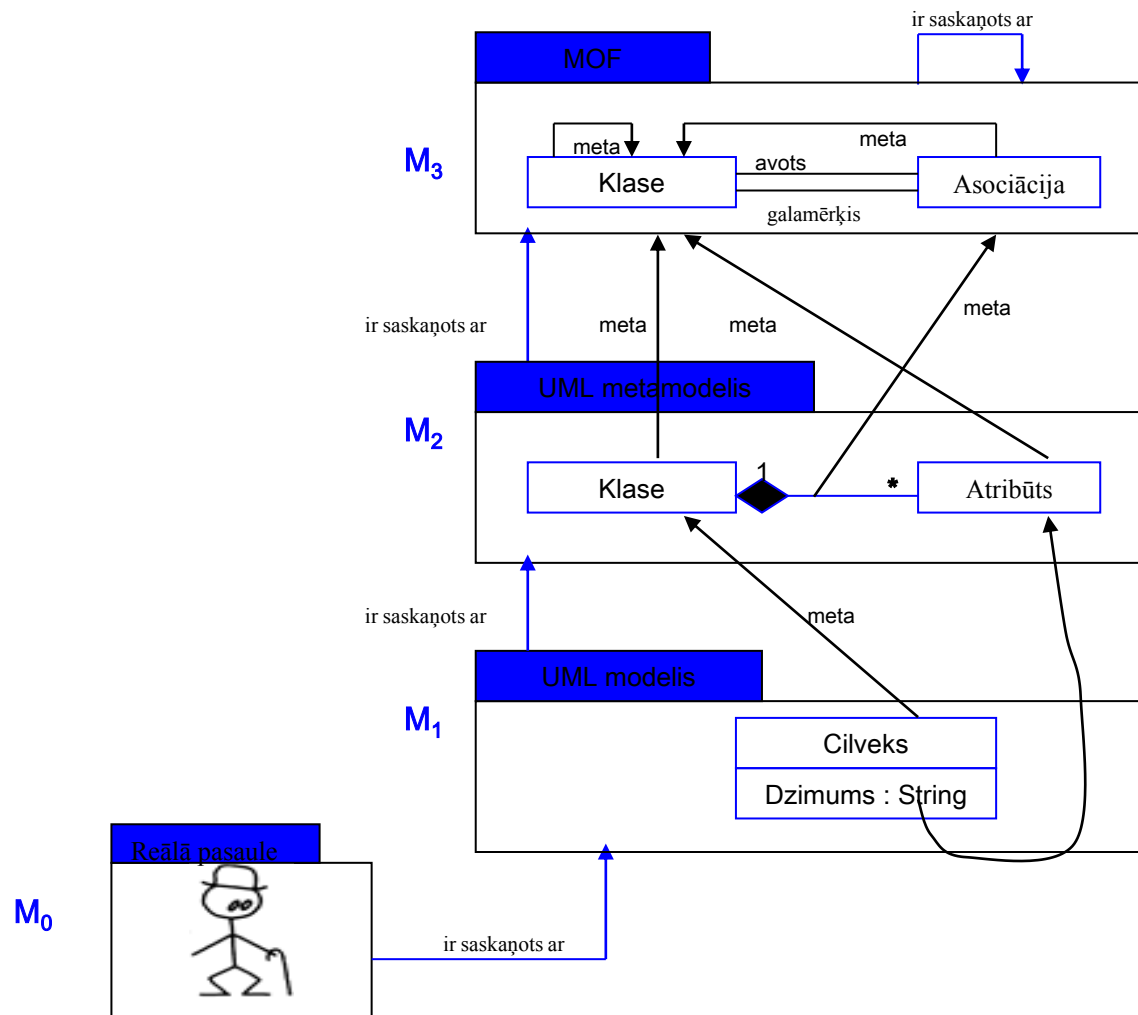
# Atmiņas atsvaidzināšanai: OMG metamodeļu arhitektūra



# Metalīmeņi salīdzinājumā ar abstrakcijas līmeņiem



# Meta līmeņi



MOF metadatu arhitektūra

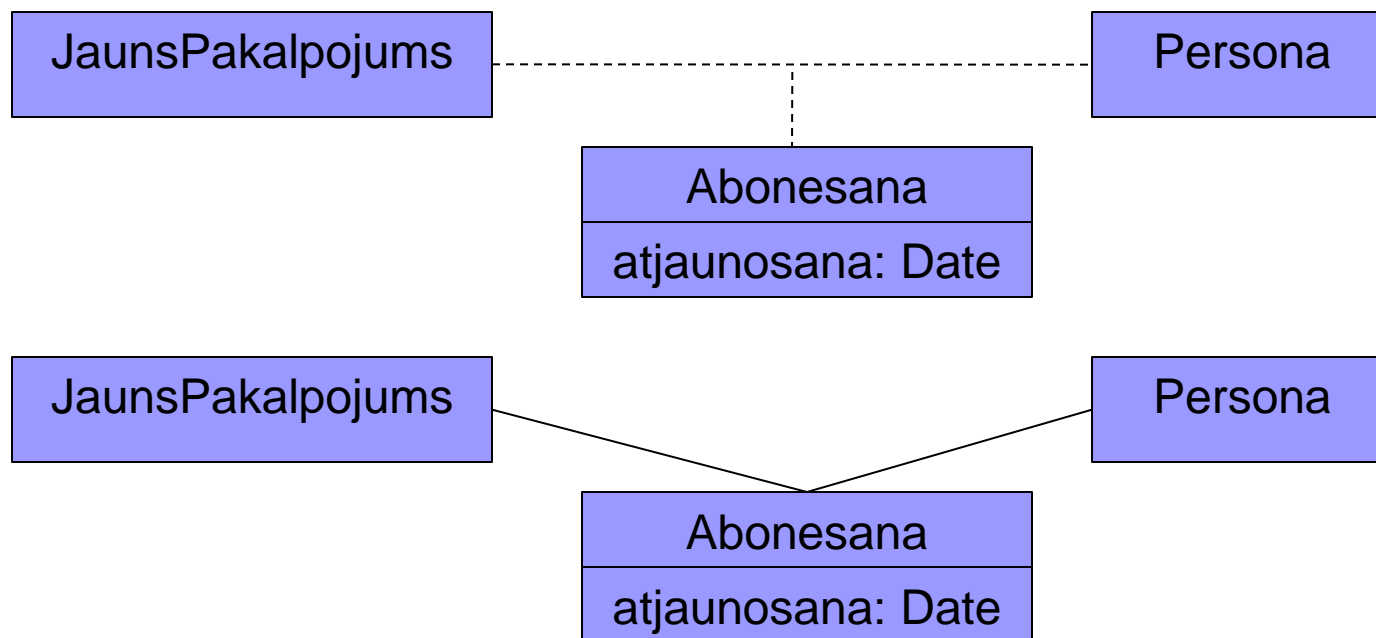
# Meta Object Facility (MOF)

