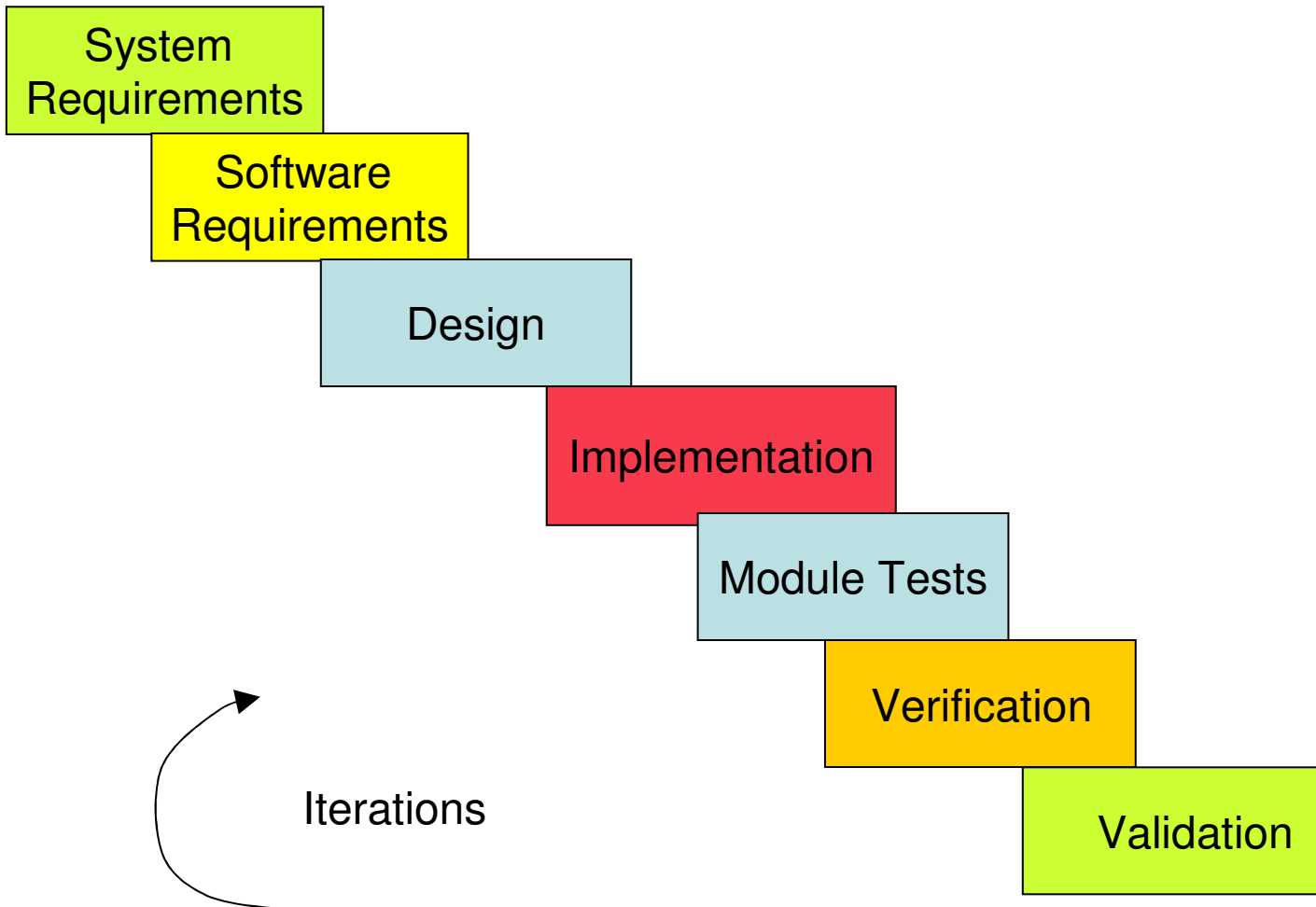


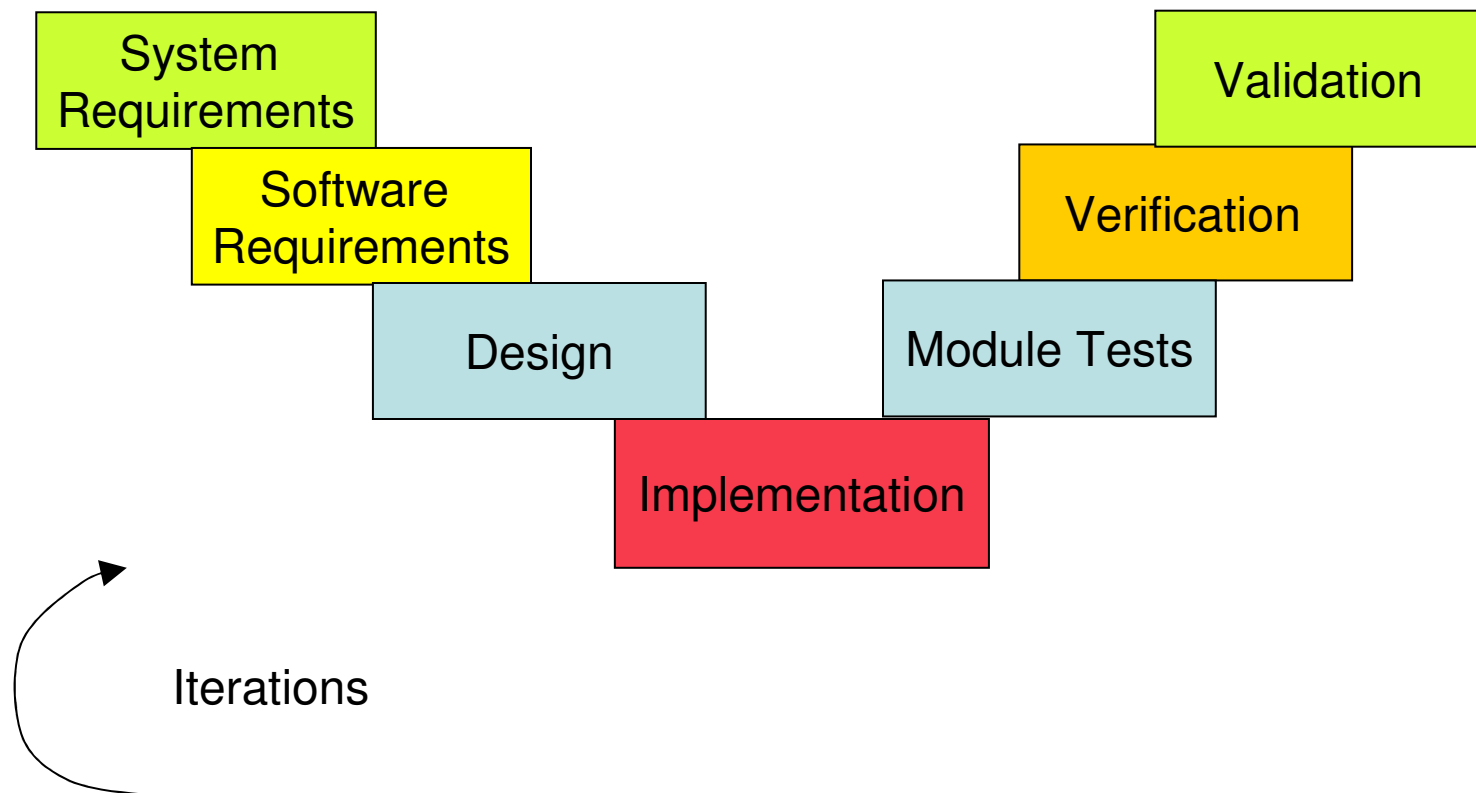
Objektově-orientovaná Analýza & Návrh

Analýza požadavků

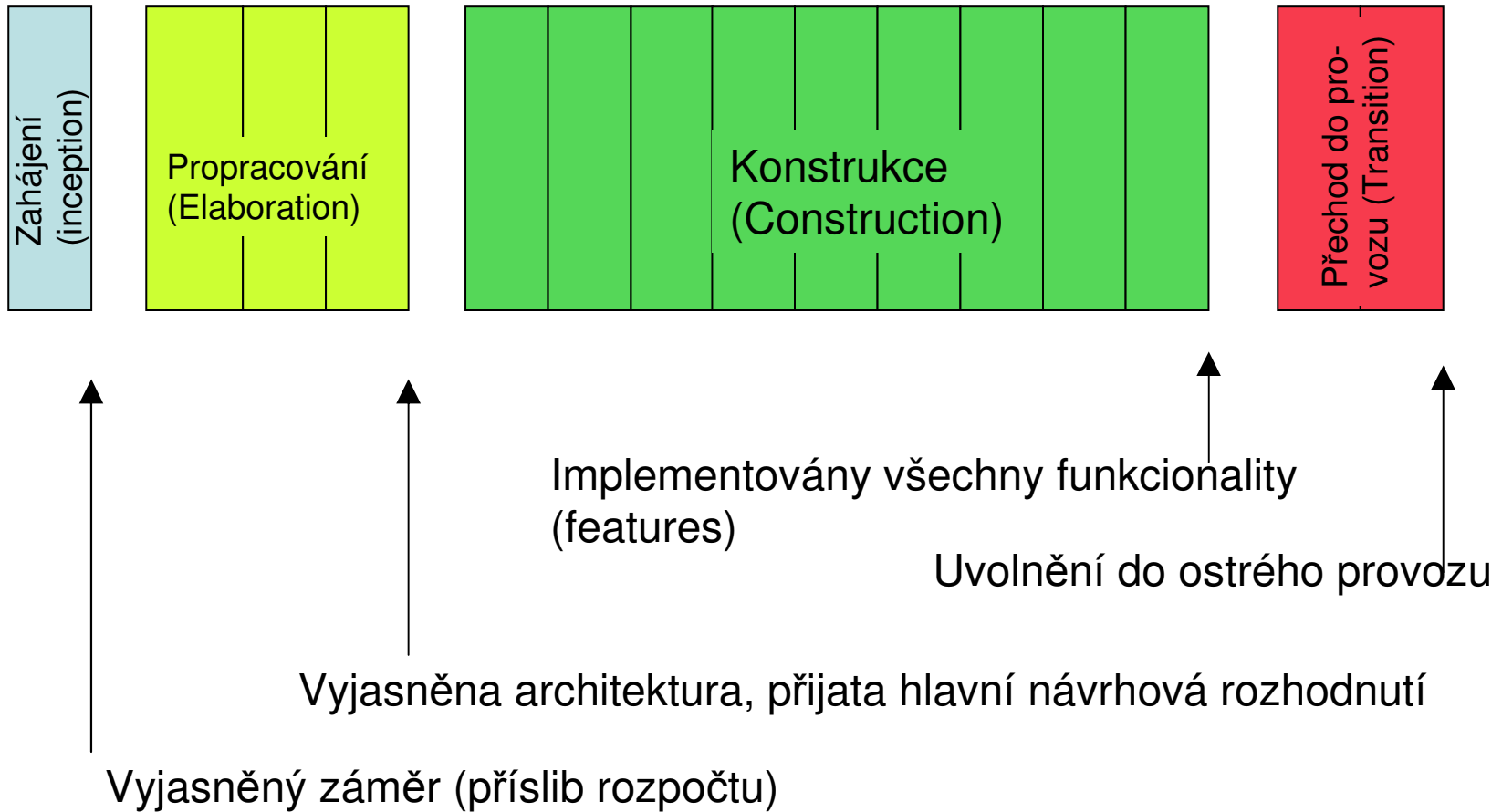
Vodopádový model (Waterfall Model)



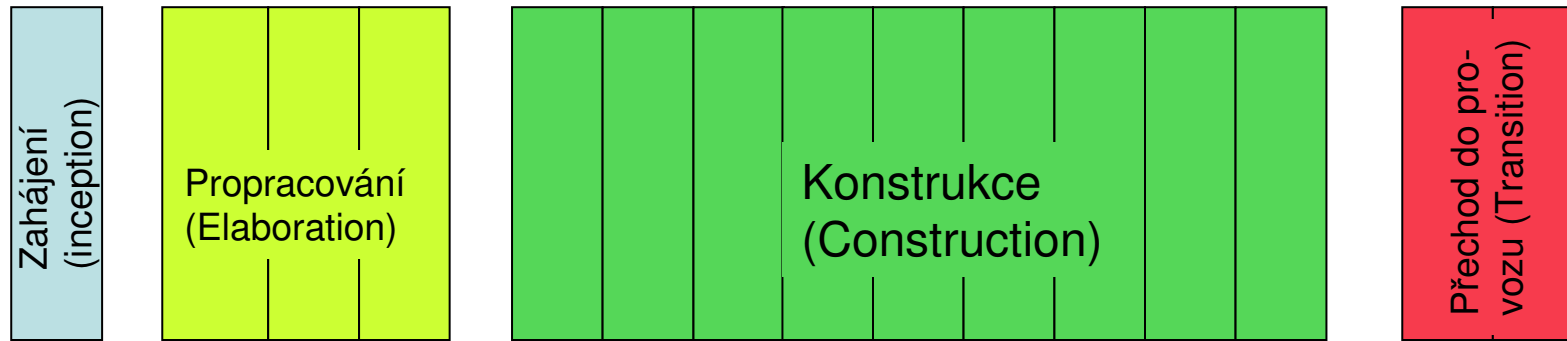
Vodopádový model (Waterfall Model)



Agilní unifikovaný proces (Agile Unified Process)



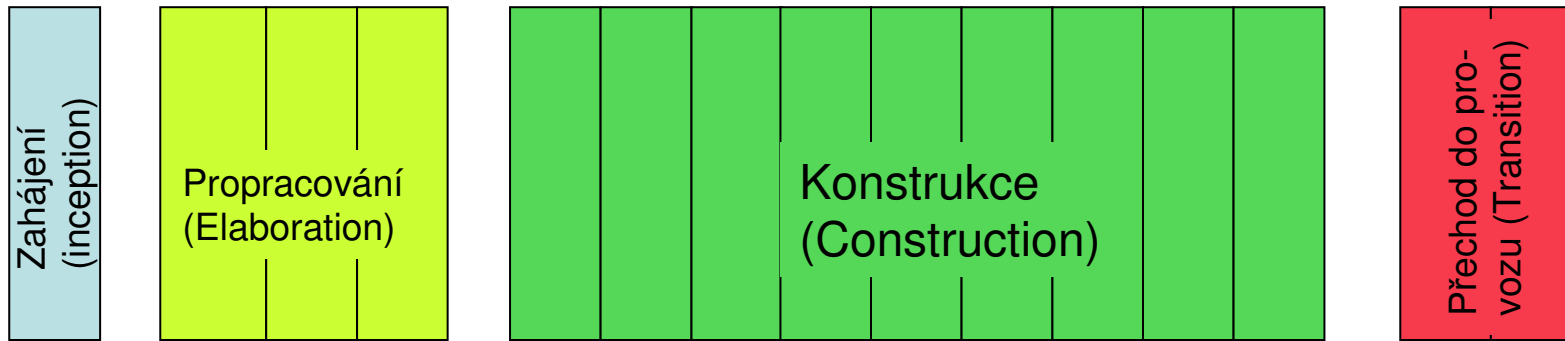
Agilní unifikovaný proces (Agile Unified Process)



Zahájení:

- hrubá vize vyvíjeného systému
- business case
- věcný rozsah (scope)
- hrubé základní požadavky
- hrubé odhady

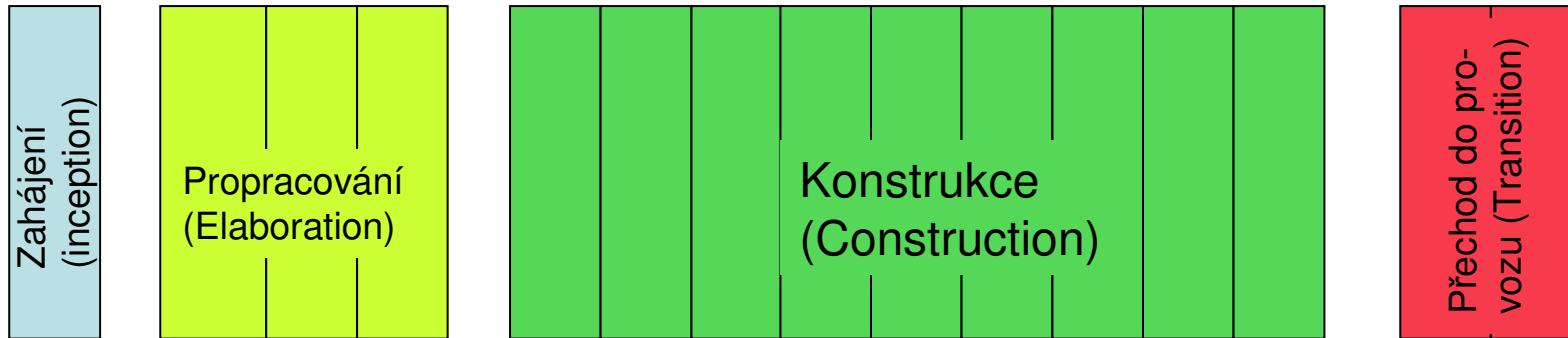
Agilní unifikovaný proces (Agile Unified Process)



propracování:

- podrobná vize vyvíjeného systému
- iterativní implementace základní (core) architektury
- iterativní implementace funkcionalit s vysokou úrovní rizika
- identifikace většiny požadavků
- identifikace věcného rozsahu (scope)
- realističtější odhady

