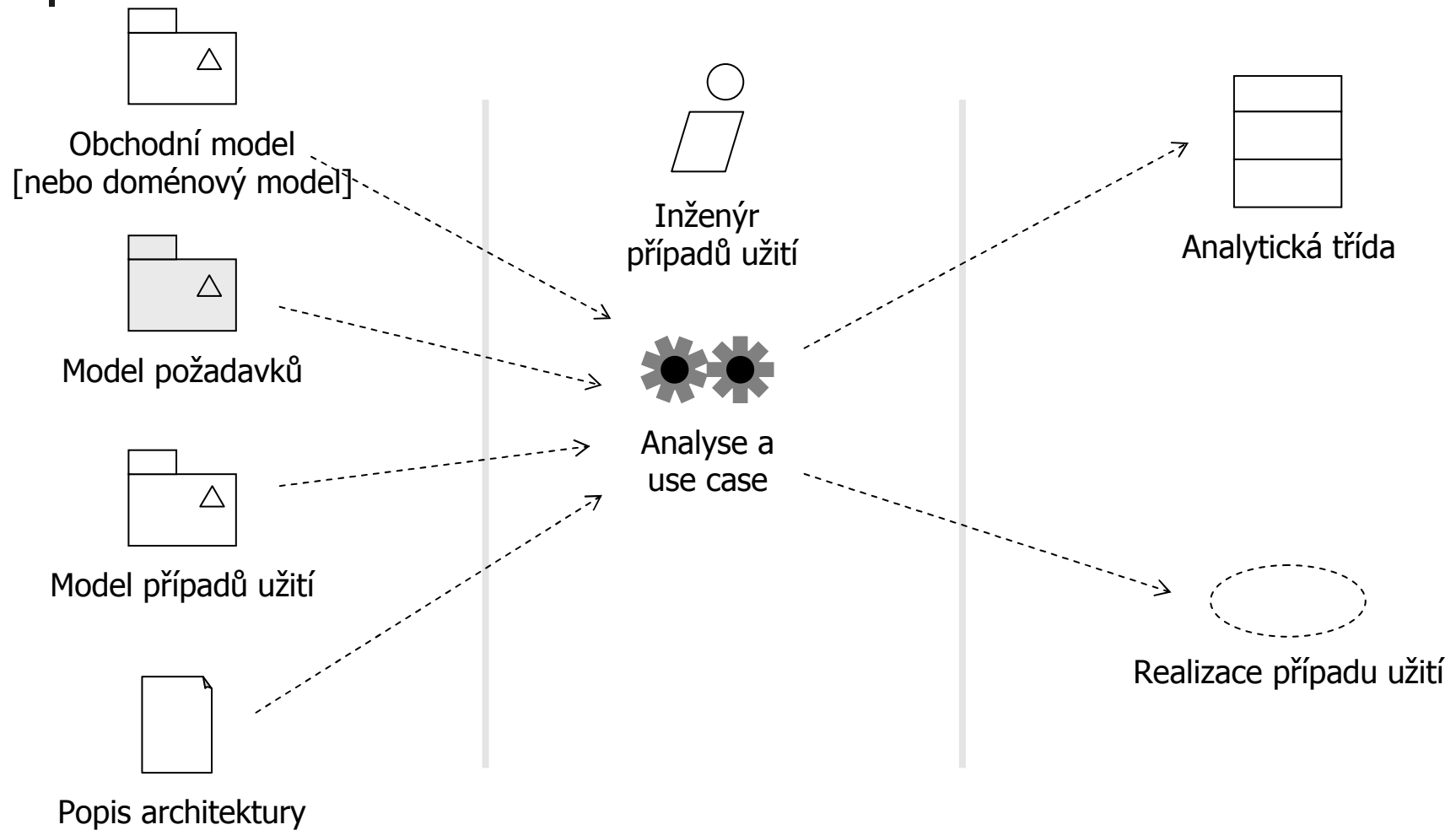


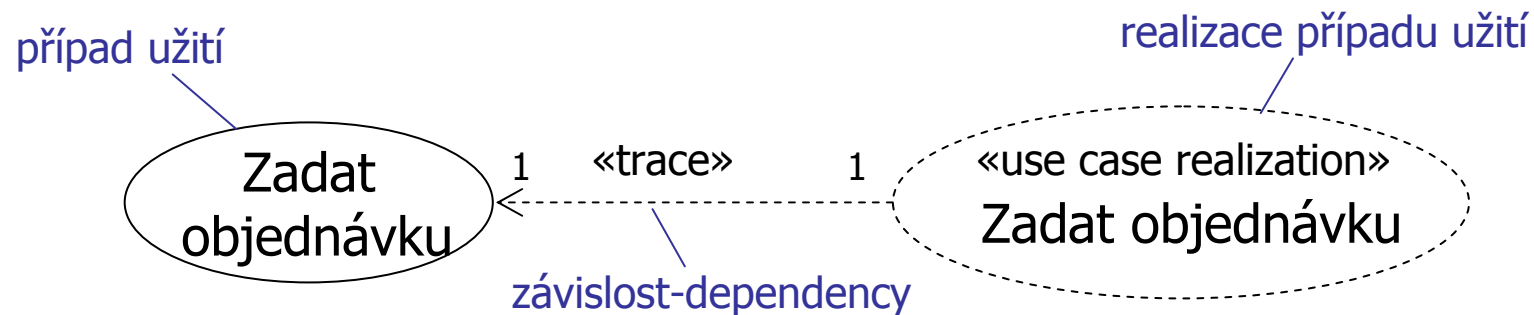
Analýza – Realizace případů užití

Analýza část 9

Analýza případu užití



Co jsou realizace případu užití?

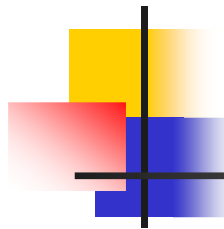


- Každý případ užití má *realizaci případu užití*
 - je to část modelu, která ukazuje jak analytické třídy mezi sebou spolupracují při realizaci chování specifikované v případě užití
 - je to model, *jak* je případ užití realizován nalezenými analytickými třídami
- Nejsou obvykle přímo modelovány
 - nejsou zachyceny žádné další informace, lze použít přímo případ užití
 - můžeme zachytit stereotypem spolupráce



Realizace případu užití - prvky

- Realizace případu užití se skládá z těchto prvků:
 - Diagramy analytických tříd
 - Ukazují interakci analytických tříd při realizaci případů užití
 - Diagramy interakce
 - Ukazují spolupráci a interakci specifických instancí uskutečňujících případ užití – jsou to „snímky“ spouštěného systému
 - Speciální požadavky
 - Realizace případu užití může odhalit nové požadavky, které se vztahují k případu užití, které je třeba zachytit
 - Upřesnění případů užití
 - Během realizace mohou být odhaleny nové informace, které mohou upřesnit původní případ užití



Interakce

- Interakce jsou jednotkami chování kontextového klasifikátoru