- MOF ir valoda modelēšanas valodu definēšanai
  - OMG standarta metamodelēšanas valoda
  - UML metamodelis ir definēts MOF valodā
- MOF evolūcijā ir vairākas versijas
  - MOF 1.x ir visplašāk atbalstīts izstrādes rīkos
  - MOF 2.4 Beta2 ir spēkā esošais standarts, kuru būtiski ietekmēja UML 2.0 (MOF ir ISO/IEC 19502:2005 standarts)
  - MOF 2 ir kritiski svarīgs transformāciju atbalstam, piemēram, QVT un Model-to-text
    - MOF 2.0 Facility and Object Lifecycle
    - MOF Models to Text Transformation Language
- MOF 2 un UML 2 ir kopējais kodols
  - Vienkāršāki likumi metadatu modelēšanai
  - Vienkāršāk kartēt no/uz MOF
  - Metamodelēšanas plašs rīku atbalsts (var būt izmantots jebkurš rīks, kas atbalsta UML 2)



# UML konstrukcijas, kuras MOF valoda neatbalsta

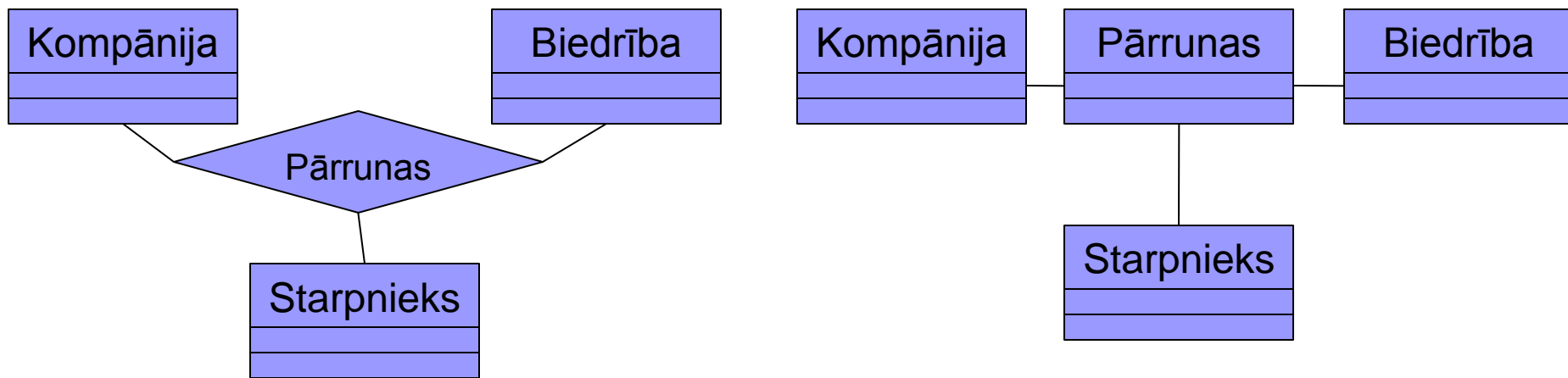
## ■ Asociāciju klase



## ■ Kvalifikatorus

# UML konstrukcijas, kuras MOF valoda neatbalsta (turp.)

- N-āras asociācijas (izmanto tikai bināras asociācijas)