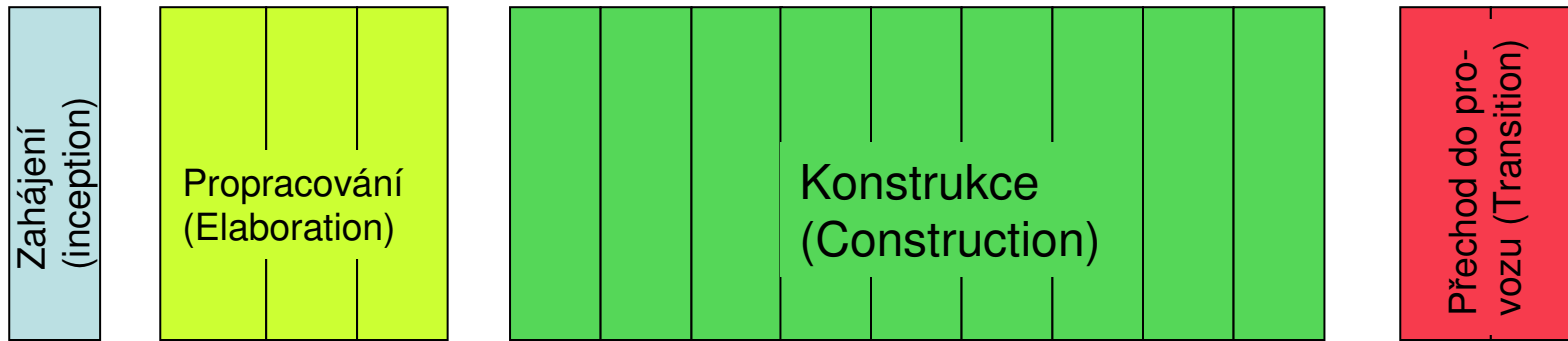
Agilní unifikovaný proces (Agile Unified Process)



Konstrukce:

- iterativní implementace zbývajících funkcionalit
- testy před nasazením

Agilní unifikovaný proces (Agile Unified Process)



Přechod do provozu:

- deployment
- beta tests
- release

OOA/D

UML se používá na třech úrovních abstrakce:

- **Konceptuální úroveň**

Diagramy reprezentují věci reálného světa (domain of interest)

Konceptuální (analytická) třída

Reprezentuje pojem (věc) reálného světa

- **Úroveň specifikace (softwaru)**

Diagramy reprezentují SW-ovou abstrakci nebo SW komponenty (s jejich interface) ale ne jejich konkrétní implementaci

Návrhová (SW) třída

Reprezentuje SW komponentu

- **Úroveň implementace (softwaru)**

Diagramy reprezentují SW-ovou abstrakci nebo SW komponenty (s jejich interface) ale ne jejich konkrétní implementaci

Implementační třída

Třída ve smyslu C++ nebo Java

OOA/D

Definuj
případy užití

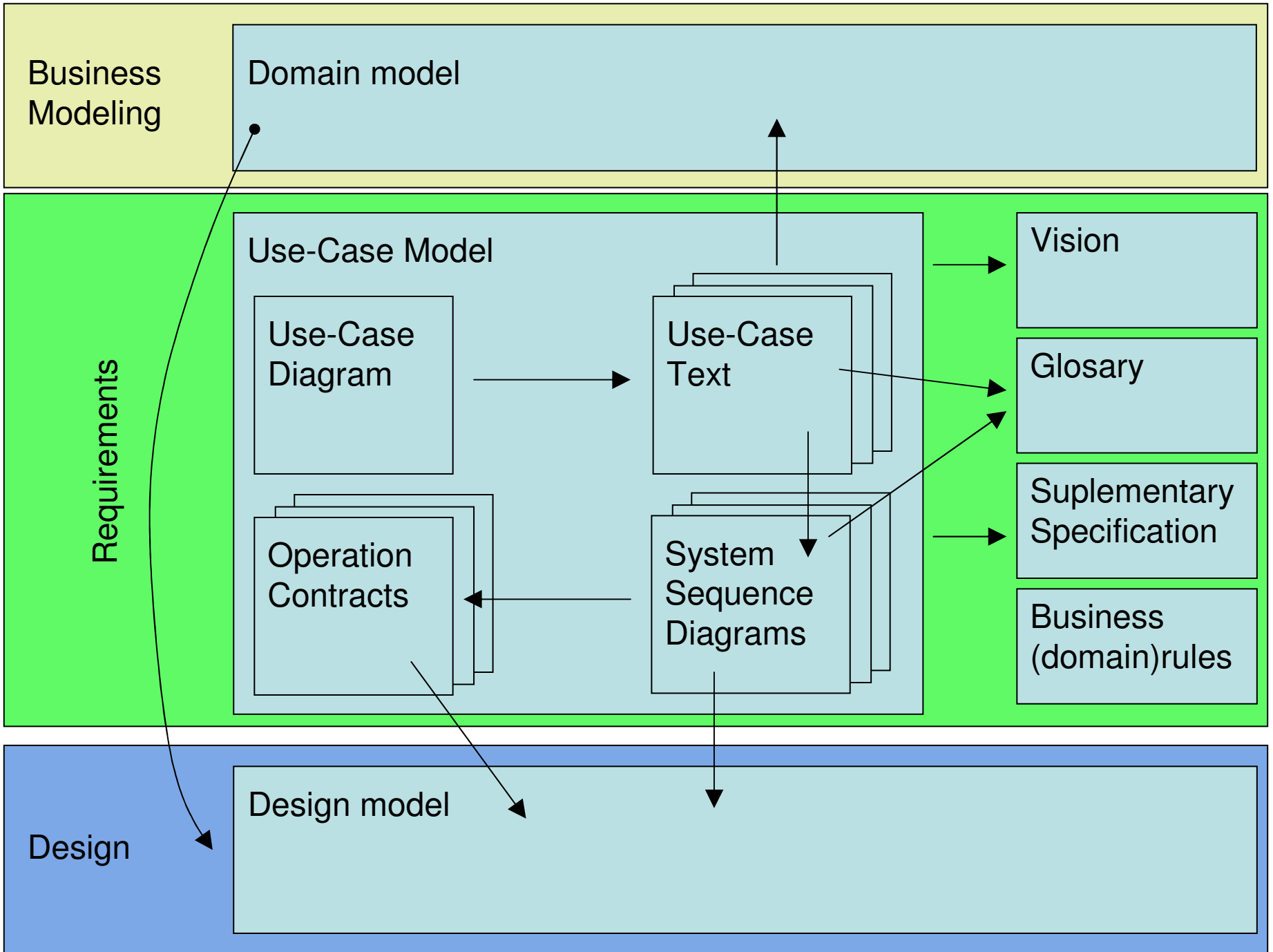
Definuj
doménový
model

Definuj
interakční
diagramy

Definuj
diagramy
návrhových tříd

Requirements (FURPS+)

- **Functional** (features, zabezpečení, ...)
- **Useability** (GUI, ergonomie, lidský faktor, help, dokumentace, ...)
- **Reliability** (přípustná frekvence chyb, obnovení z chyb, předvídatelnost,...)
- **Performance** (rychlost odezvy, průchodnost, dostupnost, spotř. paměti, ...)
- **Supportability** (GUI, ergonomie, lidský faktor, help, internacionalizace, dokumentace, ...)
- + **Implementation** (limity zdrojů, jazyky a nástroje, platforma)
Interface (požadavky a omezení vyvolané externími systémy)
Operations (management systému, konfigurovatelnost)
Packaging
Legal



Discipline	Artifact	Incept	Elab	Const	Trans
		I1	E1..En	C1..Cn	T1..T2
Business Modeling	Domain Model		s		
Requirements	Use-Case Model	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model		s	r	
	SW Architecture Document		s		
	Data Model		s	r	

Příklady užití (use-cases) jsou textové dokumenty, nikoliv diagramy.

Modelování příkladů užití (use-case modeling) je primárně psaní textu, nikoliv kreslení diagramů.

Kreslení use-case diagramů má pomoci identifikovat jednotlivé use-casy a nalézt jejich strukturu

3 obvyklé stupně rozpracování use-casů:

Brief

„Zákazník přijde na pokladnu s vybranými položkami sortimentu. Pokladní použije pokladní systém k tomu, aby zaevidoval každou prodanou položku. Systém průběžně zobrazuje průběžný součet cen a detaily o jednotlivých položkách. Zákazník poskytne platební kartu, kterou systém ověří a zaeviduje. Systém upraví stav zásob na skladě. Zákazník obdrží účtenku a odejde s nákupem.“

Casual

Podrobnější popis, více odstavců. Tak, jak postupně roste stupeň porozumění danému use-casu.

Fully dressed

Podrobně popsány všechny kroky a varianty.
Podpůrné sekce jako pre-conditions a post-conditions

Fully dressed use case

Sekce	Komentář
Jméno use-casu	Mělo by začínat slovesem
Rozsah (Scope)	Název systému, který je předmětem návrhu
Úroveň	User-goal nebo subfunction
Primární aktér	Ten, kdo vyvolává služby systému pro splnění svých cílů
Stakeholders and interests	Kdo je zainteresován na use-casu a co od něj očekává
Pre-conditions	Co musí být splněno, aby se use case mohl vykonat
Post-conditions	Co je splněno po úspěšném splnění use-casu
Hlavní (úspěšný) scénář	Typický (nepodmíněný) úspěšný scénář (průběh).
Extenze	Alternativní scénáře (úspěšné nebo chybové)
Speciální požadavky	Související nefunkční požadavky
Změny technologie a dat	Seznam změn technologie a formátu dat
Frekvence výskytu	Může mít vliv na prioritu
Různé	Typicky seznam nevyjasněných otázek

Fully dressed use case

Hlavní scénář

1. Zákazník přijde na pokladnu se zbožím.
2. Pokladní zahájí nový prodej.
3. Pokladní zadá (čarový) kód zboží.
4. Systém zaznamená položku prodeje a zobrazí popis zboží, cenu a průběžnou částku.
Pokladní opakuje kroky 3-4 pro všechny položky zboží.
5. Systém zobrazí celkovou částku včetně DPH.
6. Zákazník zaplatí a systém zpracuje platbu.
7. Systém předá informaci o uskutečněném prodeji a platbě externímu účetnímu systému
8. Systém vydá účtenku
9. Zákazník odchází se zbožím.

Fully dressed use case

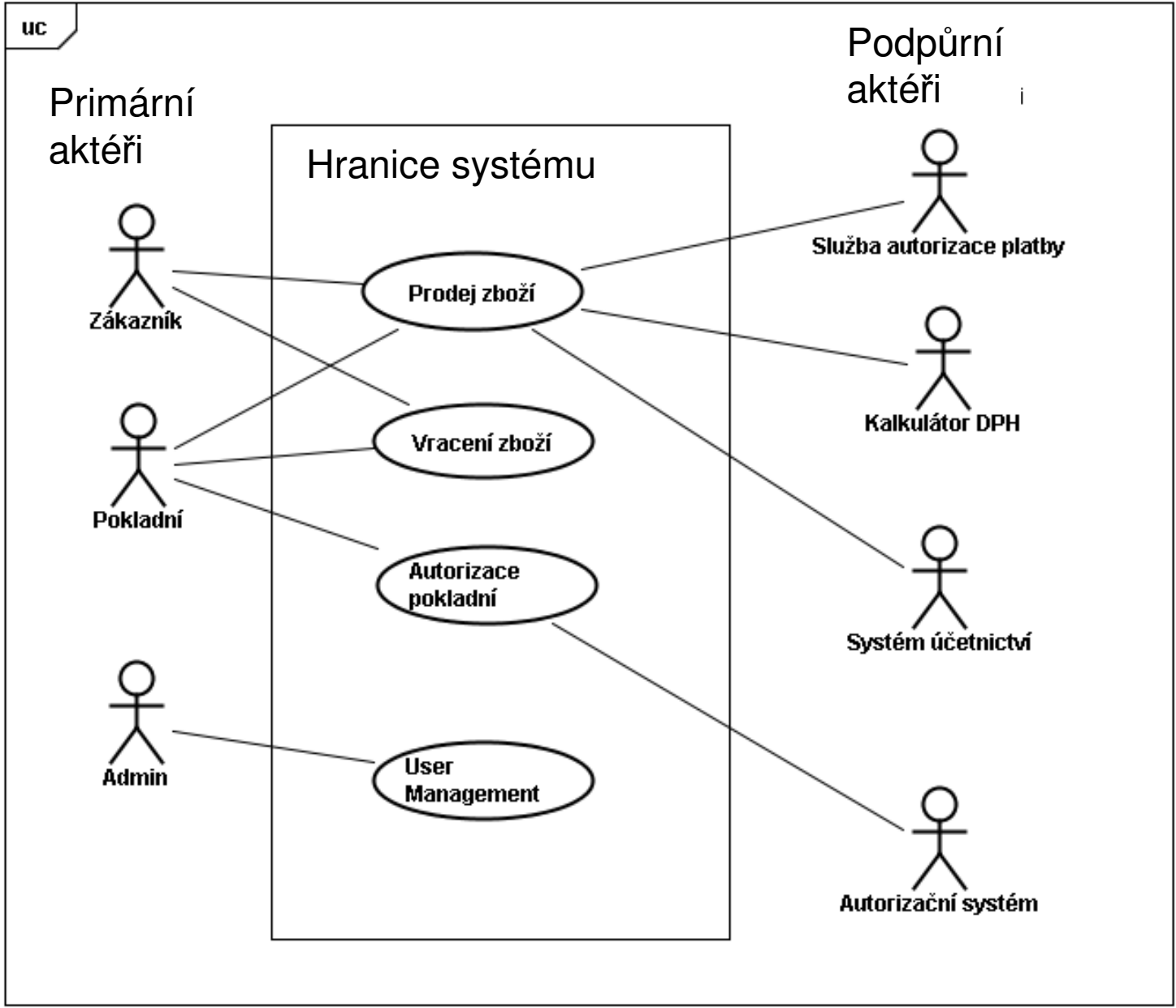
Extenze (alternativní scénáře)

- *a. Kdykoliv. Manažer provádí opravnou operaci
 - 1, Manažer zadá do systému svůj autorizační kód.
 2. Manažer nebo pokladní provede jednu operaci v manažerském módu.
 3. Systém se vrátí do módu pokladní

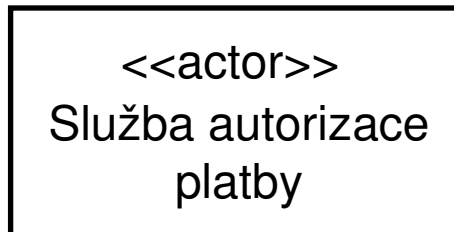
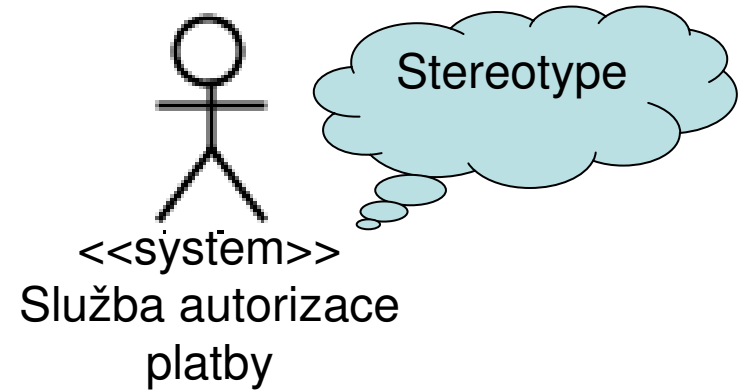
- 2-4a. Zákazník oznámí pokladní, že má nárok na slevu.
 1. Pokladní nárok ověří a zadá kód slevy.
 2. System vypočte výši uznané slevy.