- V realizaci případů užití se použije kontextový klasifikátor
 - Interakce ukazují jaké je chování instancí analytických tříd, které je specifikováno případem užití.
 - Během práce na diagramech interakce se odhalují další operace a atributy
- Diagramy interakce zachycují:
 - Čáry života (Lifelines) – účastníci interakce
 - Zprávy (Messages) – komunikace mezi čárami života

Čáry života

jimůvÚčet [id = "1234"] : Účet

název

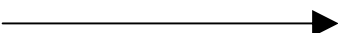
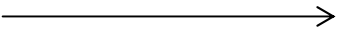
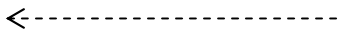
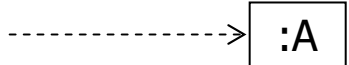
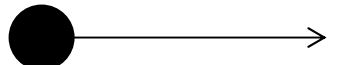
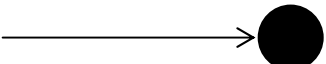
selektor

typ

- Čára života reprezentuje jednoho účastníka v interakce
 - Ukazuje jak klasifikovaná instance účastní interakce
- Čára života může mít:
 - název- používá se jako identifikátor čáry života v rámci interakce
 - selektor – logická podmínka pro výběr určité instance
 - typ – klasifikátor, jehož instanci čára života znázorňuje
- Musí být jednoznačně identifikována v interakci názvem, typem nebo oběma
- Čára života má stejnou ikonu jako klasifikátor, který reprezentuje
 - Čára života jimůvÚčet reprezentuje instanci třídy Účet
 - Selektor [id = "1234"] určuje specifickou instanci třídy Účet id "1234"