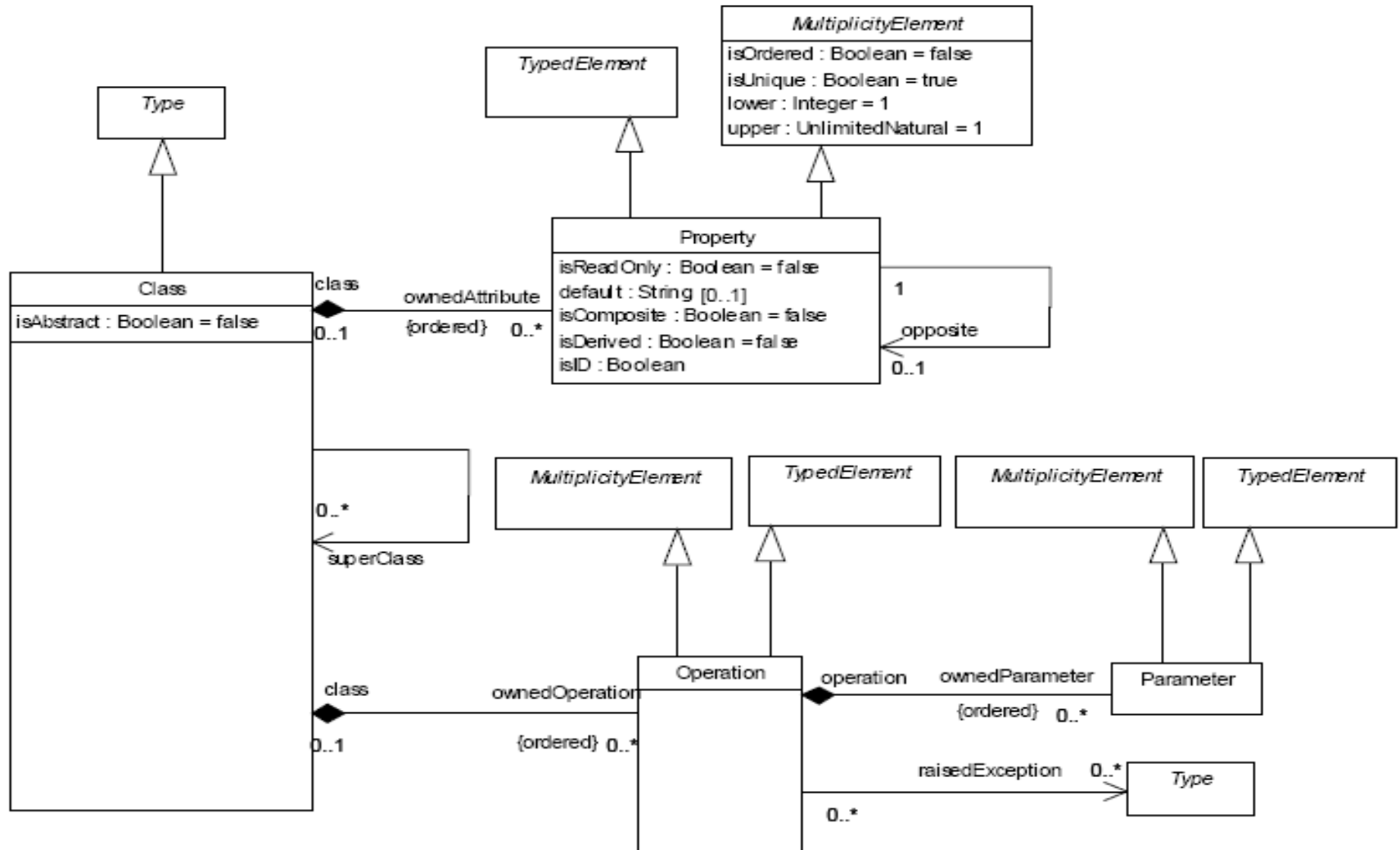


- Atkarības

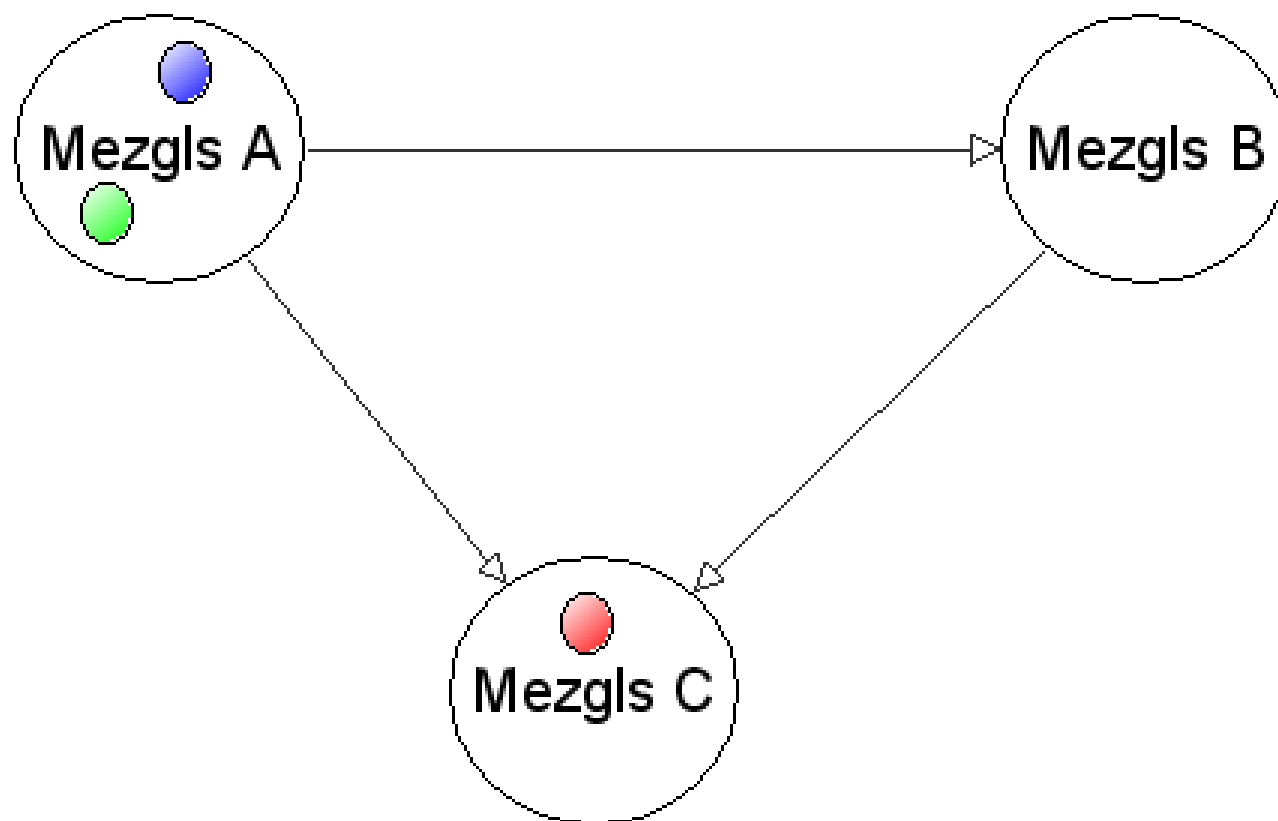
# MOF 2.0. struktūra

- MOF ir sadalīts divās daļās: Essential MOF (EMOF) un Complete MOF (CMOF)
- EMOF attiecas uz jēdzieniem, kuri atrodas OOP un XML ziņā
  - Tādēļ var viegli kartēt EMOF modeļus uz Java Metadata Interface, XMI (XML Metamodel Interchange), un citiem transformācijas standartiem
- CMOF tiek pielietots, lai aprakstītu metamodeļus tādām valodām kā UML 2
  - Tas ir uzkonstruēts no EMOF un UML 2 kodola konstrukcijām
  - Realitātē, EMOF un CMOF pamatā ir UML 2 valodas varianti