Jak identifikovat případy užití

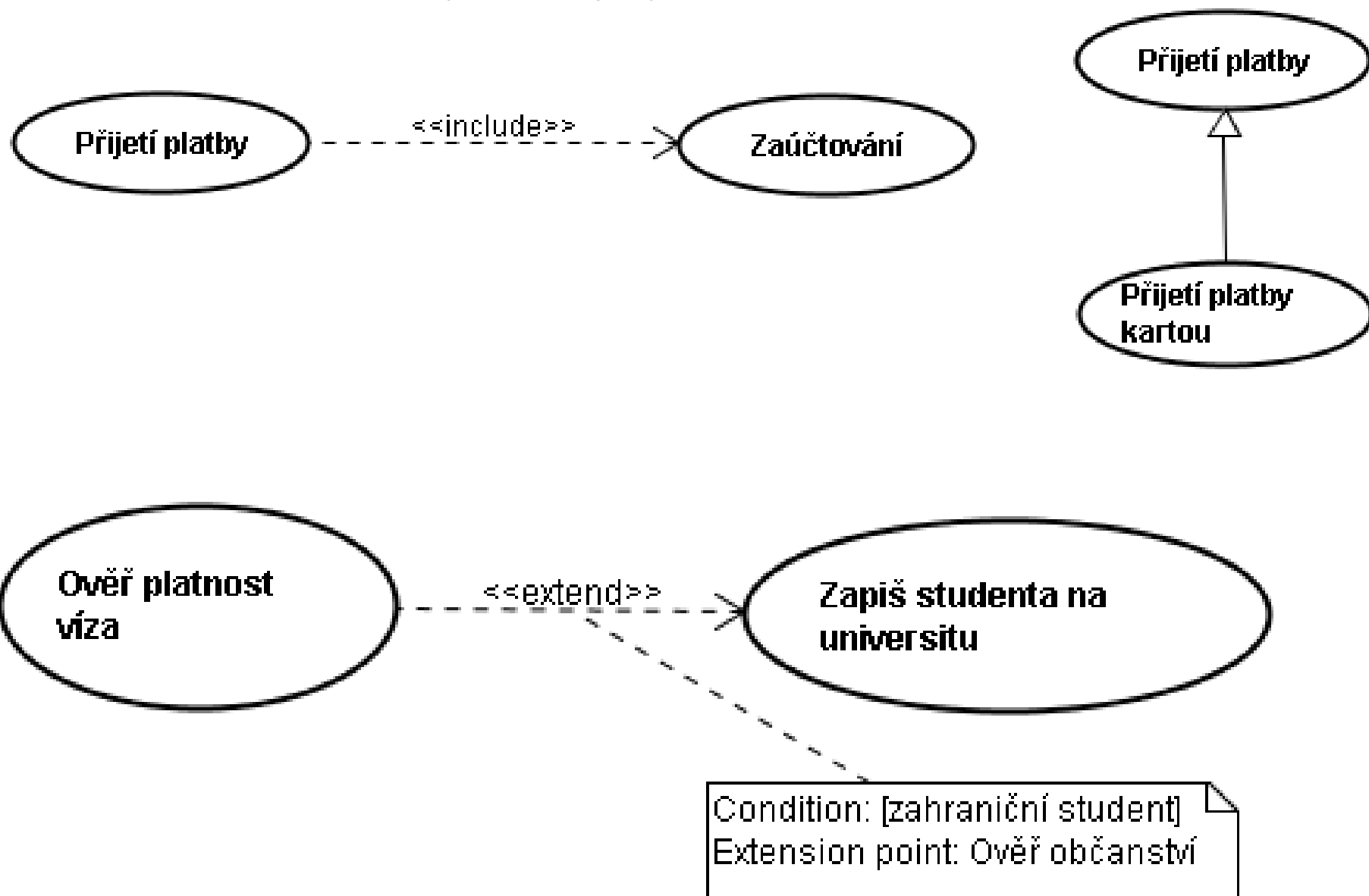
1. Identifikace hranic systému.
2. Nalezení primárních aktérů a jejich cílů.
3. Analýza systémových událostí (vyžadujících obsluhu).
4. Definice případů užití (za pomoci use-case diagramů).

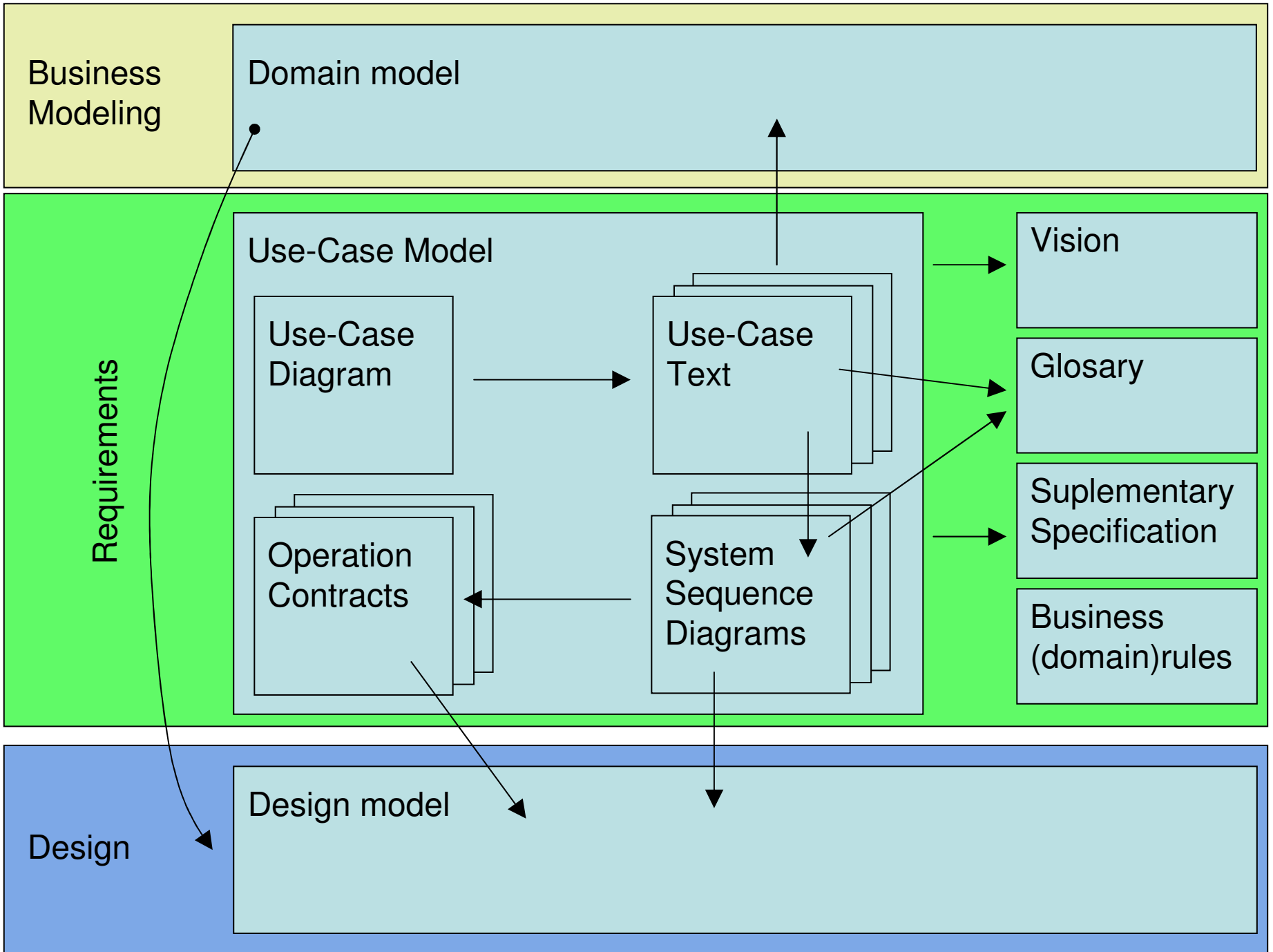


Alternativní notace aktéra



Jeden use-case využívá jiný use-case



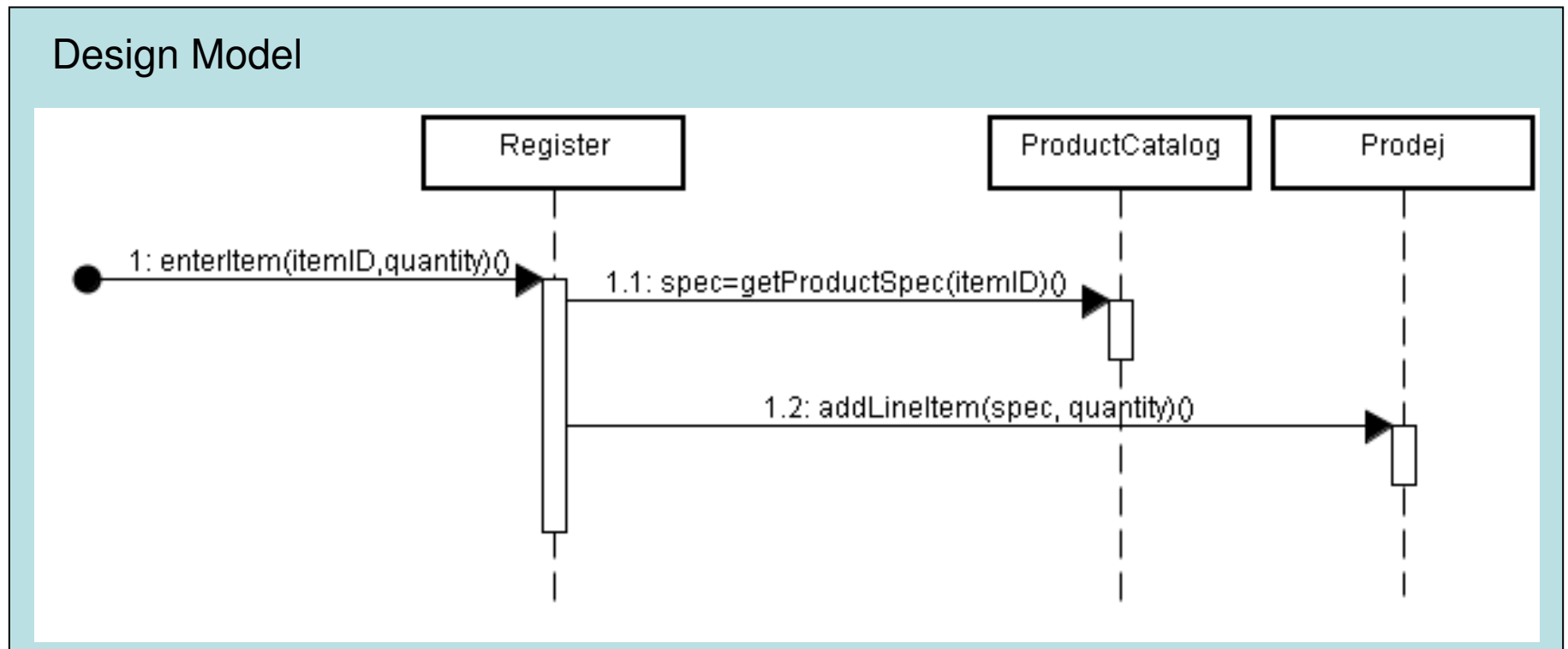
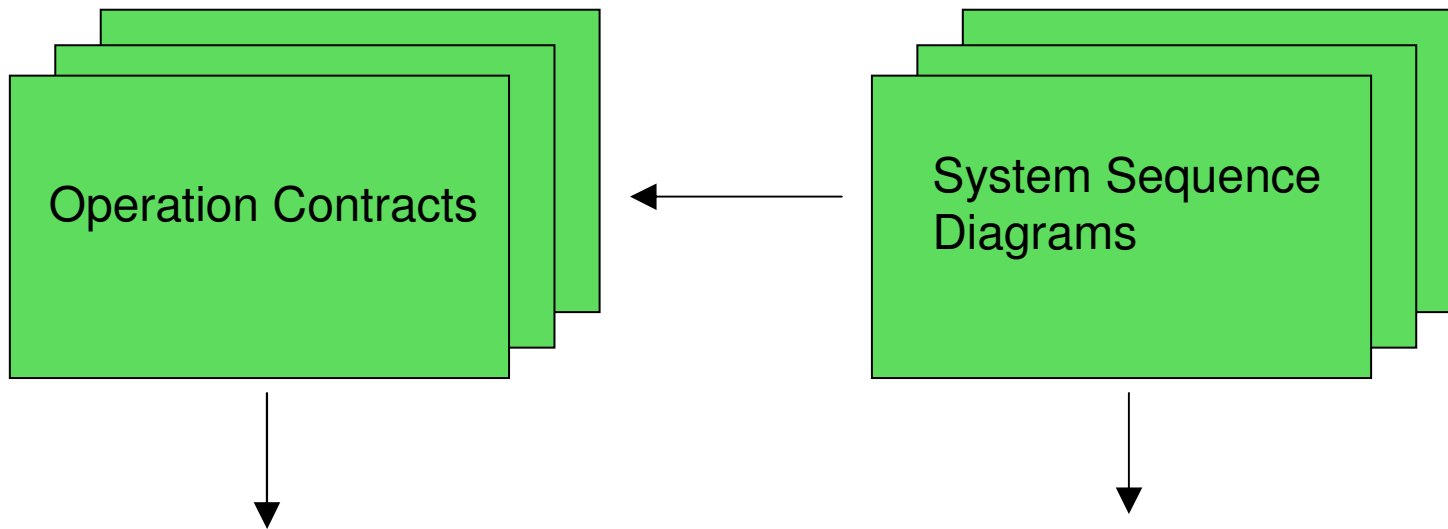


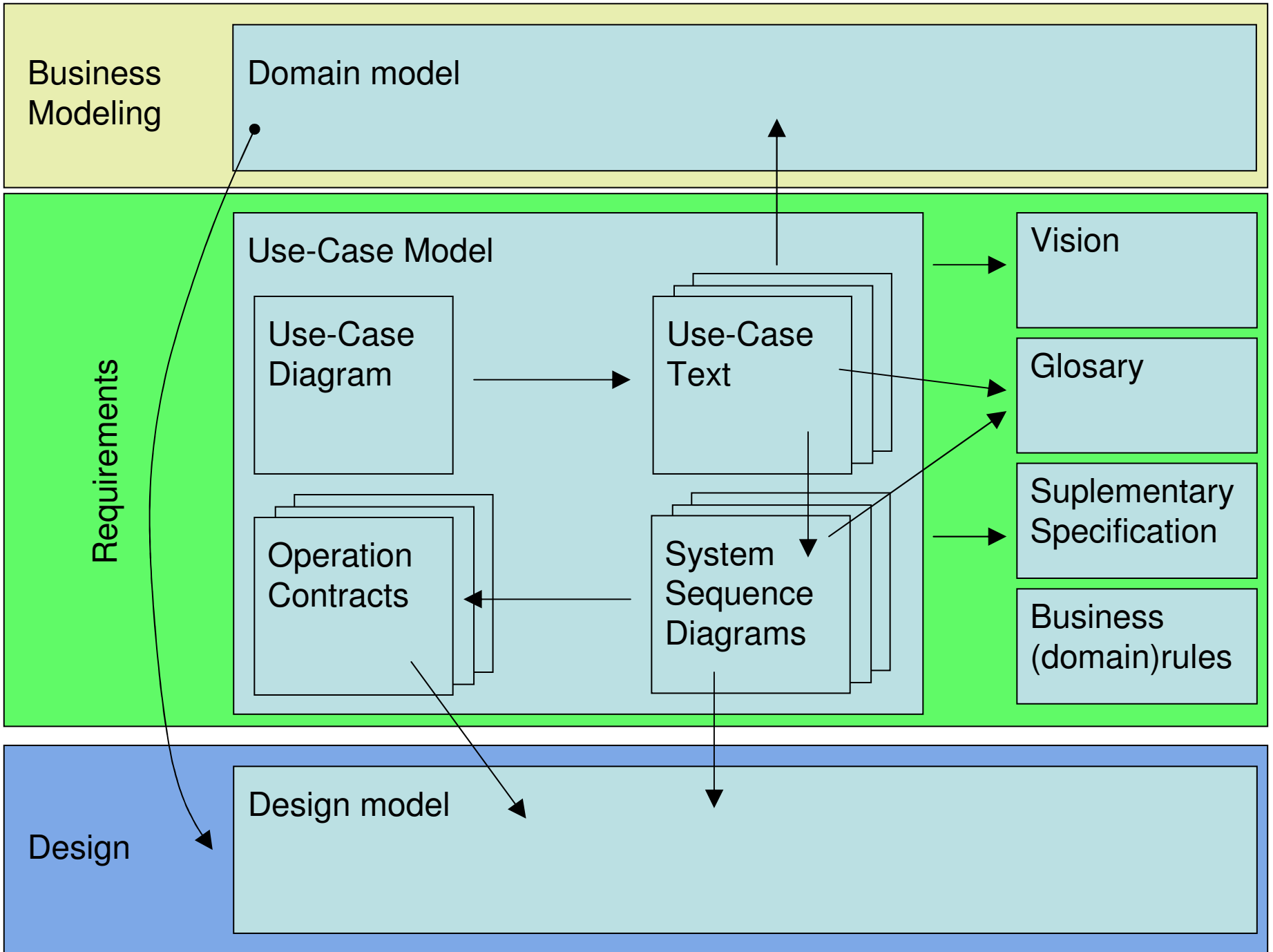
Doménový model



Návrhový model



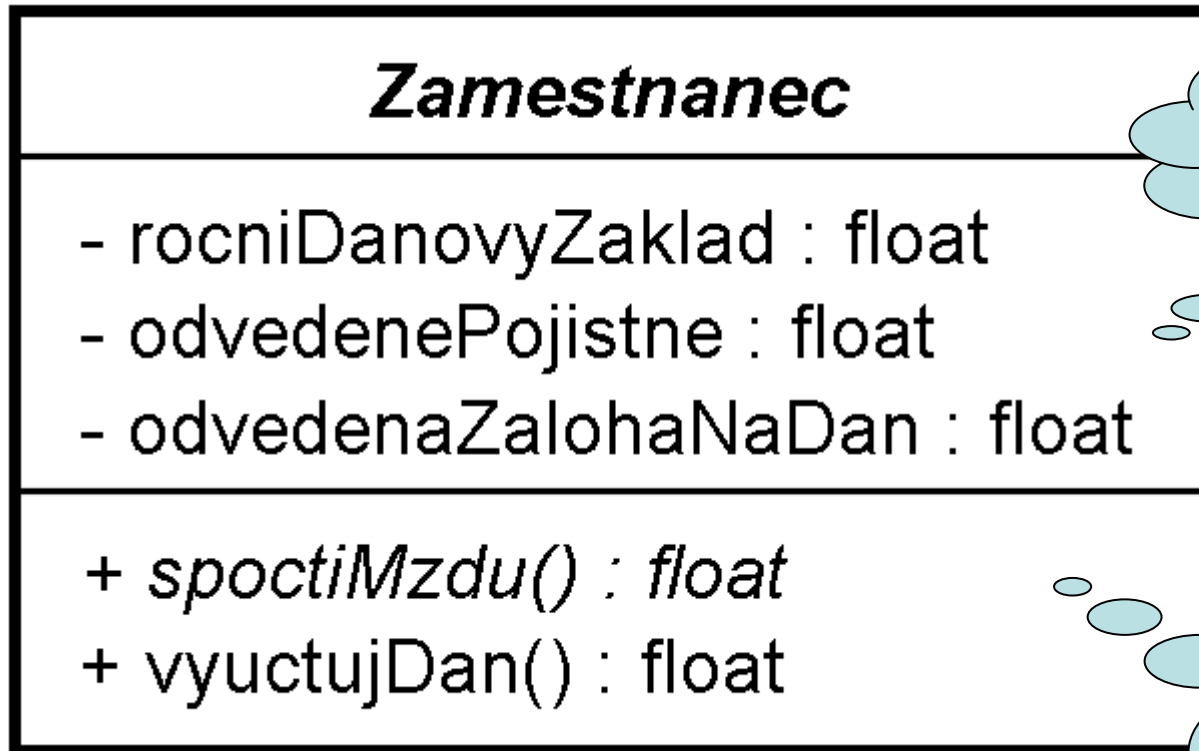




Discipline	Artifact	Incept	Elab	Const	Trans
		I1	E1..En	C1..Cn	T1..T2
Business Modeling	Domain Model		s	r	
Requirements	Use-Case Model	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model		s	r	
	SW Architecture Document		s		
	Data Model		s	r	