Zprávy

- Zpráva vyjadřuje komunikaci mezi dvěma čarami života

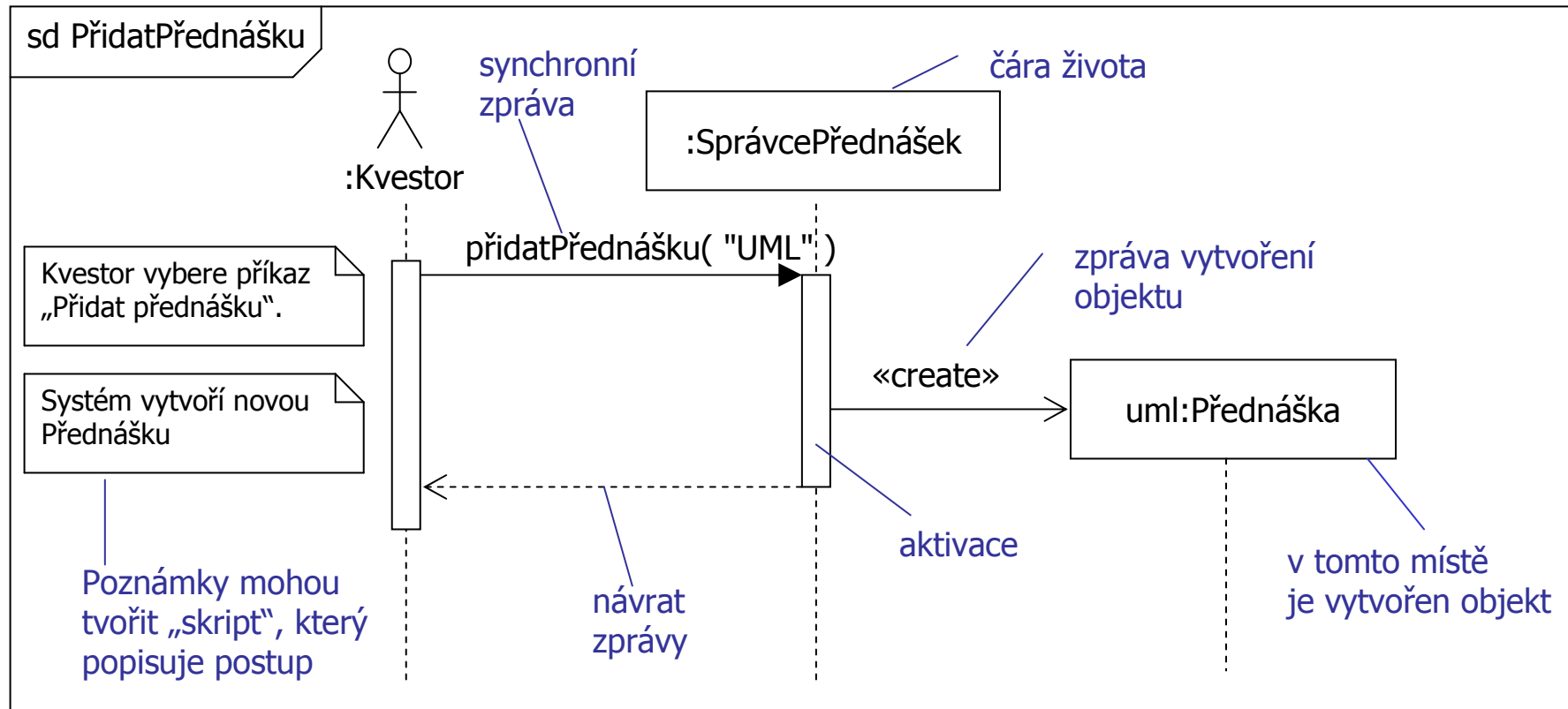
odesílatel  příjemce/cíl	Typ zprávy	Význam
	synchronní zpráva	Volání operace synchronně, kdy odesílatel čeká na dokončení operace.
	asynchronní zpráva	Volání operace asynchronně, kdy odesílatel nečeká na dokončení operace
	návrat zprávy	Návrat ze synchronní operace, kdy příjemce vrací aktivitu odesílateli.
 	vytvoření objektu uvolnění objektu	Odesílatel vytváří instanci klasifikátoru Odesílatel ukončuje život příjemce
	nalezená zpráva	Odesílatel je mimo interakci
	ztracená zpráva	Zpráva nebude nikomu doručena