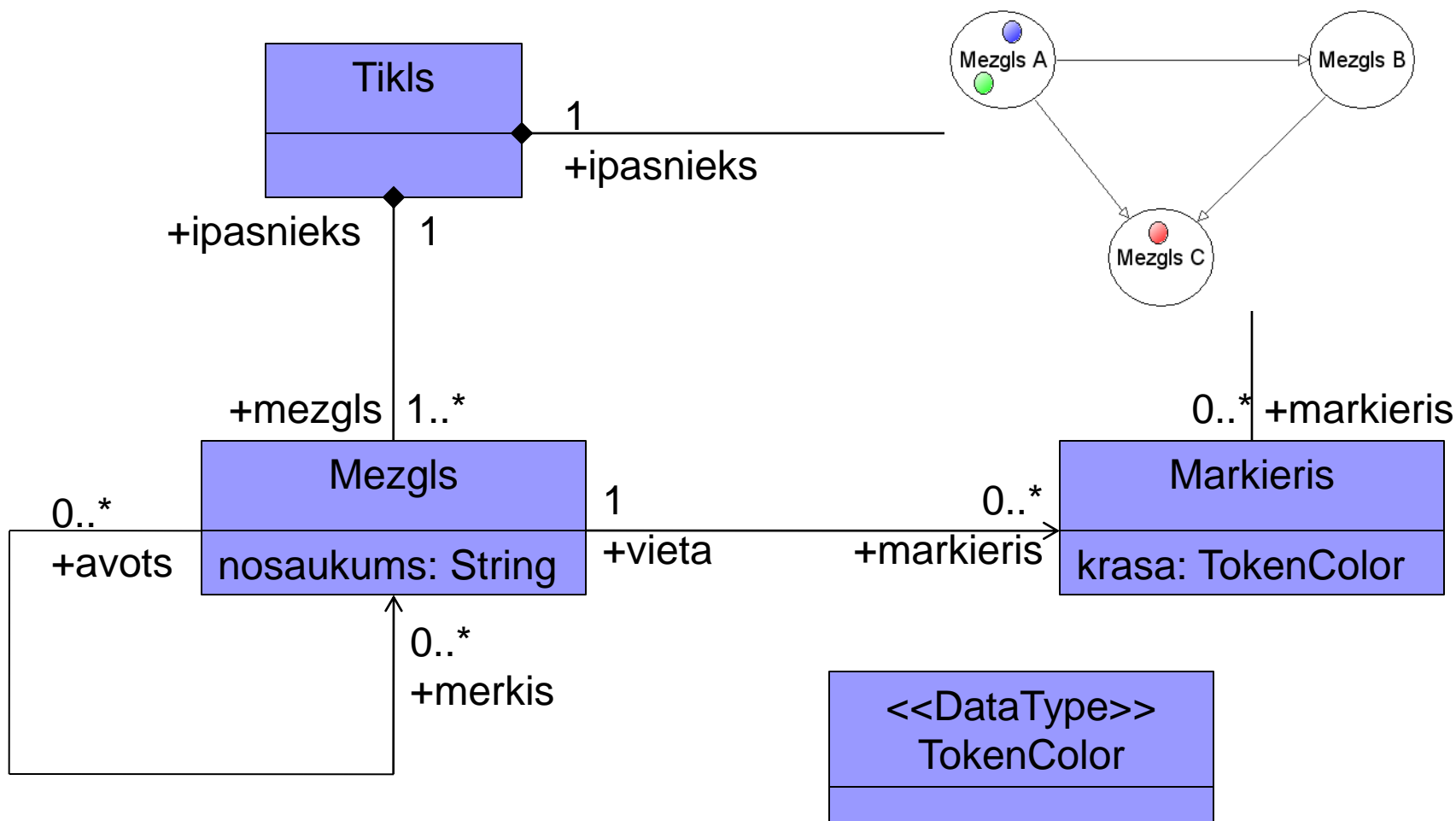
# EMOF struktūra



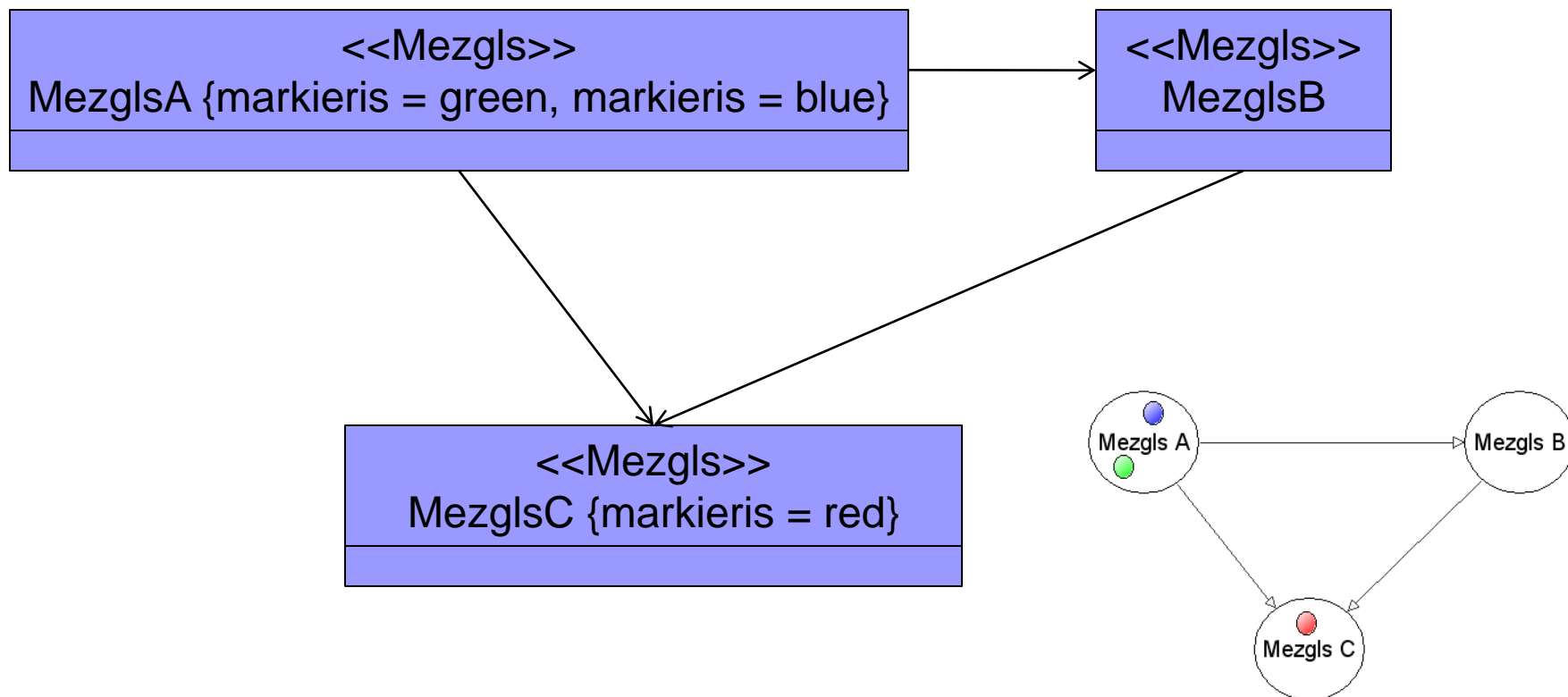
# Piemērs: grafa metamodelēšana



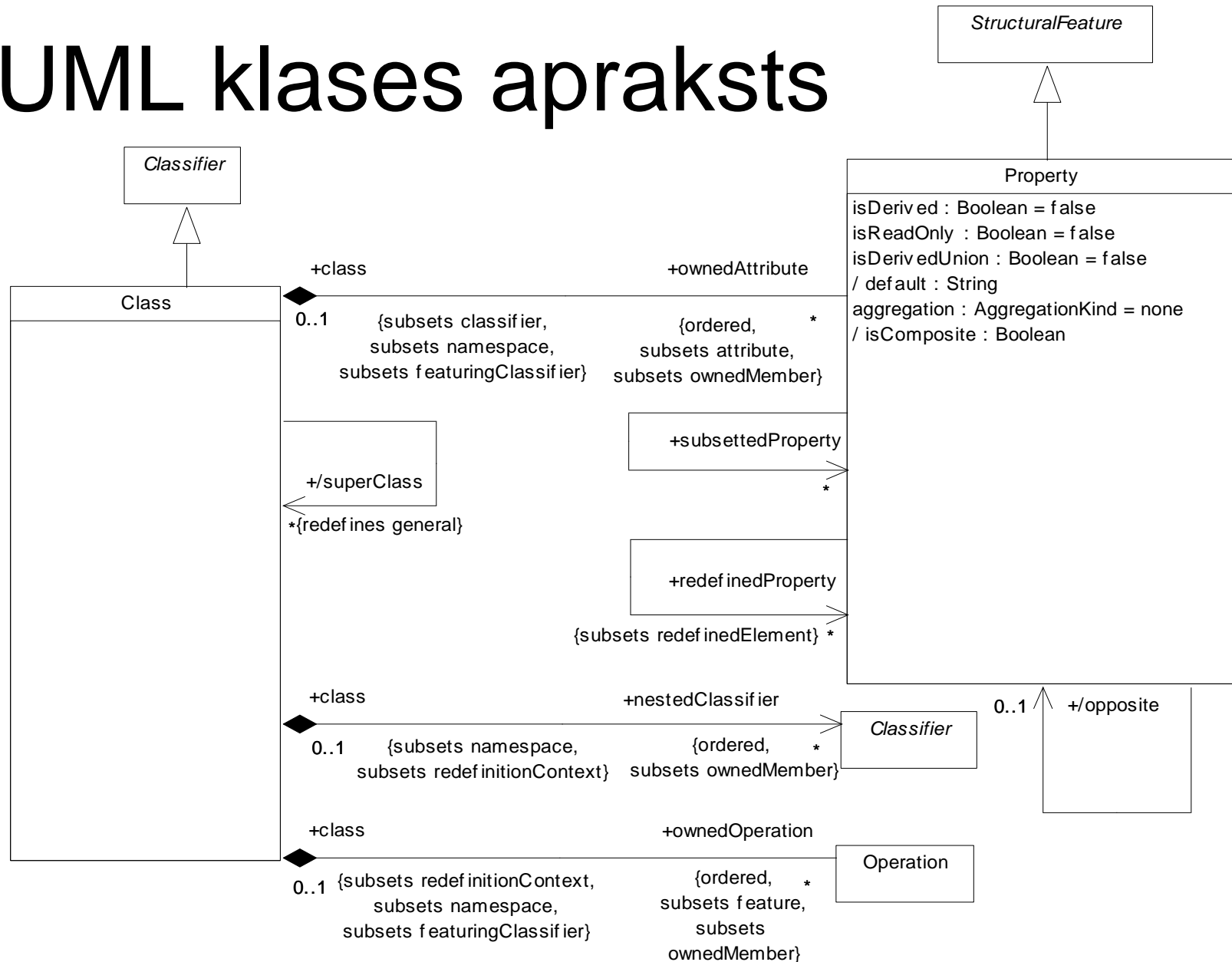
# Piemērs: grafa metamodelēšana



# Piemērs: grafa metamodelēšana

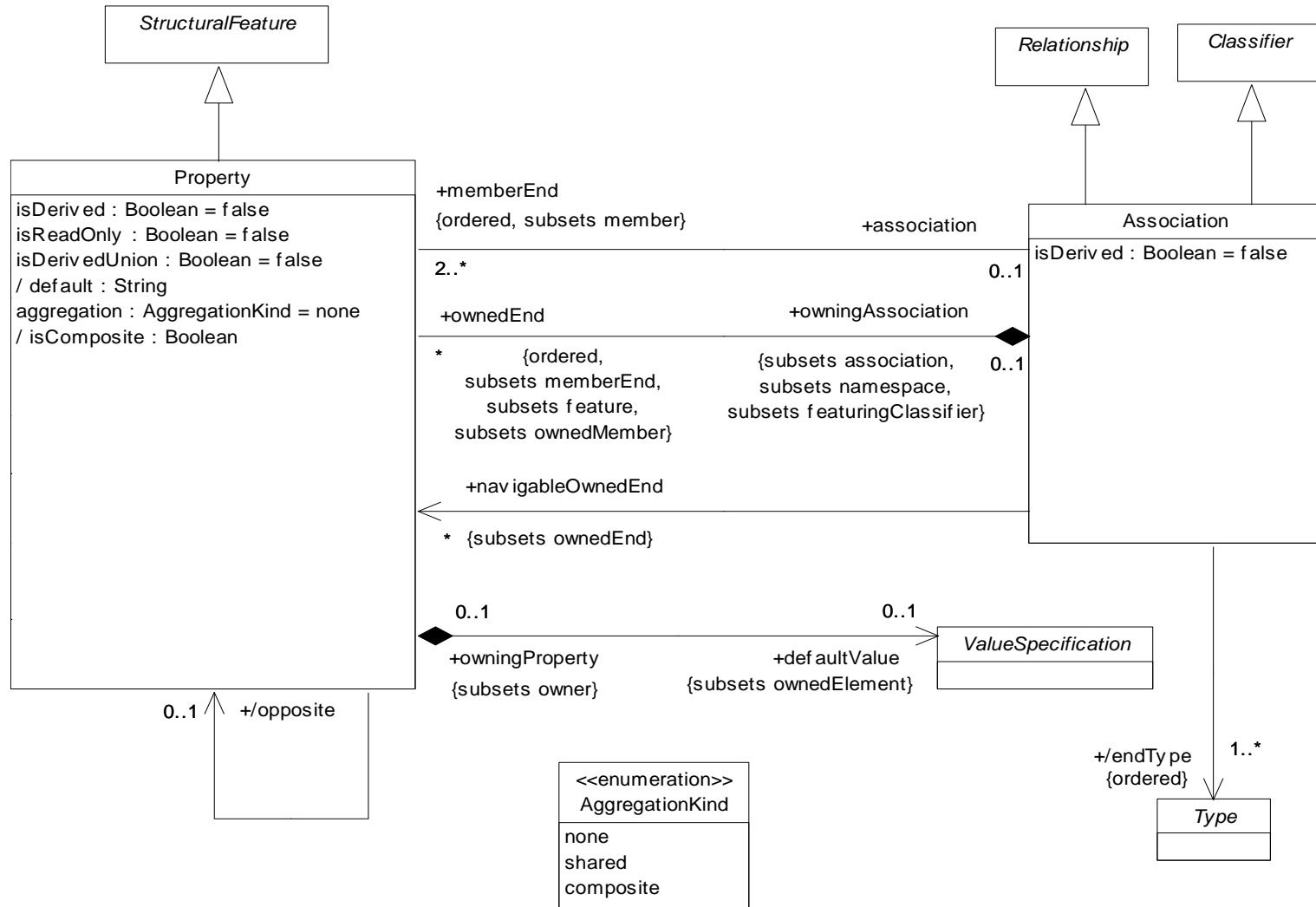


# UML klases apraksts

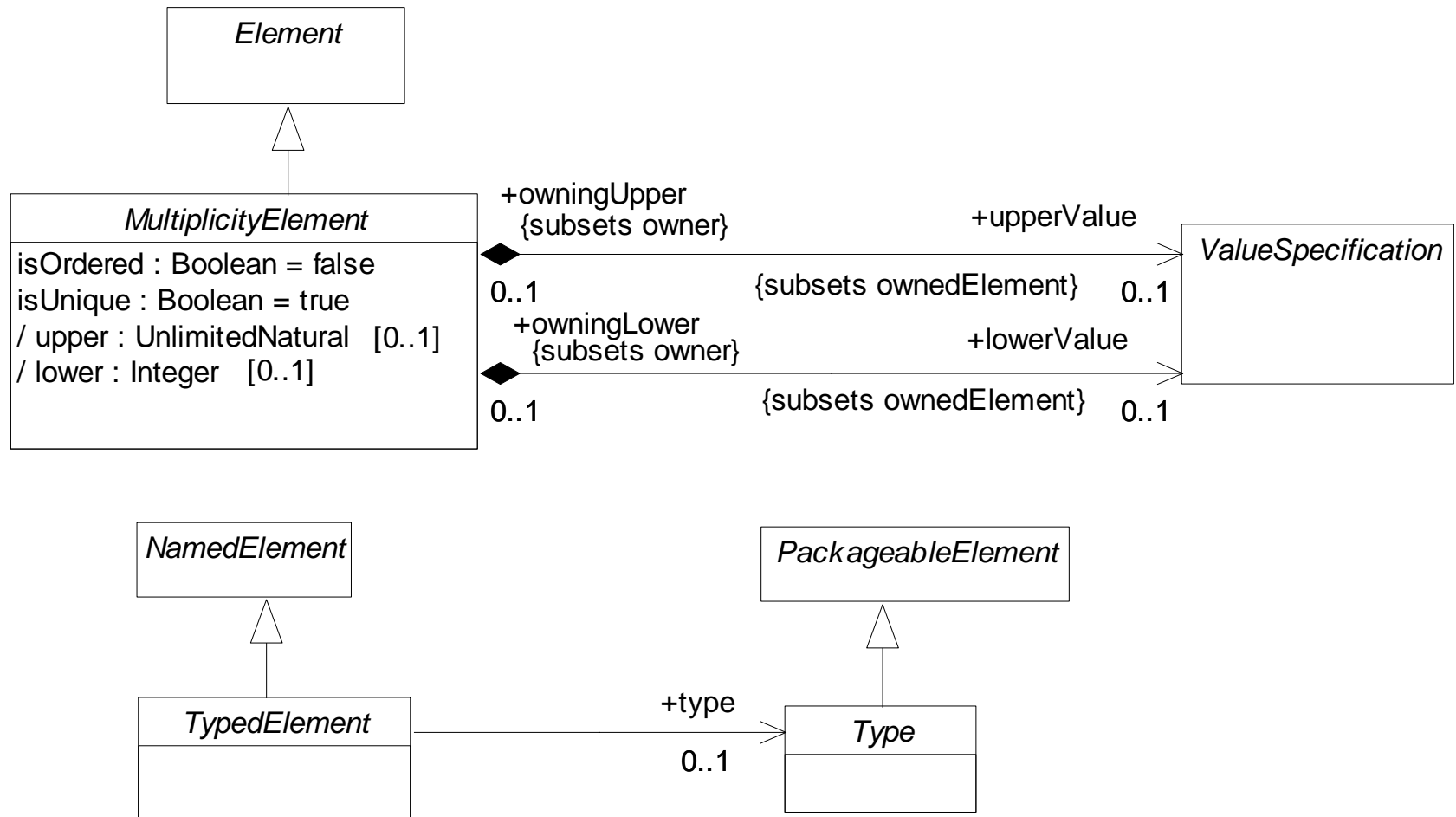




# UML klases apraksts (turp.)



# Multiplicity



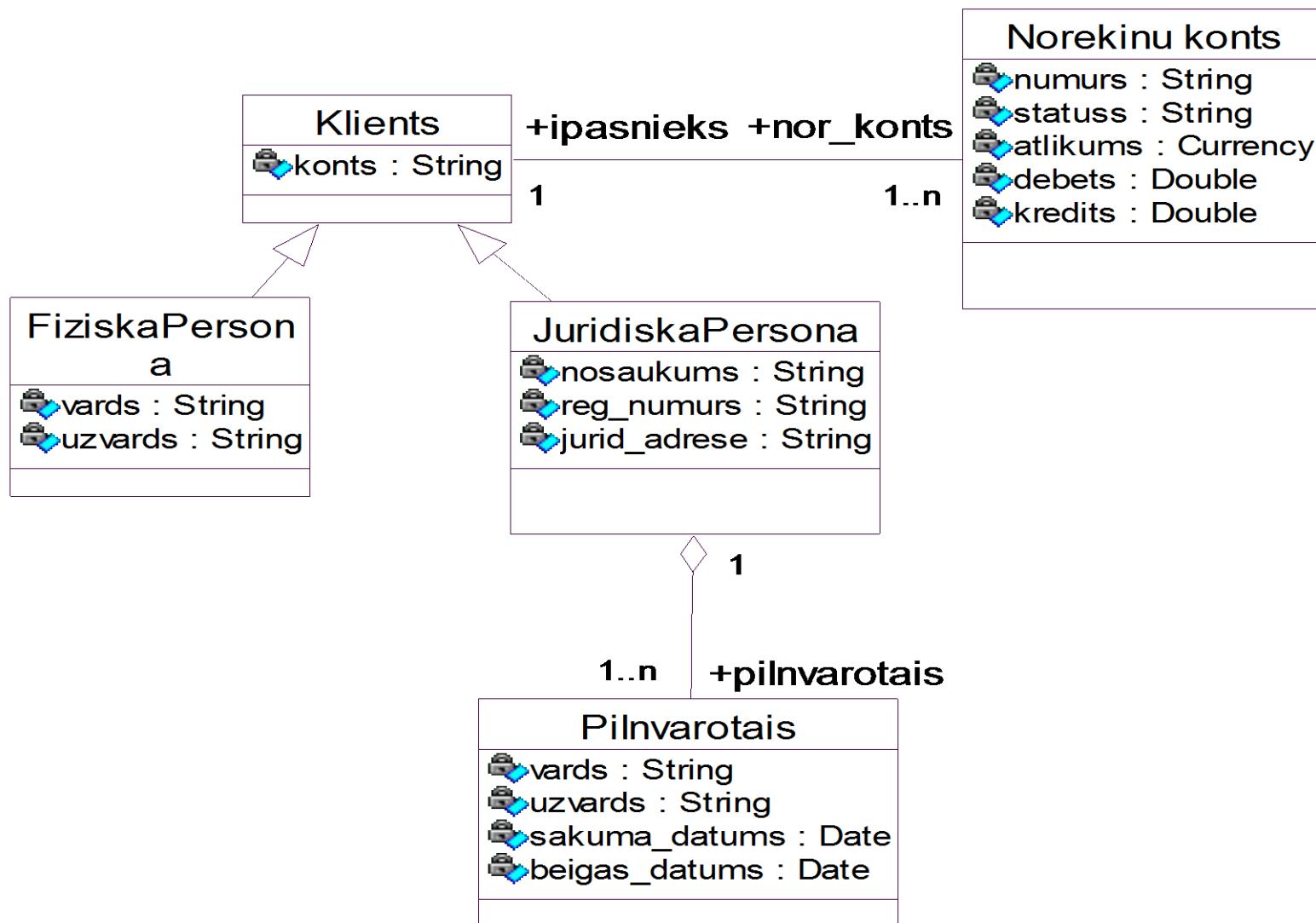
# M2 līmenis: M1 diagrammas UML metamodelis

- Virsotne – UML::Class
- Attiecības:
  - Agregācija, Kompozīcija – UML::AggregationKind
  - Asociācija – UML::Association
  - Atkarība – UML::Dependency
  - Vispārinājums – UML::Generalization