Class Diagrams

Třída



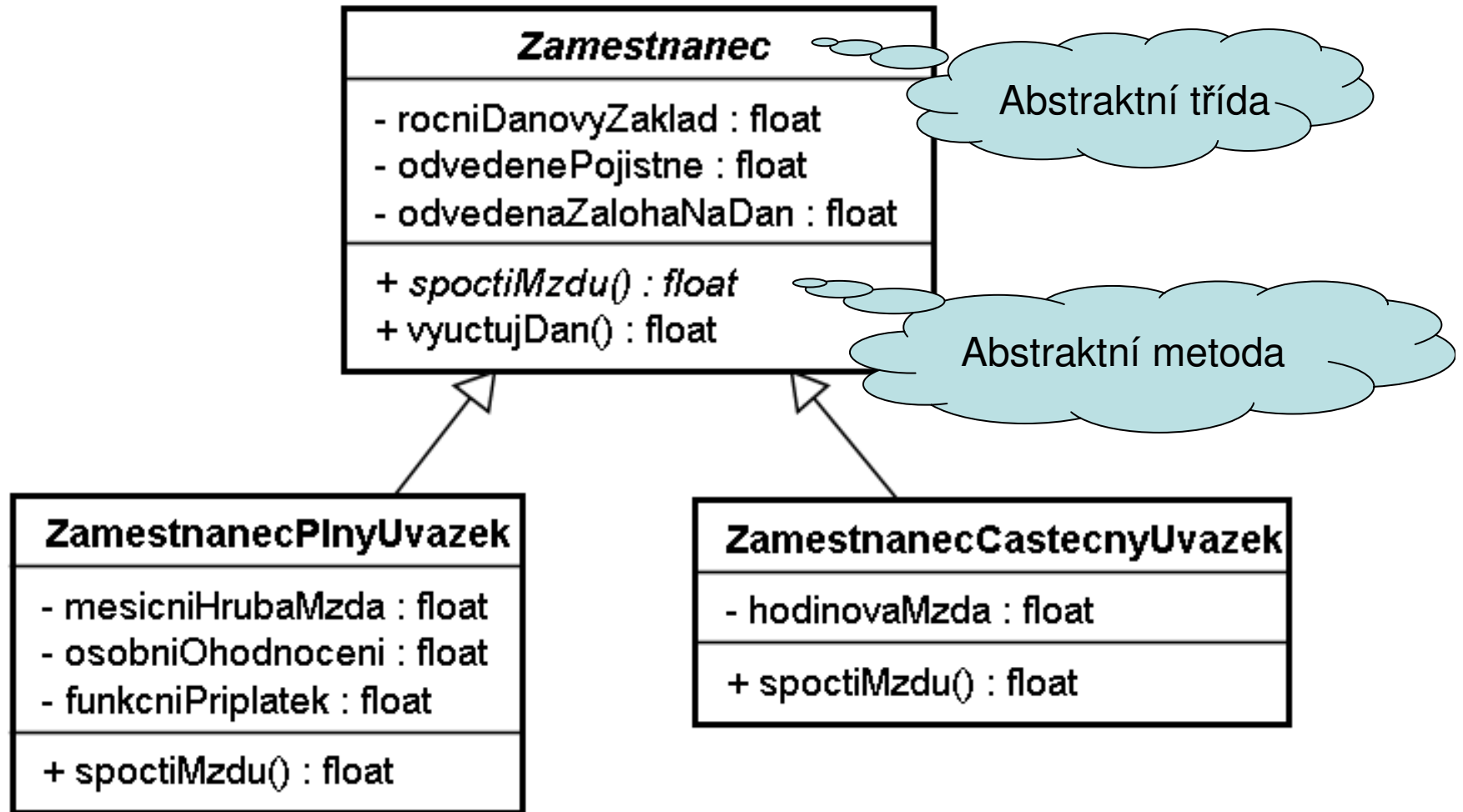
Atributy

Operace

Třída

- Třída je jakýsi prototyp objektů.
- Za třídou si můžeme představit množinu jejích instancí.
- Každý objekt dané třídy má stejnou množinu atributů (proměnných) a operací (metod).
- Tzv. abstraktní třída nedefinuje implementaci (algoritmus) jedné nebo více metod.
- Je-li třída abstraktní, není možno vytvořit její instanci.

Dědičnost (Inheritance) 1



Dědičnost

- Třída B, která dědí od třídy A, má všechny atributy a operace jako třída A (a možná nějaké navíc).
- Třída B je specializací třídy A.
- Třída A je zobecněním třídy B.
- Pokud třída A definuje implementaci nějaké operace (metody) a, pak tato implementace operace a bude použita i pro třídu B, ledaže by třída B tuto operaci předefinovala (overriding).

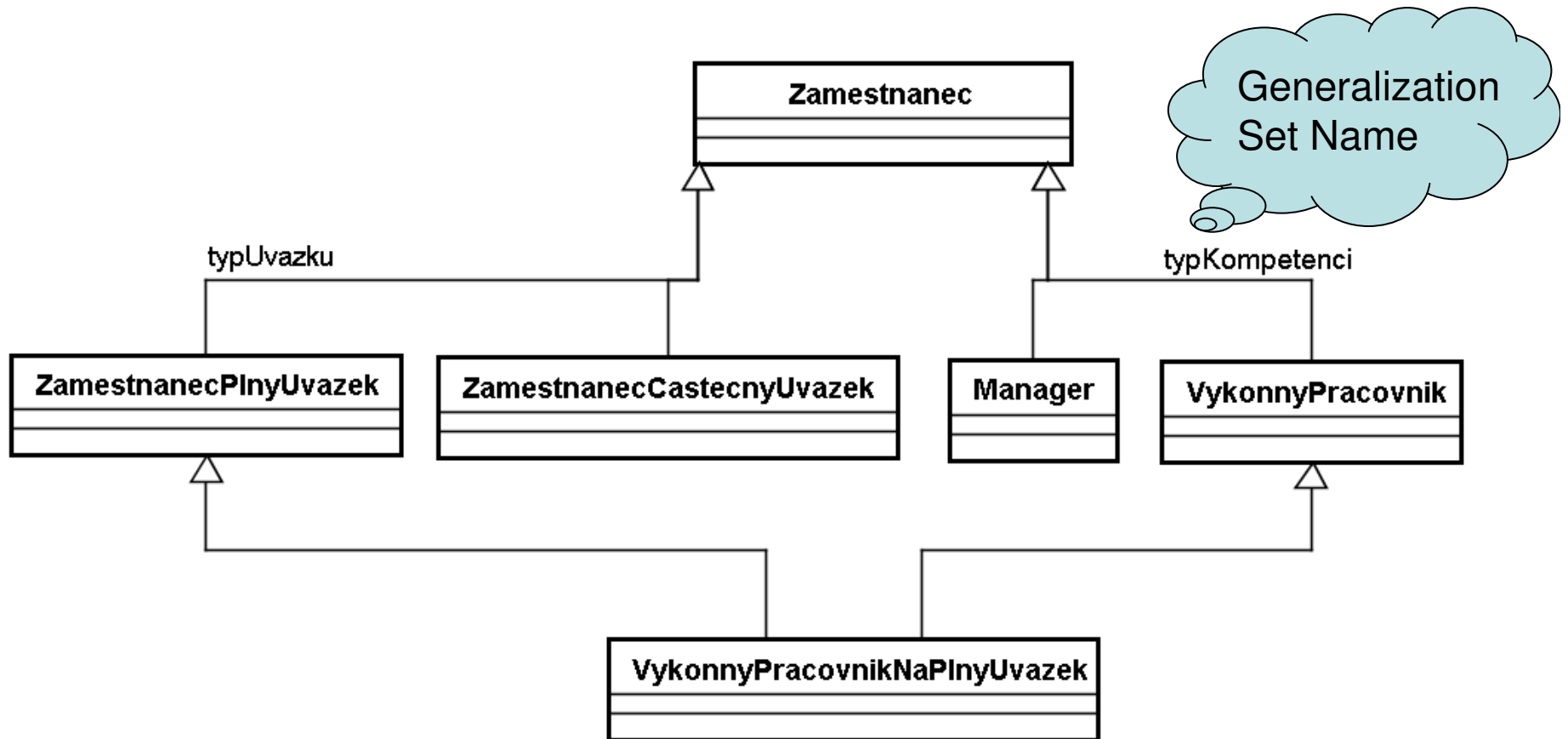
Polymorfismus

- Dědí-li třída B od třídy A, pak všechny instance třídy B mohou vystupovat rovněž jako instance třídy A.
- Instance třídy B jsou tedy „mnohotvaré“ (polymorfní), mají tvar definovaný třídou B i třídou A.
- Dědičnost je transitivní. Tedy, dědí-li C od B a B od A, dědí i C od A.
- V roli instanci třídy A může vystupovat každá instance tříd B nebo C

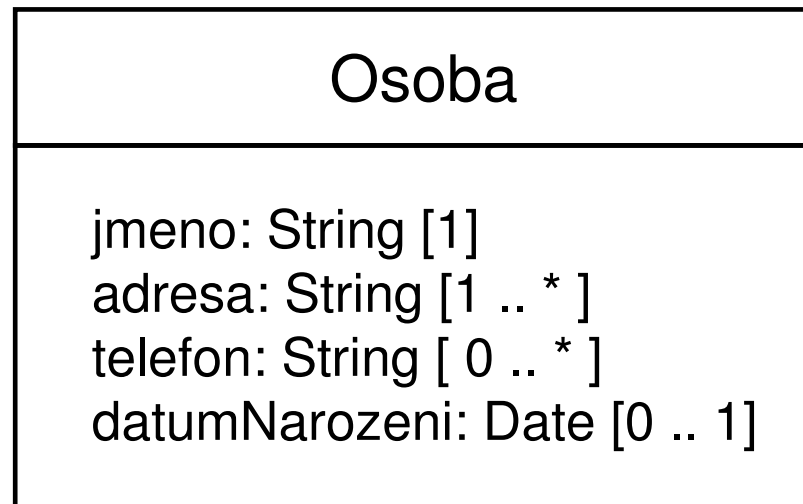
Atributy-operace třídy/instance

- Atributy/operace třídy mohou být použity, aniž by existovala jediná instance této třídy. Jsou to vlastnosti této třídy jako celku, nikoliv jednotlivých instancí.
- Atributy instance jsou vlastnostmi této instance a mohou mít instance od instance rozdílné hodnoty.
- Operace instance jsou prováděny pro danou instanci a proto tato instance musí v okamžiku vyvolání operace existovat.