Diagramy interakce

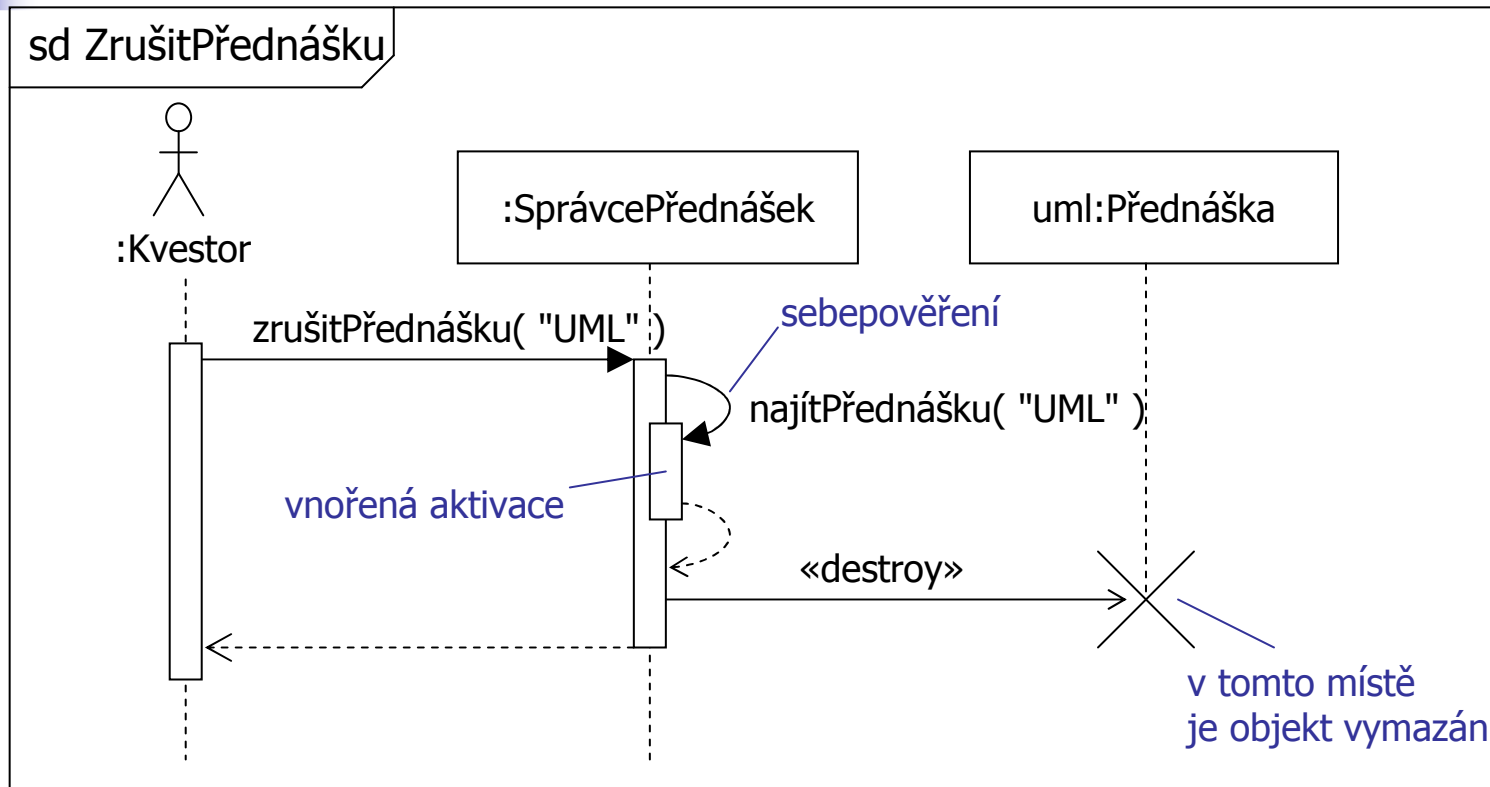
- Sekvenční diagramy
 - Zdůrazňují časově řazené sekvence zasílaných zpráv
 - Ukazují interakce v časové sekvenci
 - Jsou velmi přehledné a dá se jim dobře porozumět
 - Lze je velmi rychle přeplnit
- Komunikační diagramy
 - Zdůrazňují strukturální vztahy mezi objekty
 - Jsou velmi užitečné při analýze – zachycují vazby mezi objekty
- Diagramy zjednodušené interakce
 - Ukazují složité chování pomocí jednoduchých interakcí
- Diagramy časování
 - Zdůrazňují aspekty s časováním interakcí