# UML::Class

Īpašība	Skaidrojums
/superClass:Class[*]	Klases superklase
ownedAttribute:Property[*]	Klases atribūti
ownedOperation:Operation[*]	Klases operācijas
isAbstract : Boolean	Ja klasei nevar būt eksplēplāri, tad <i>true</i> . Pēc noklusēšanas - <i>false</i>
/inheritedMember : NamedElement[*]	Apraksta visus elementus (atribūtus un operācijas), kas ir mantoti no superklases

# M1 līmenis: UML klašu diagramma



# UML::Class apraksta piemēri

Īpašība	JuridiskaPersona
/superClass:Class[*]	“Klients”
ownedAttribute:Property[*]	nosaukums :String, reg_numurs : String, jur_adrese : String <b>pilnvarotais:Pilnvarotais[1..n]</b>
ownedOperation:Operation[*]	----
isAbstract : Boolean	false
/inheritedMember : NamedElement[*]	konts : String, <b>/nor_konts:Norekinukonts[1..n]</b>

# UML::Class apraksta piemēri

Īpašība	Norekinukonts
/superClass:Class[*]	nav
ownedAttribute:Property[*]	numurs : String, statuss : String, atlikums : Currency, debets : Double, kredits : Double <i>/ipasnieks:Klients [1]</i>
isAbstract : Boolean	false
/inheritedMember : NamedElement[*]	nav

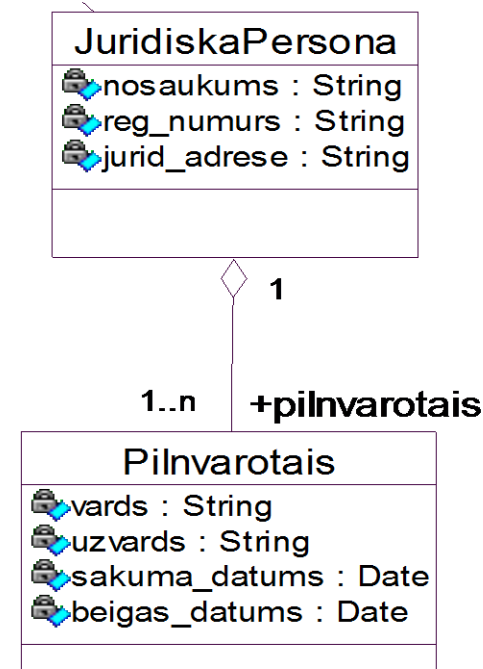
# UML::Property

Īpašība	Skaidrojums
visibility:VisibilityKind[1] = “private”	Atribūta redzamība – private, public, protected
isDerived : Boolean = false	Vai tas ir atvasināts atribūts. Pēc noklusēšanas <i>false</i>
isReadOnly:Boolean = false	Vai atribūta vērtība var būt izmainīta. Pēc noklusēšanas <i>false</i>
aggregation:AggregationKind	Ja atribūts ir cits klasifikators, tad kādas ir tā attiecības ar doto klasi. Var būt <i>none</i> , <i>shared</i> , <i>composite</i>
owningAssociation:Association[*]	Kurai asociācijai pieder šis atribūts
MultiplicityElement.isOrdered : Boolean	Vai atribūta vērtības ir sakārtotas. Pēc noklusēšanas <i>false</i>
MultiplicityElement.isUnique : Boolean	Vai atribūta vērtībām jābūt unikālām. Pēc noklusēšanas <i>false</i> .
MultiplicityElement./upper : UnlimitedNatural[0..1]	Vērtības augšējā robeža
MultiplicityElement./lower : Integer[0..1]	Vērtības apakšēja robeža
Type : DataType	Atribūta tipa nosaukums