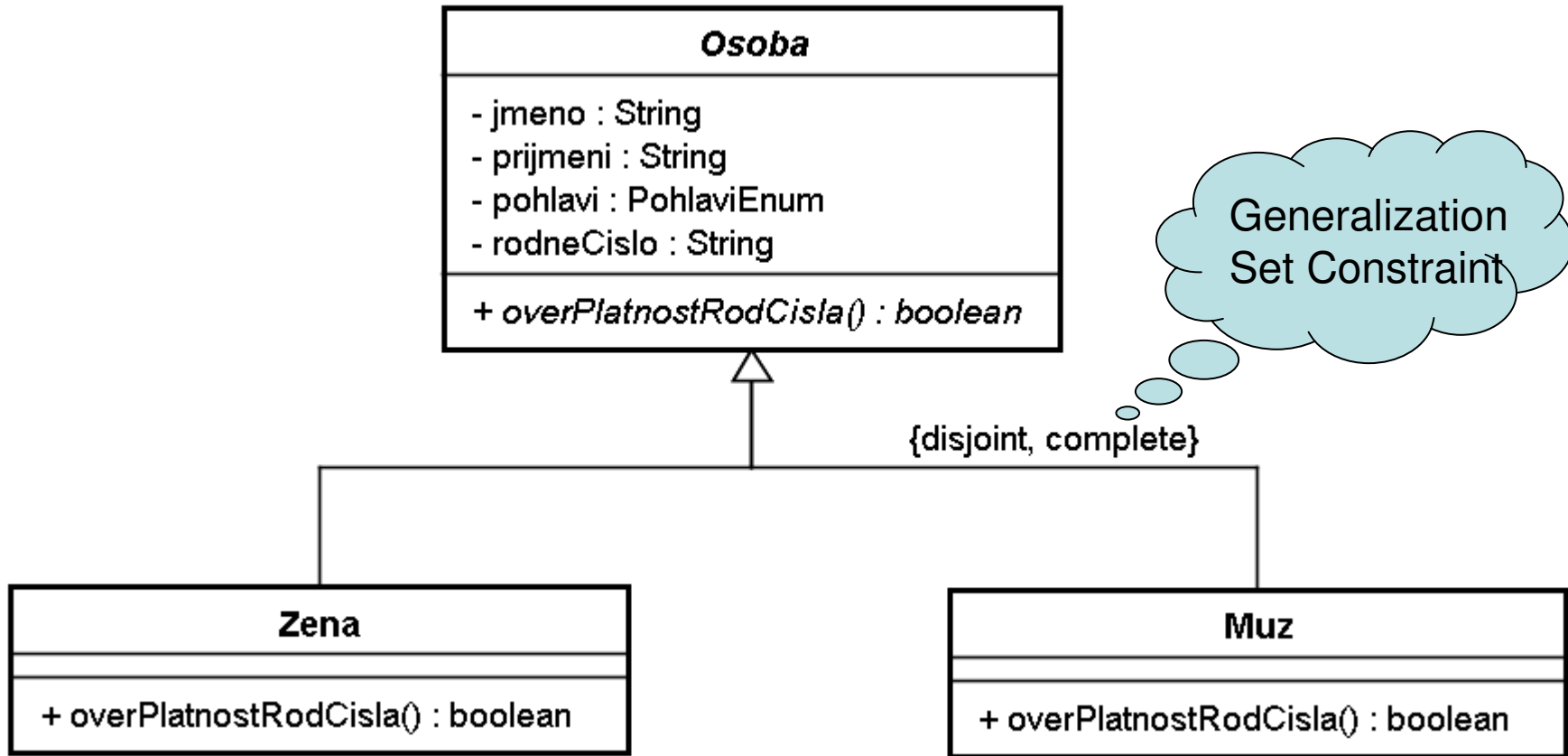
Dědičnost (Inheritance) 2



Multiplicita atributů

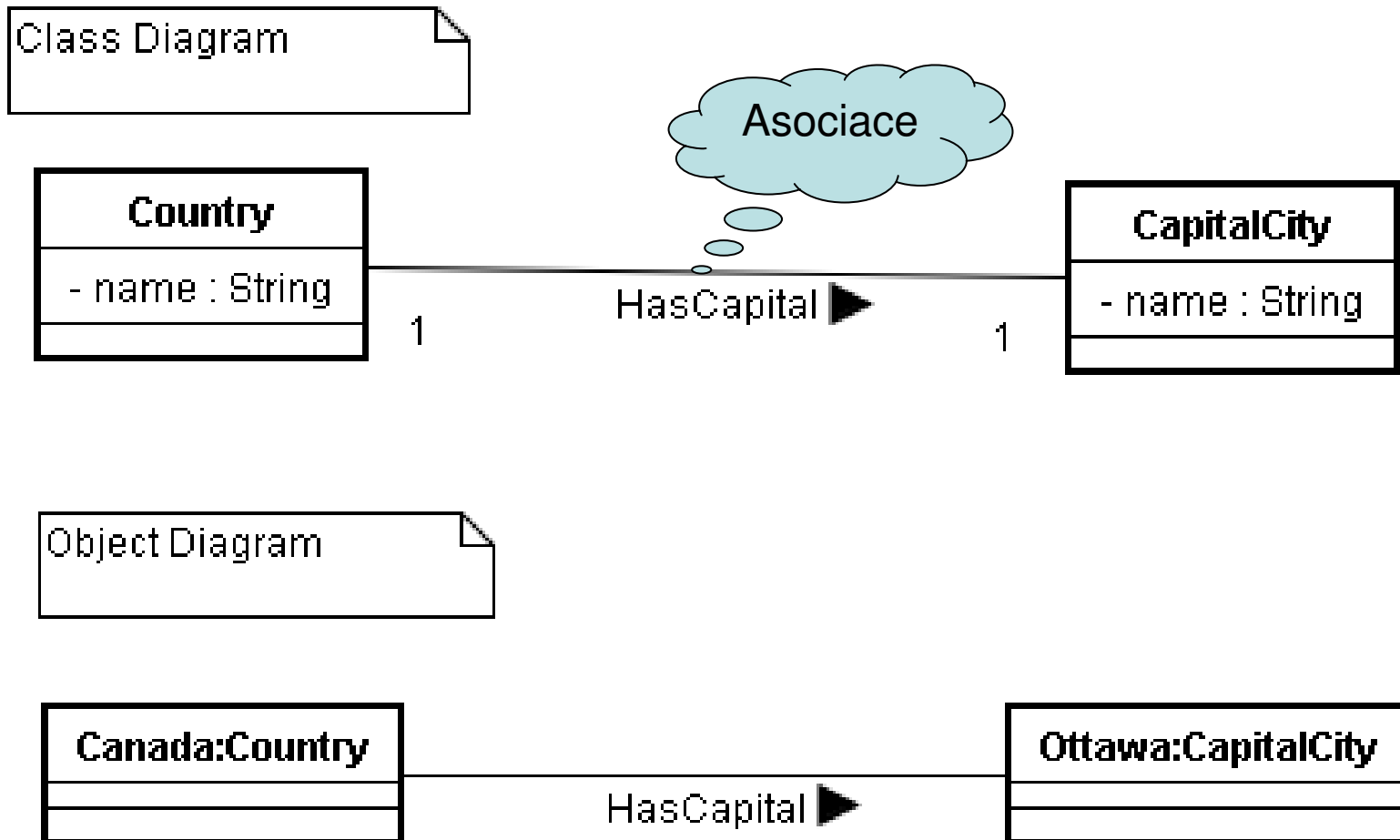


Dědičnost (Inheritance) 3

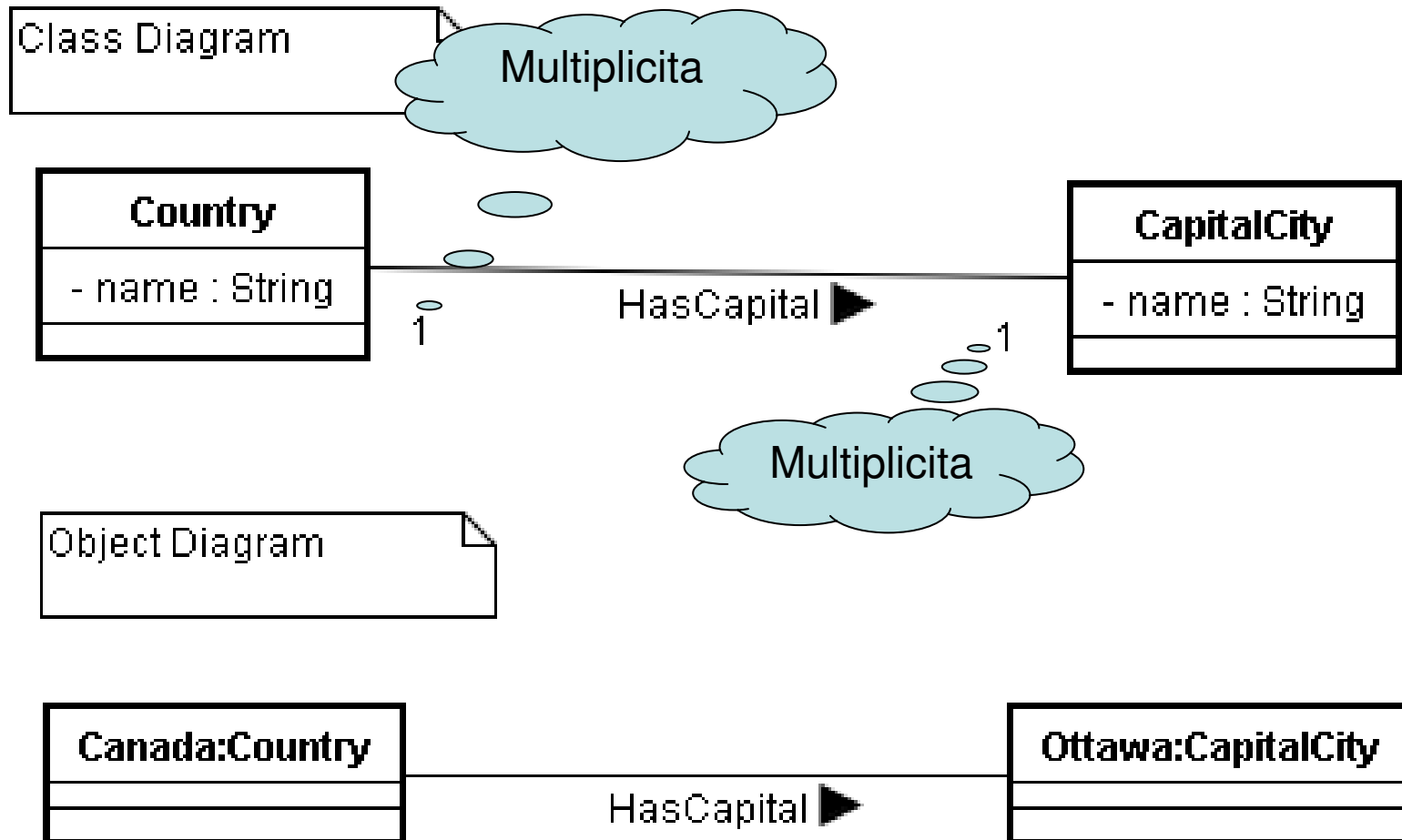


disjoint/overlapping
complete/incomplete

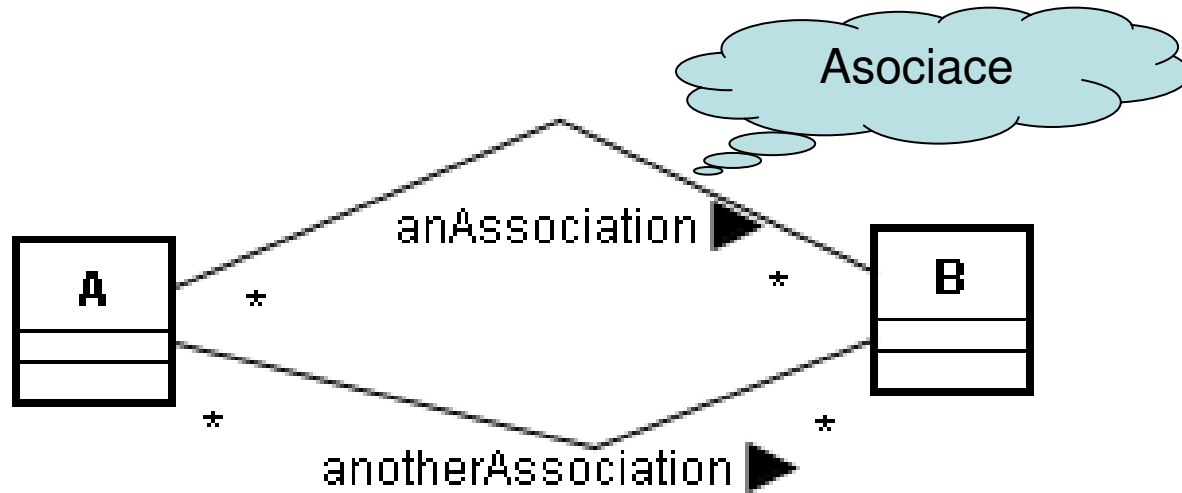
Asociace 1



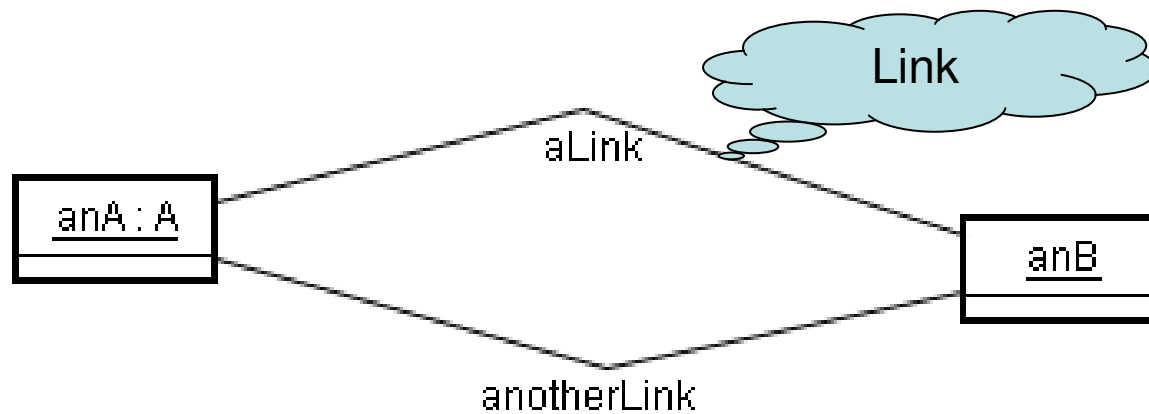
Asociace 2



Asociace 3

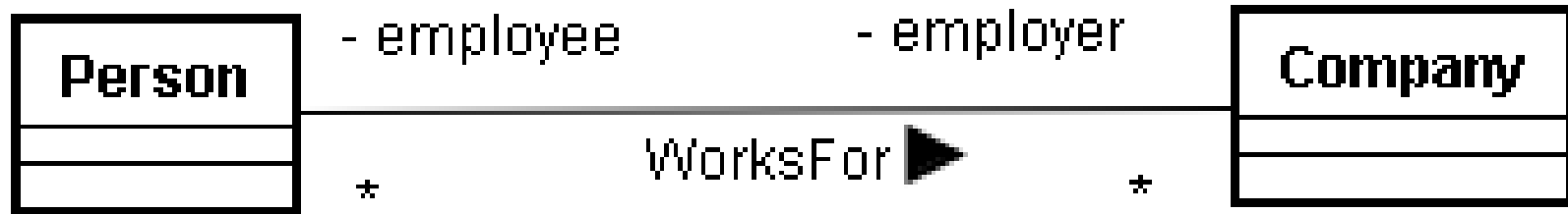


Class Diagram

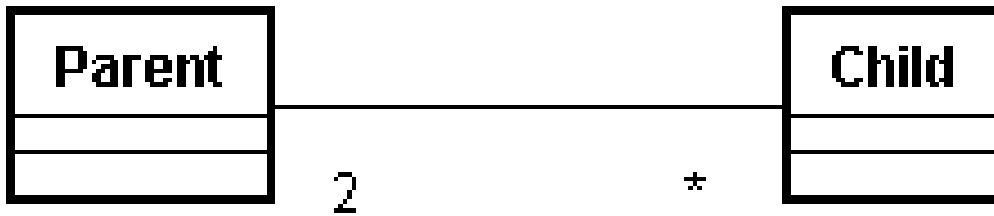


Object Diagram

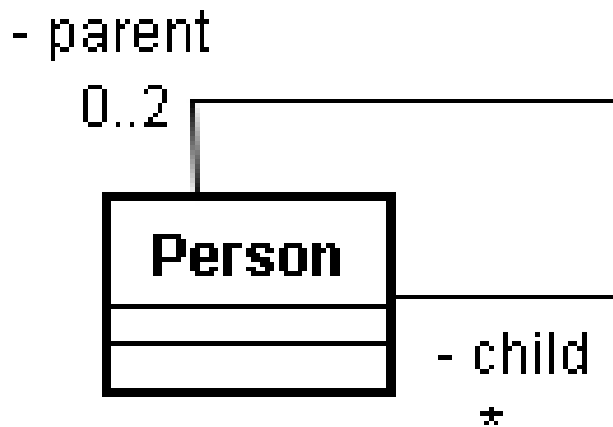
Asociace 4



Asociace 5



Chybný model



Správný model

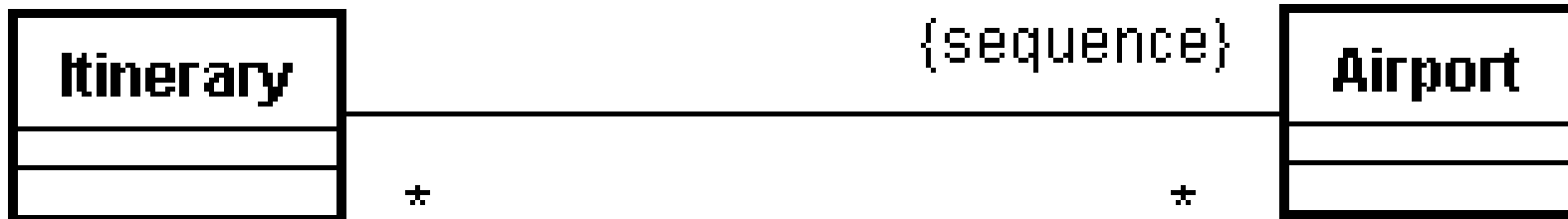
Asociace 6

Uspořádaná kolekce bez opakování

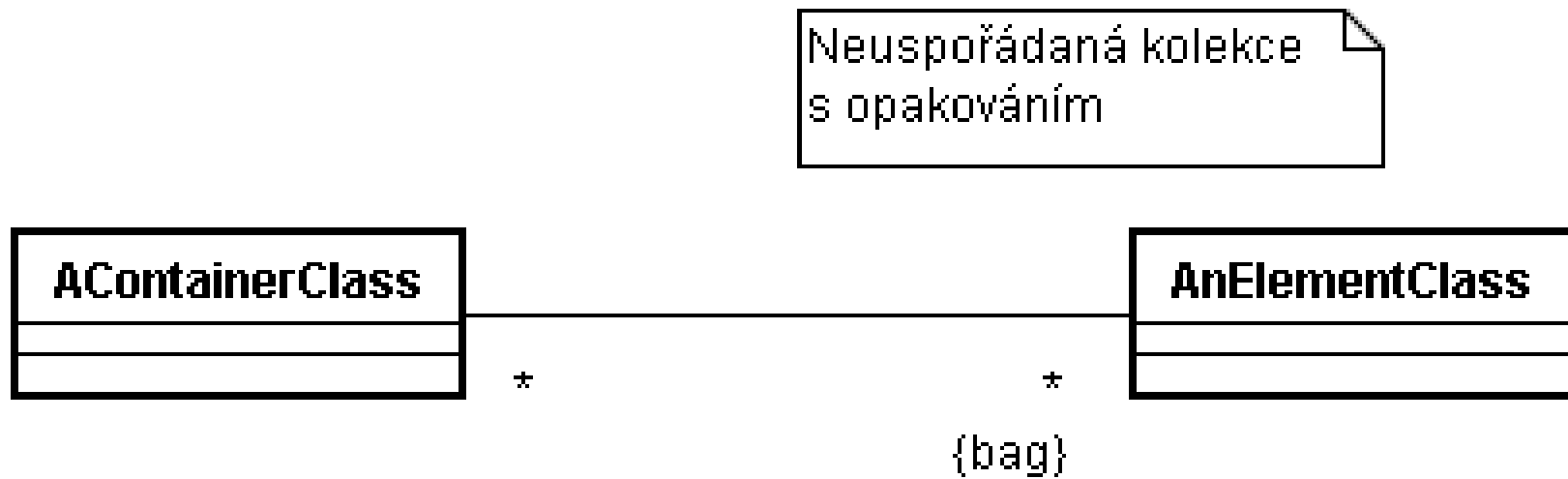


Asociace 7

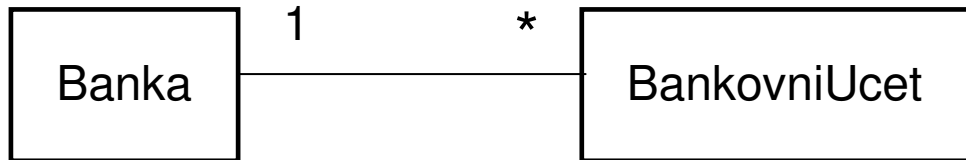
Uspořádaná kolekce
s opakováním



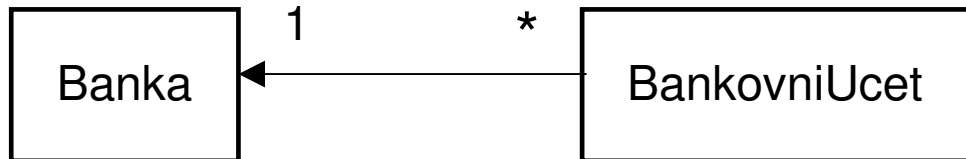
Asociace 8



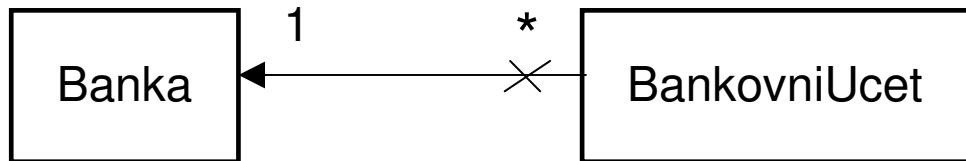
Navigovatelnost



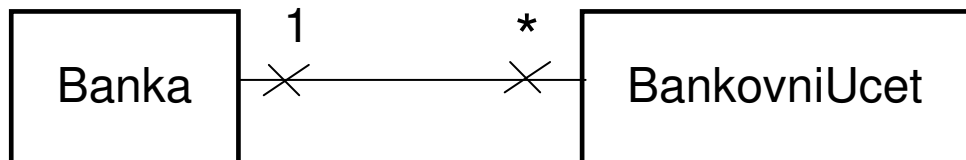
Není specifikována ani v jednom směru



Asociace je navigovatelná v jednom směru, ve druhém není navigovatelnost specifikována