# Piemērs: UML::Property






Īpašība	JuridiskaPersona. pilnvarotais
visibility:VisibilityKind[1] = "private"	public
isDerived : Boolean = false	<b>true</b>
isReadOnly:Boolean = false	false
aggregation:AggregationKind	shared
owningAssociation:Association[*]	Agregācija Pilnvarotais – Juridiska persona
MultiplicityElement.isOrdered : Boolean	false
MultiplicityElement.isUnique : Boolean	false
MultiplicityElement./upper : UnlimitedNatural[0..1]	n
MultiplicityElement./lower : Integer[0..1]	1
Type : Type	Pilnvarotais



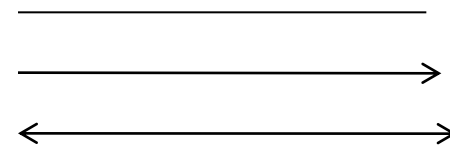
# Piemērs: UML::Property

Īpašība	NorekinuKonts. numurs
visibility:VisibilityKind[1] = "private"	private
isDerived : Boolean = false	false
isReadOnly:Boolean = false	false
aggregation:AggregationKind	none
owningAssociation:Association[*]	nav
MultiplicityElement.isOrdered : Boolean	<i>false</i>
MultiplicityElement.isUnique : Boolean	<i>true</i>
MultiplicityElement./upper : UnlimitedNatural[0..1]	1
MultiplicityElement./lower : Integer[0..1]	0
Type : Type	String

## Norekinu konts

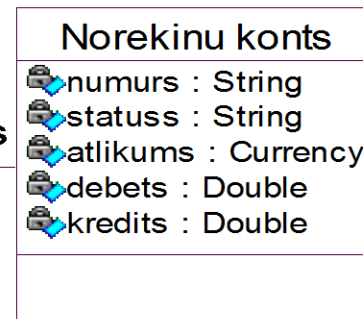
 numurs : String  
 statuss : String  
 atlikums : Currency  
 debets : Double  
 kredits : Double

# UML::Association



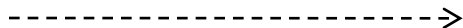
Īpašība	Skaidrojums
isDerived : Boolean	Norādā vai asociācija ir atvasināta no citas asociācijas vai ierobežojuma. Pēc noklusēšanas <i>false</i>
memberEnd : Property [2..*]	Klasifikatori saites galos
/endType : Type [1..*]	Atsaucas uz klasifikatoriem, kuri tiek izmantoti kā tipi asociācijas galos

# Asociācija



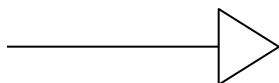
Īpašība	Asociācija
isDerived : Boolean	false
memberEnd : Property [2..*]	nor_konts:Norekinukonts [1..n], ipasnieks:Klients[1]
/endType : Type [1..*]	Norekinukonts, Klients
general : Classifier [1] (target)	_____
specific : Classifier [1] (source)	_____

## UML::Dependency (UML::DirectedRelationship)



Īpašība	Skaidrojums
client : NamedElement [1..*] (source)	Elements, kas ir atkarīgs no “piegādātāja” (avots)
Supplier : NamedElement [1..*] (target)	Elements, kas ir neatkarīgs no “klienta” (mērķa)

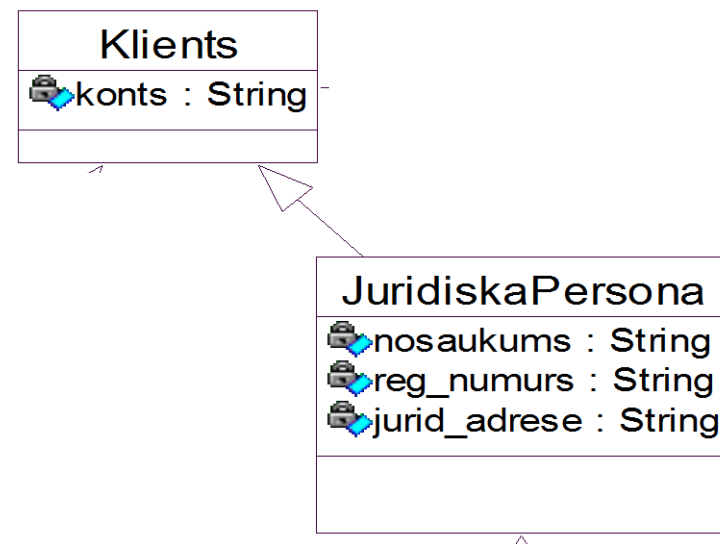
## UML::Generalization (UML::DirectedRelationship)



Īpašība	Skaidrojums
general : Classifier [1] (target)	Vispārīgs klasifikators
specific : Classifier [1] (source)	Specializēts klasifikators

# Vispārinājums UML::Generalization

Īpašība	Vispārinājums
isDerived : Boolean	———
memberEnd : Property [2..*]	———
/endType : Type [1..*]	———
general : Classifier [1] (target)	Klients
specific : Classifier [1] (source)	JuridiskaPersona



# UML::Operation

- Operācijas īpašnieks ir klase
- Operācija var būt izsaukta šīs klases objektu kontekstā
- Tas ir tipizēts (atgriežamais tips) un daudzkāršības (rezultāta daudzkāršība) elements

Īpašība	Apraksts
class:Class [0..1]	Klase, kurai pieder šī operācija
ownedParameter:Parameter[*] {ordered, composite}	Operācijas parametri
raisedException:Type[*]	Izņēmumi, kuri ir deklarēti kā iespējamie operācijas izsaukšanā

# Eclipse Modeling Framework: metamodel *Ecore*