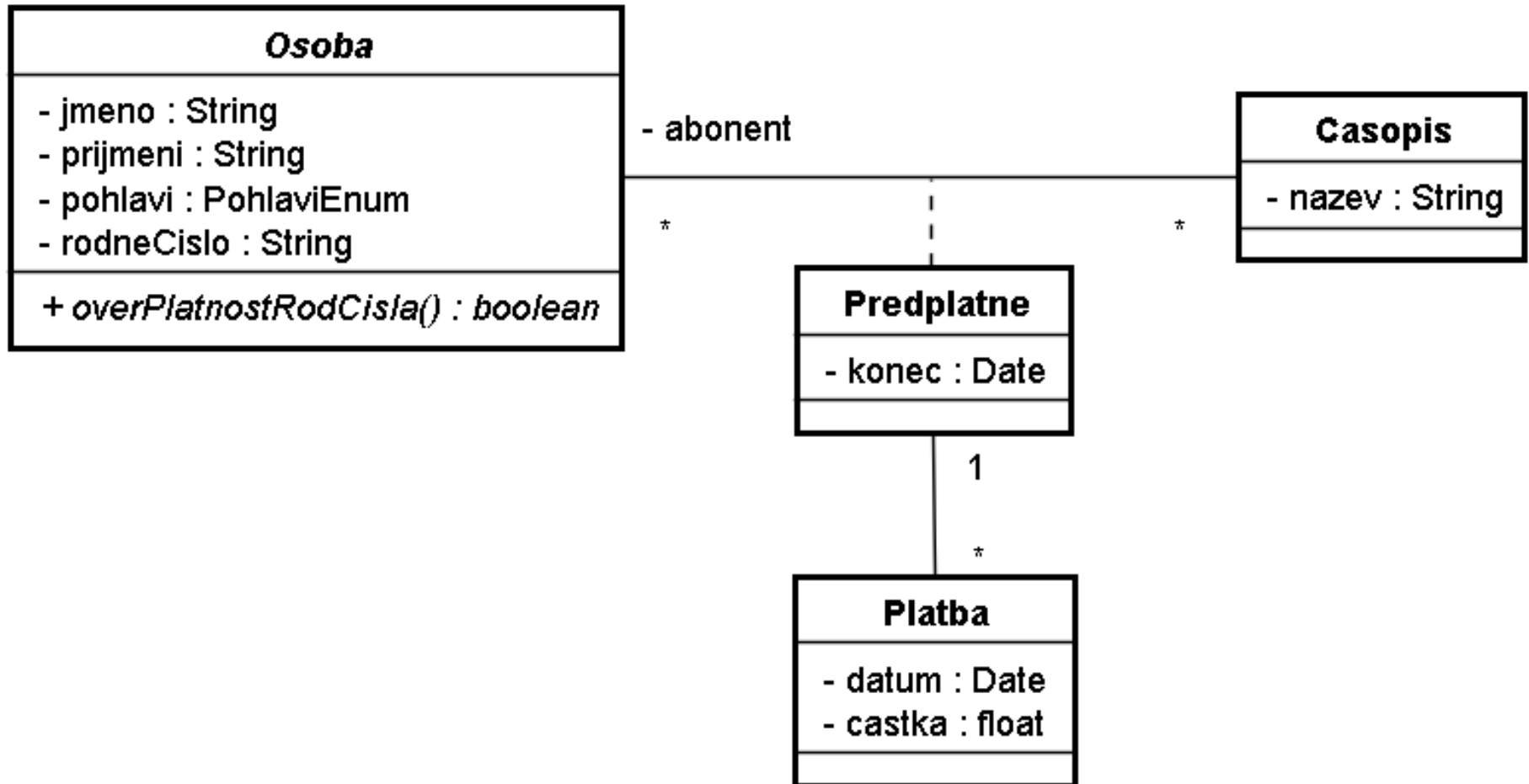


Asociace je navigovatelná v jednom směru, ve druhém navigovatelná není.



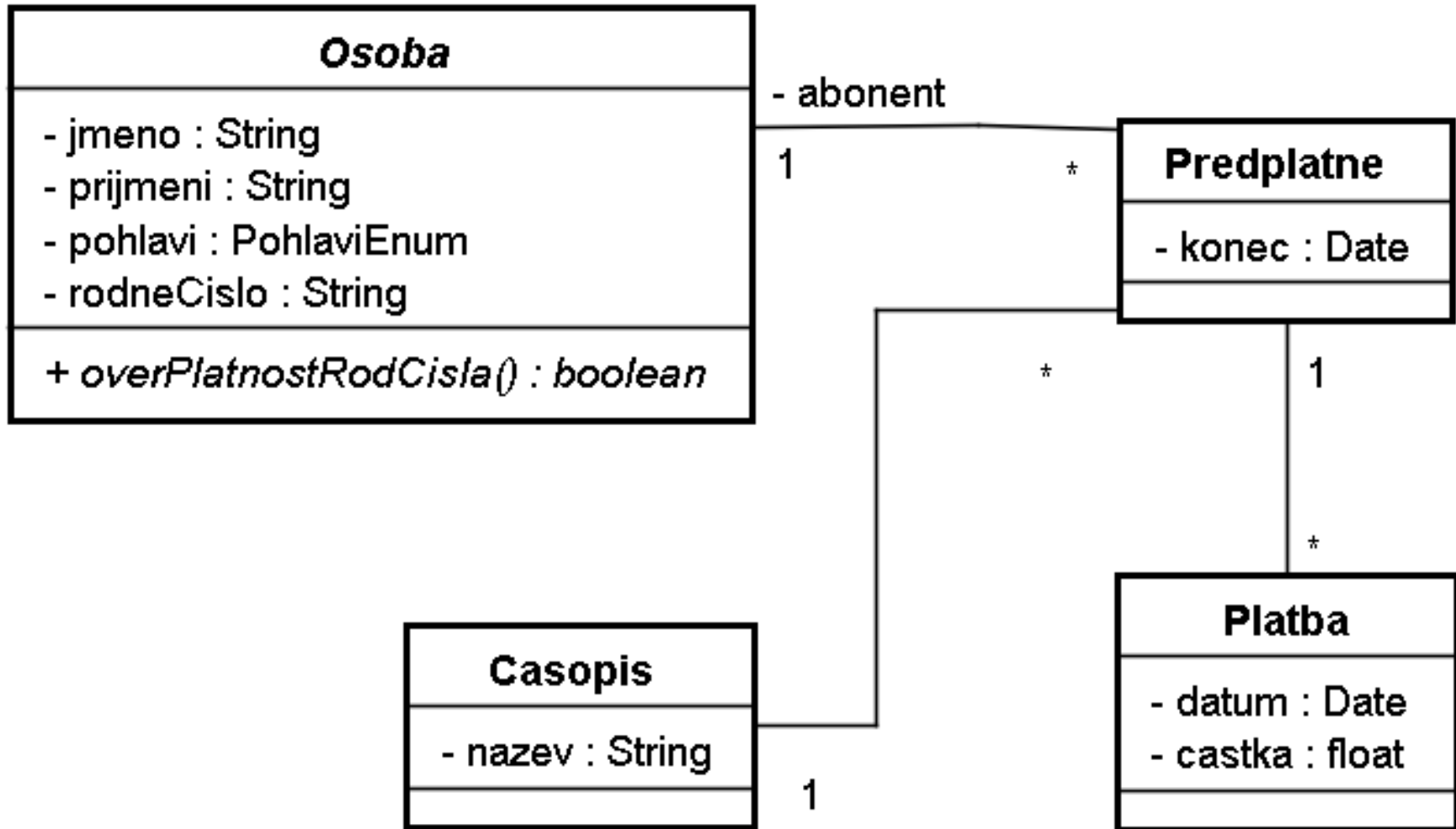
Asociace není navigovatelná v žádném směru.

Asociační třída 1

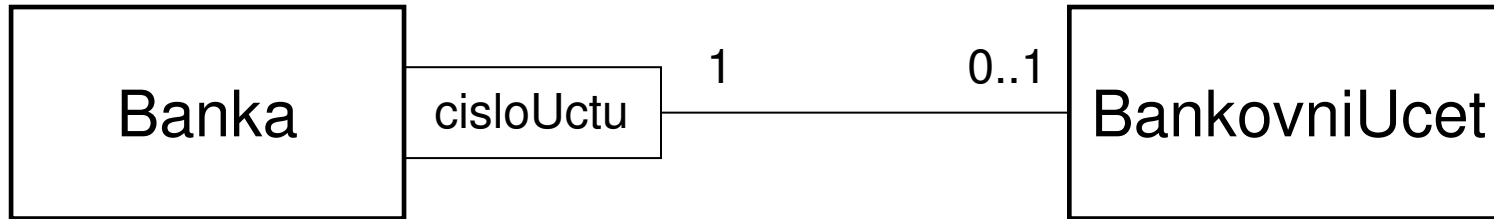


Asociační třída 2

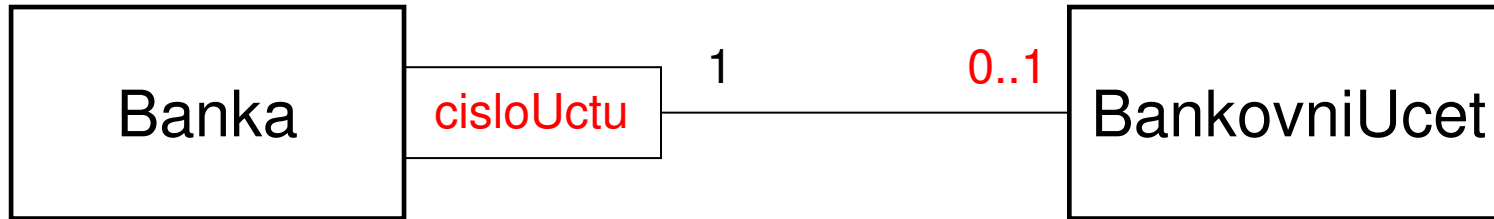
Předchozí model je ekvivalentní tomuto modelu:



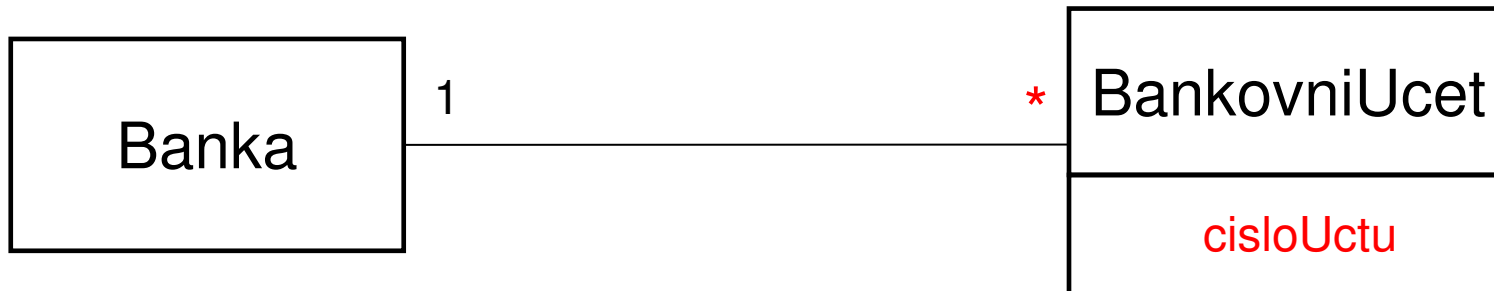
Kvalifikovaná asociace 1



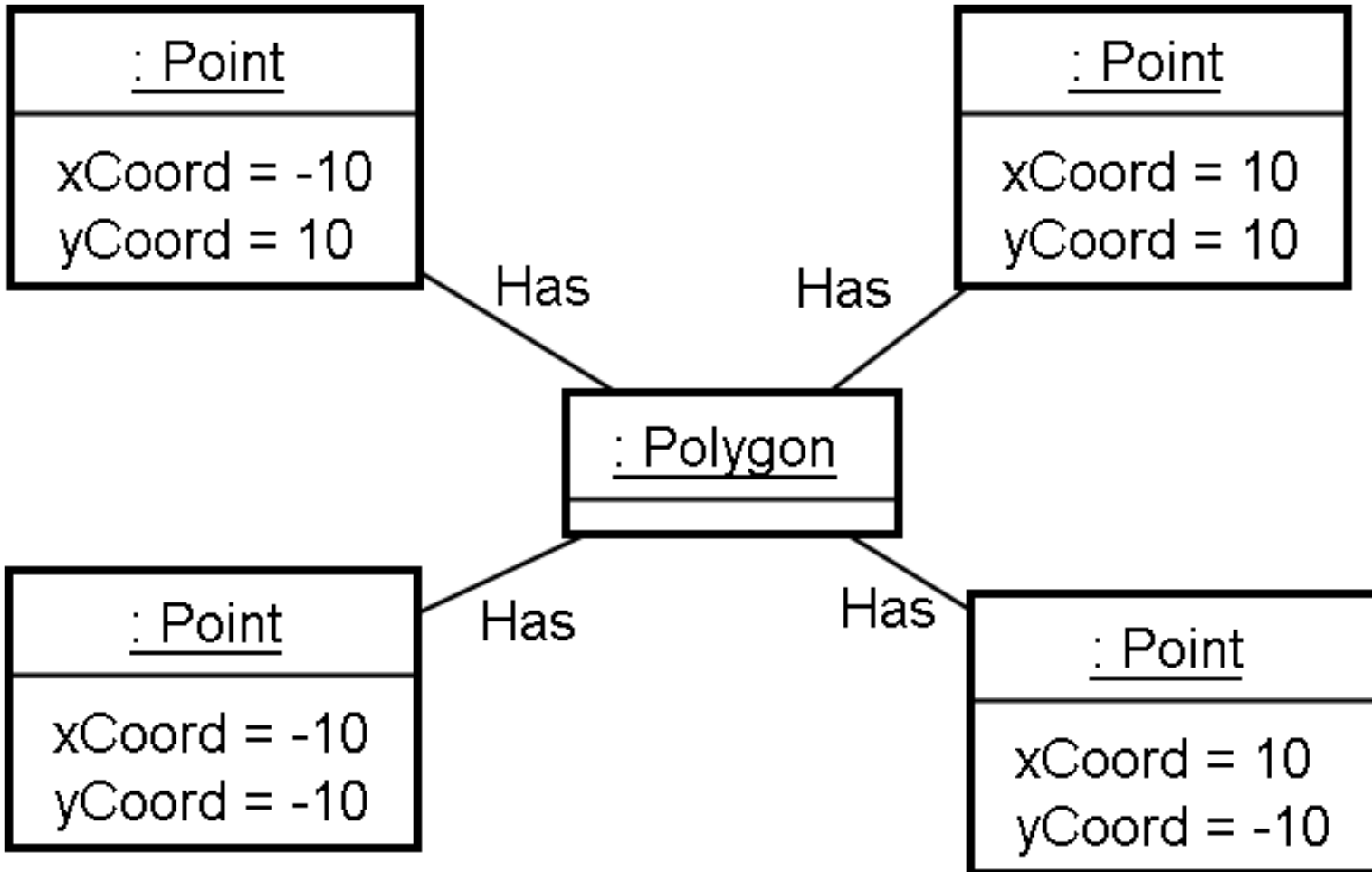
Kvalifikovaná asociace 2



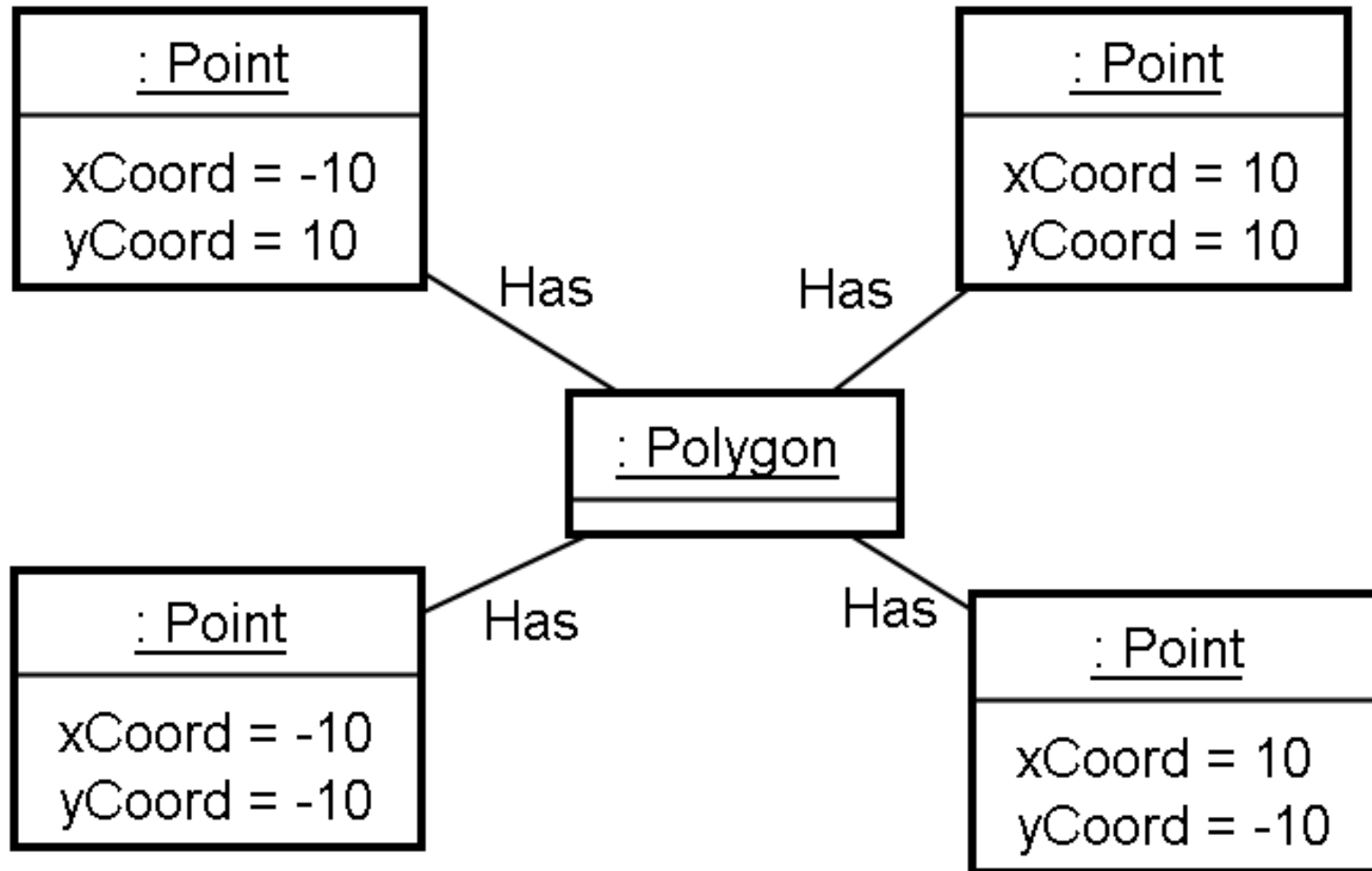
Lze přepsat bez použití kvalifikované asociace takto:



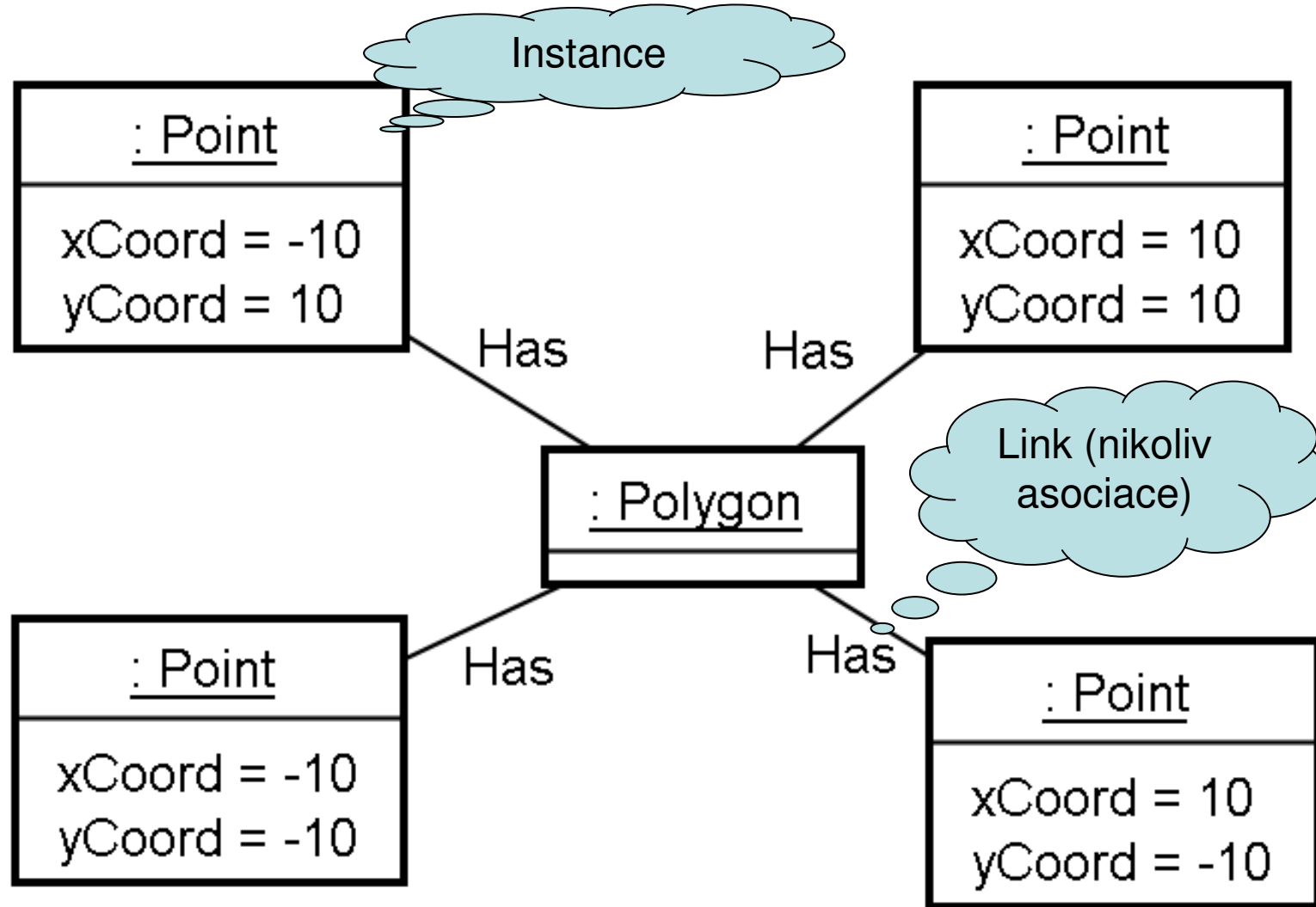
Co je toto? ...



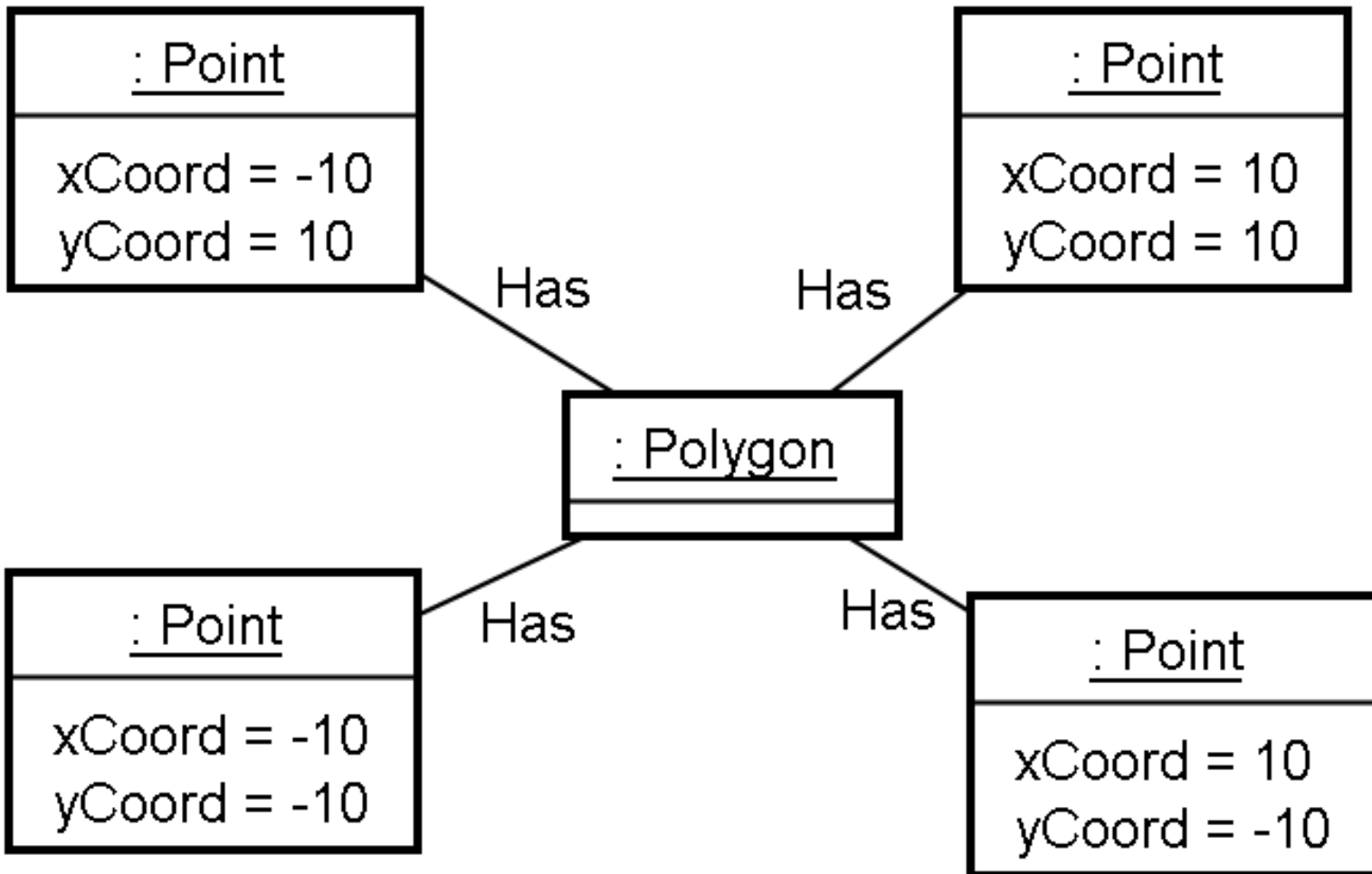
... Objektový diagram reprezentující čtverec



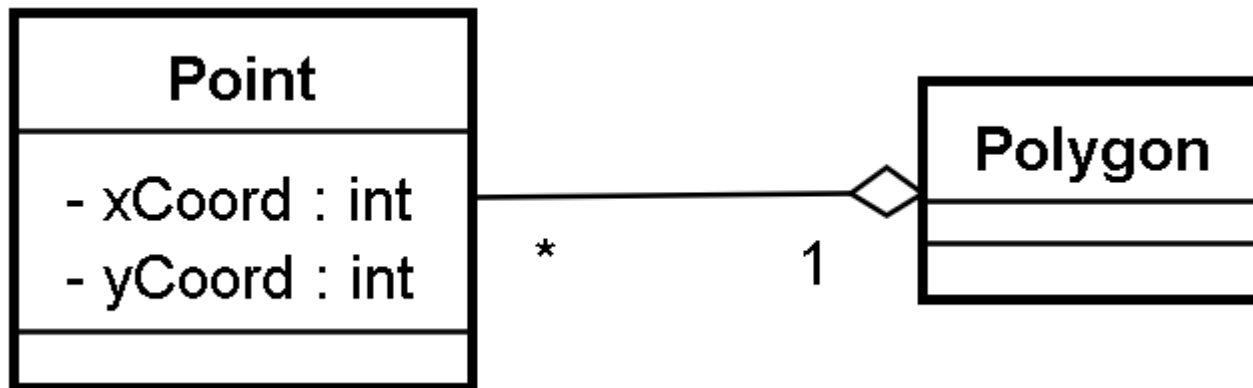
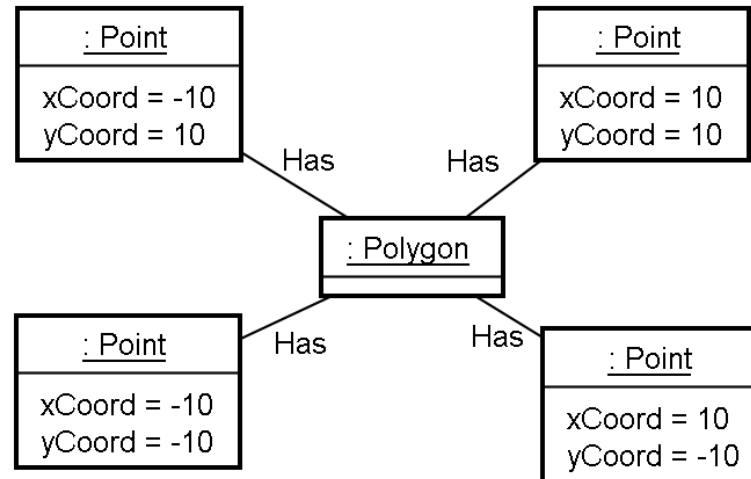
... Objektový diagram reprezentující čtverec



Umíte nakreslit diagram tříd (class diagram) k tomuto diagramu objektů (object diagram)?



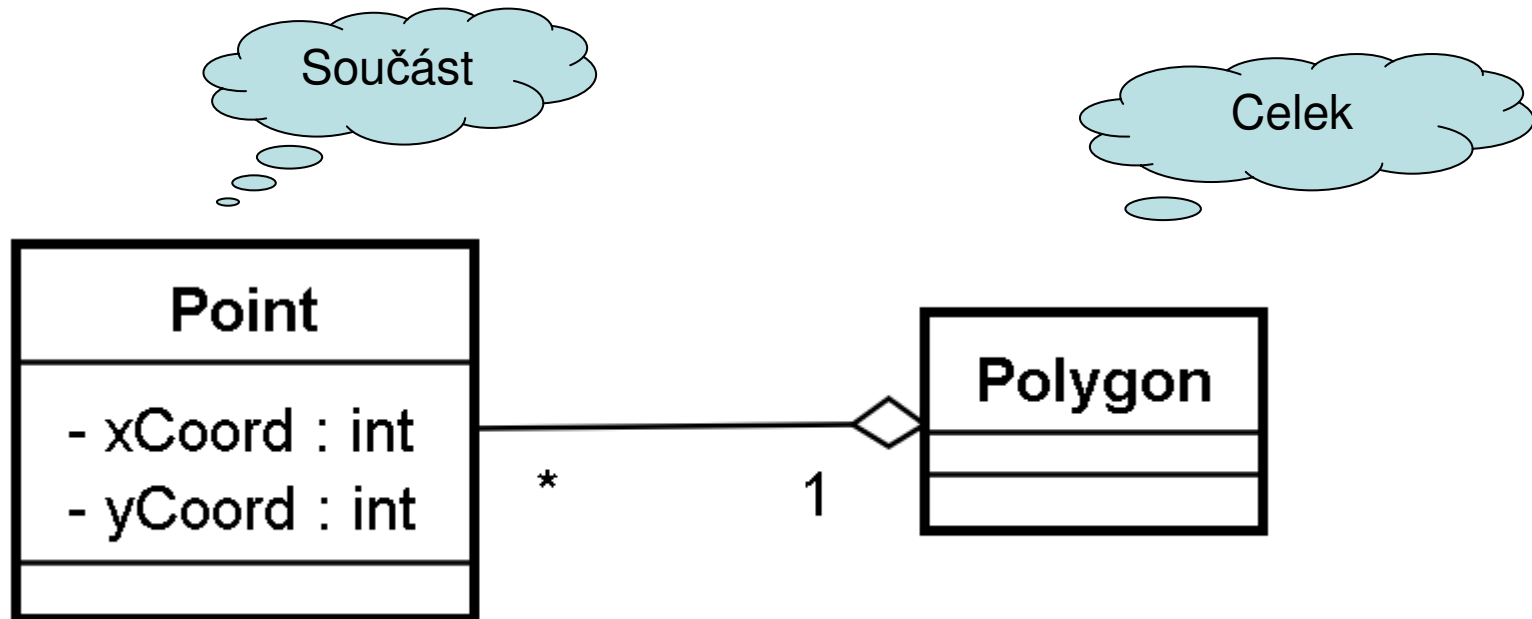
Umíte nakreslit diagram tříd (class diagram) k tomuto diagramu objektů (object diagram)?



Agregace 1

Agregace je zvláštním případem asociace, kdy jedna třída má ve vztahu výsadnější postavení než druhá třída.

Typickým příkladem jsou asociace typu celek-součást.



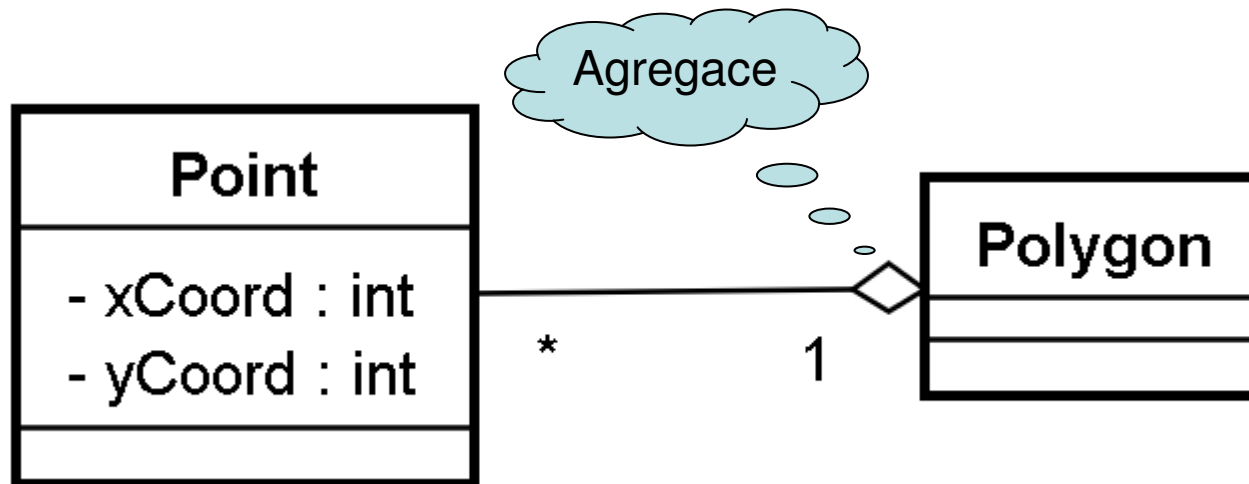
Class diagram na předchozím obrázku by byl teady asi lépe vyjádřen takto:

Agregace 2

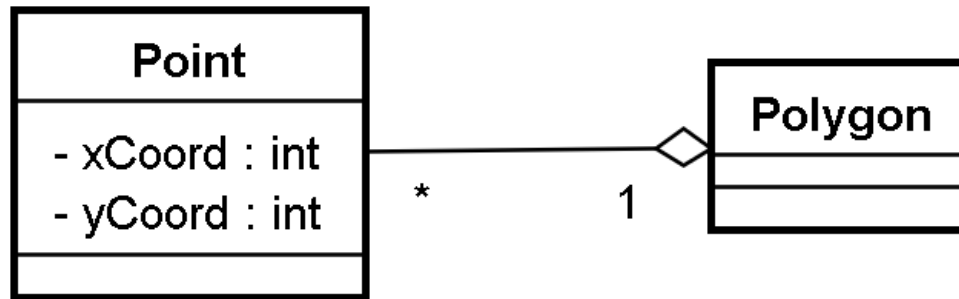
Agregace je zvláštním případem asociace, kdy jedna třída má ve vztahu výsadnější postavení než druhá třída.

Typickým příkladem jsou asociace typu celek-součást.

Class diagram na předchozím obrázku by byl tedy asi lépe vyjádřen takto:



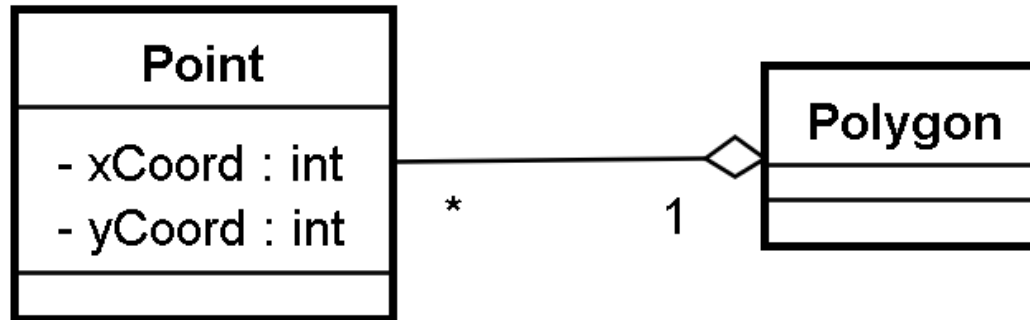
Agregace 3



Agregace je relace:

- **transitivní:** Je-li B součástí celku A a C součástí celku B, pak C je součástí celku A.
- **antisymetrická:** Je-li B součástí A, pak A není součástí B.

Agregace 4

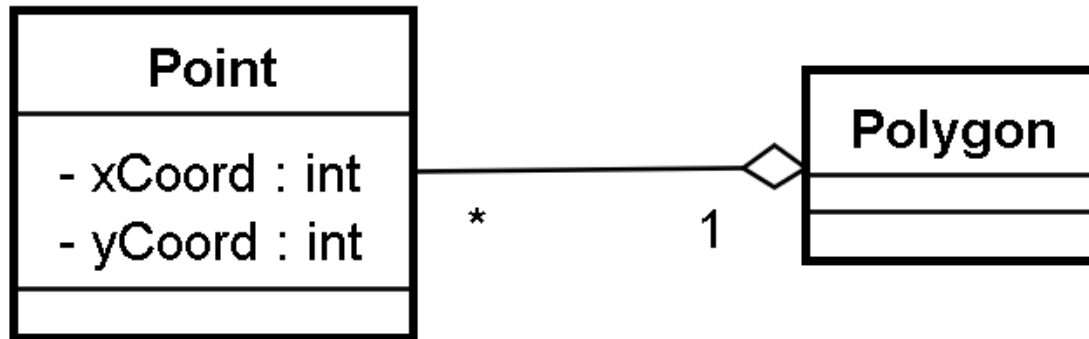


Kdy asociaci modelovat jako agregaci?

Je třeba zodpovědět následující otázky:

1. Vystihuje podstatu asociace fráze „je částí celku“ ?
2. Jsou některé operace celku aplikovatelné na jeho součásti?
3. Propagují se hodnoty některých atributů z celku na všechny nebo aspoň některé části?
4. Je asociace asymetrická (popř. transitivní)?

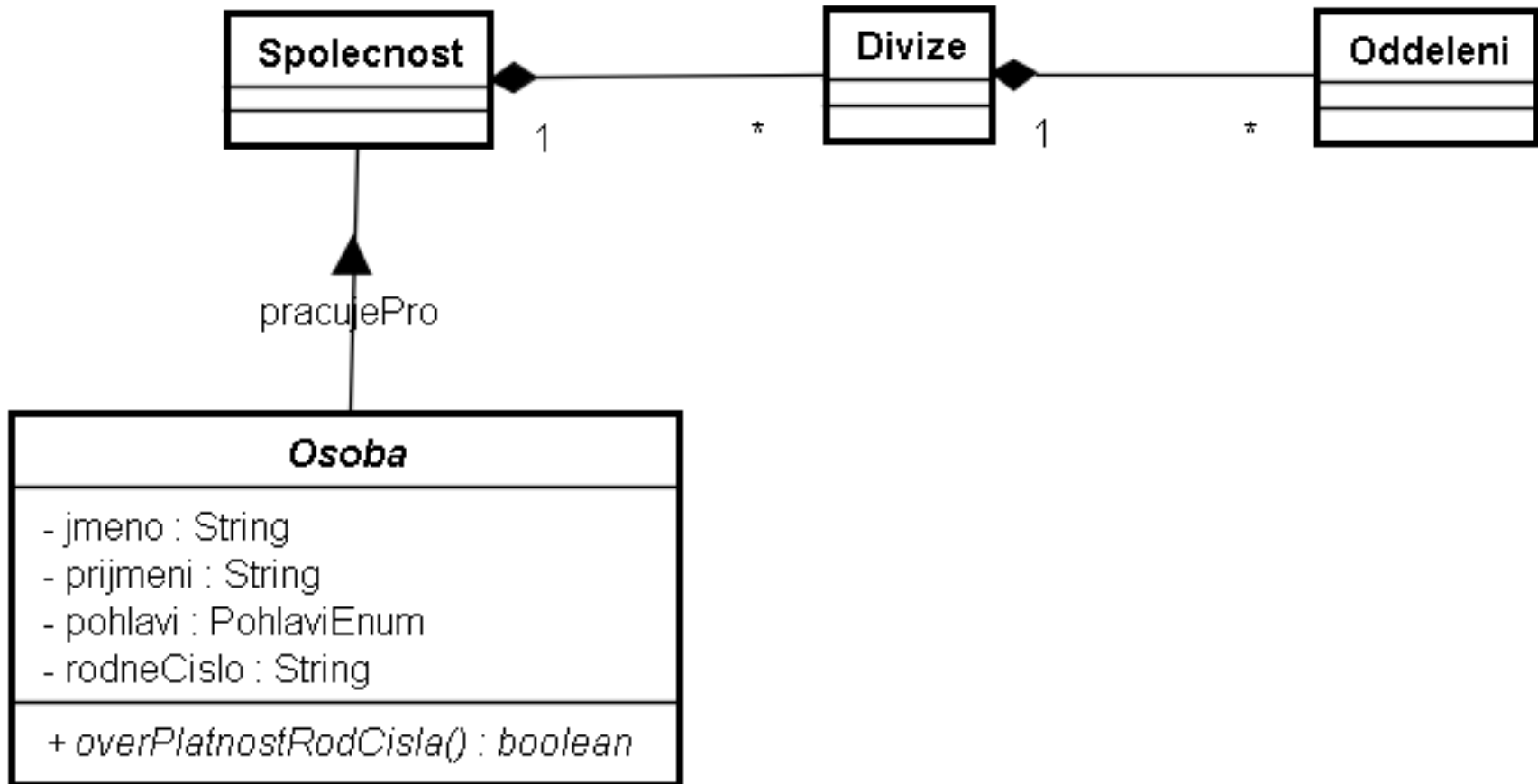
Agregace 5



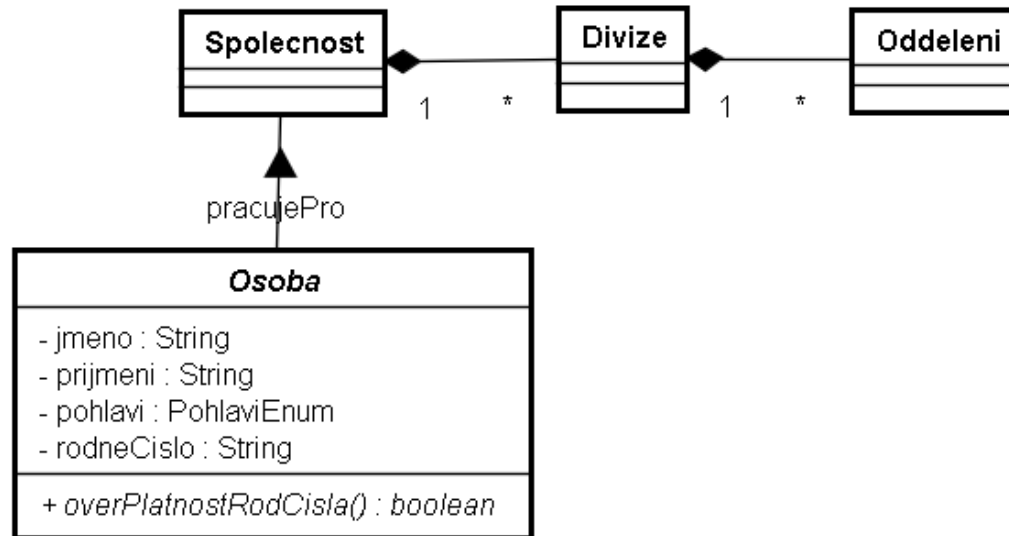
Důležité pravidlo:

Nejste-li si jisti, zda je daná asociace agregací, modelujte ji raději pouze jako prostou asociaci (nic zásadního se nepokazí).

Kompozice 1



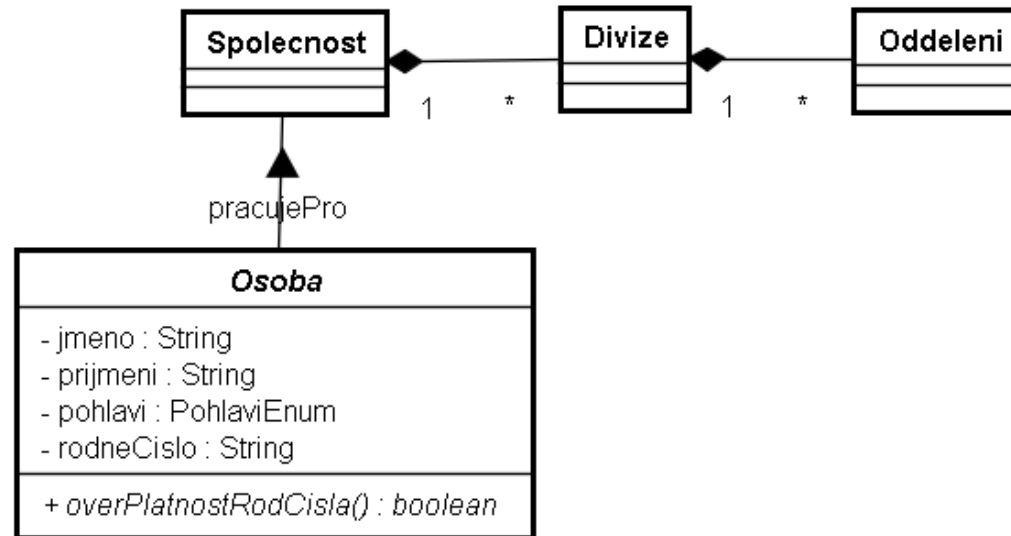
Kompozice 2



Kompozice je forma **agregace** s dvěma dodatečnými omezeními:

1. Součást může patřit maximálně do jednoho celku (oddělení nemůže být sdíleno více divizemi)
2. Jakmile je součást zařazena do celku, je její životní cyklus určen životním cyklem celku (zanikne-li společnost, zaniknou i všechny její divize).

Kompozice 3



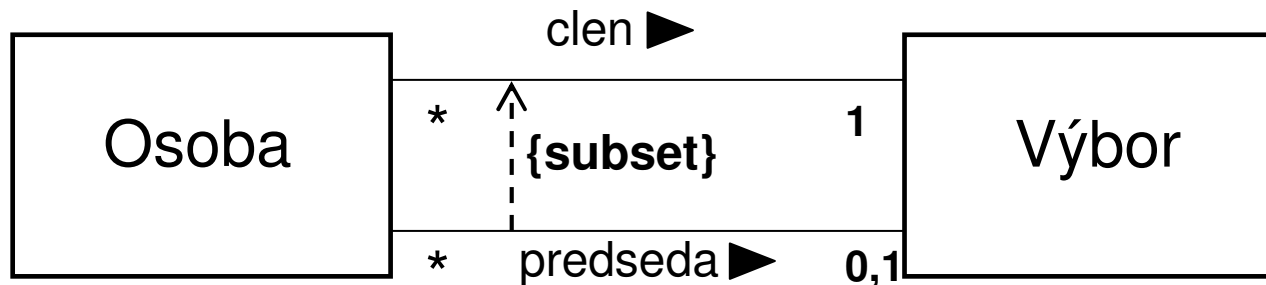
Důležité pravidlo:

Modelujeme-li agregaci jako prostou asociaci, nic podstatného se nestane.

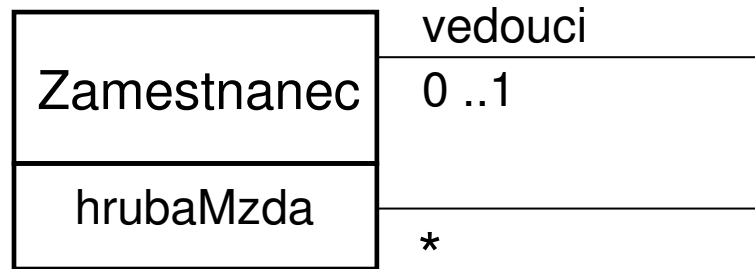
Namodelujeme-li kompozici jako prostou asociaci nebo jako agregaci, ztrácíme důležitou informaci o dvou omezeních – (i) nesdílení součásti mezi celky a (ii) determinace životního cyklu součásti životním cyklem celku.

Definice omezení pro asociace

- Multiplicita (kardinalita množiny)
- Kvalifikovaná asociace
- Uspořádanost instancí na * konci ({ordered}, {sequence}, {bag})
- **Explicitně vyjádřené omezení**



Definice omezení pro objekty 1



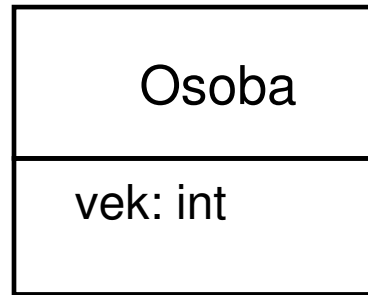
{hrubaMzda < vedouci.hrubaMzda}

Definice omezení pro objekty 2



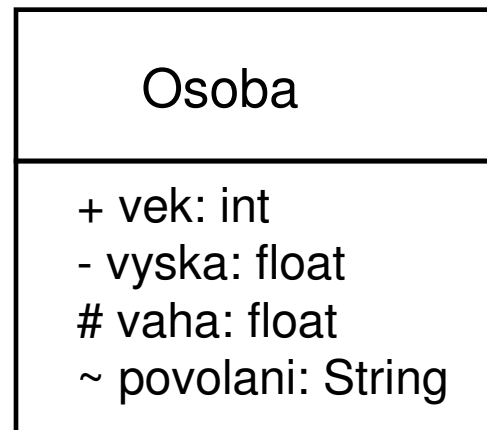
{ $0.8 \leq \text{delka}/\text{sirka} \leq 1.5$ }

Definice omezení pro objekty 3



{vek je neklesajici }

Viditelnost



+ .. public
- .. private
.. protected
~ .. package

Aplikovtelné na:

- atributy
- operace
- asociace

Scope

(vlastnosti instancí nebo tříd)

zpravaNaZaznamníku
<u>maximalniDelkaZpravy</u> <u>maximalniDobaUchovani</u> datumNahrani casNahrani priorita zprava