Syntaxe sekvenčního diagramu



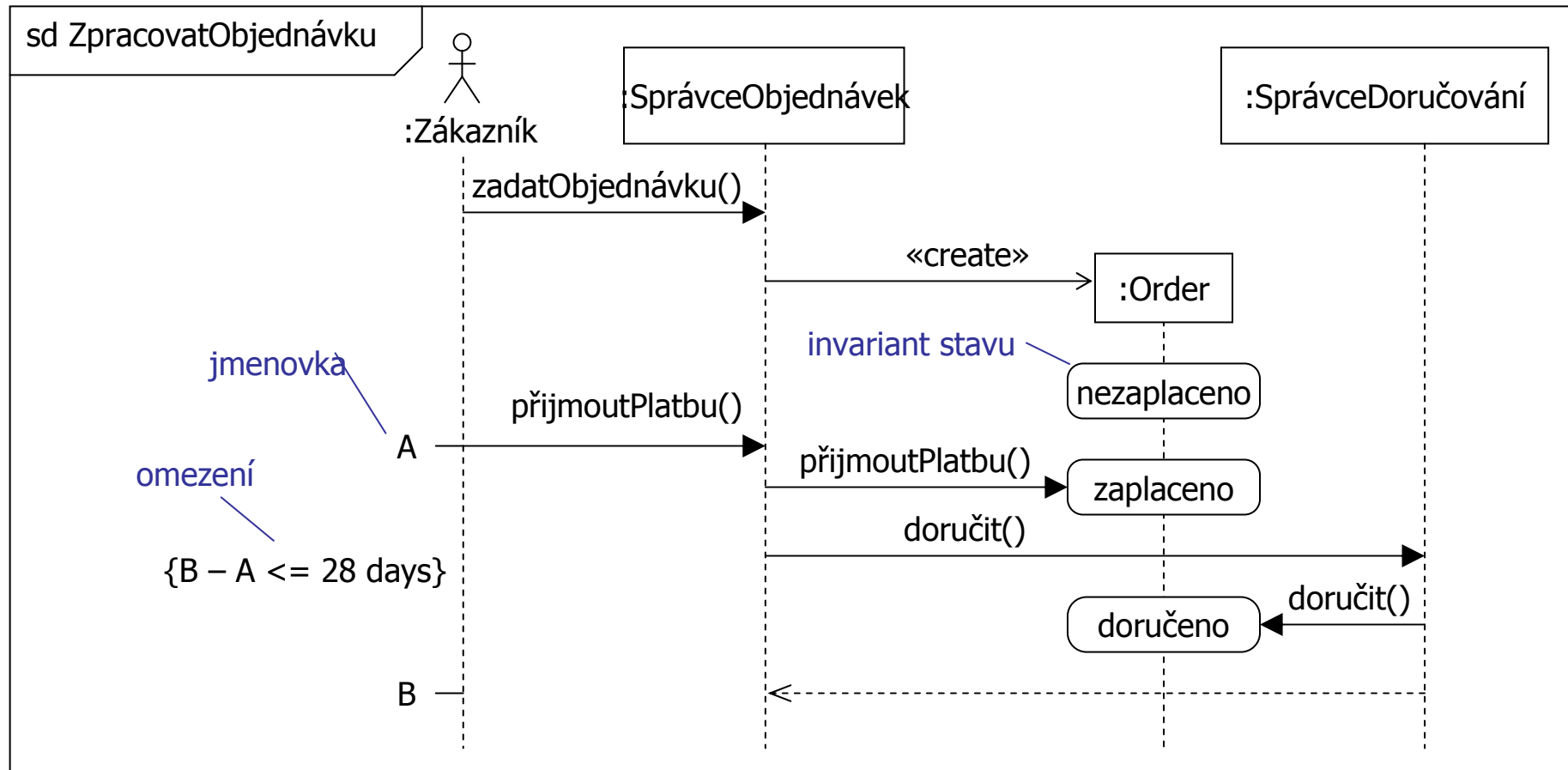
- Všechny sekvenční diagramy jsou označeny předponou sd, která indikuje typ
- Aktivace indikuje kdy je čára života aktivní

Rušení a sebepověření

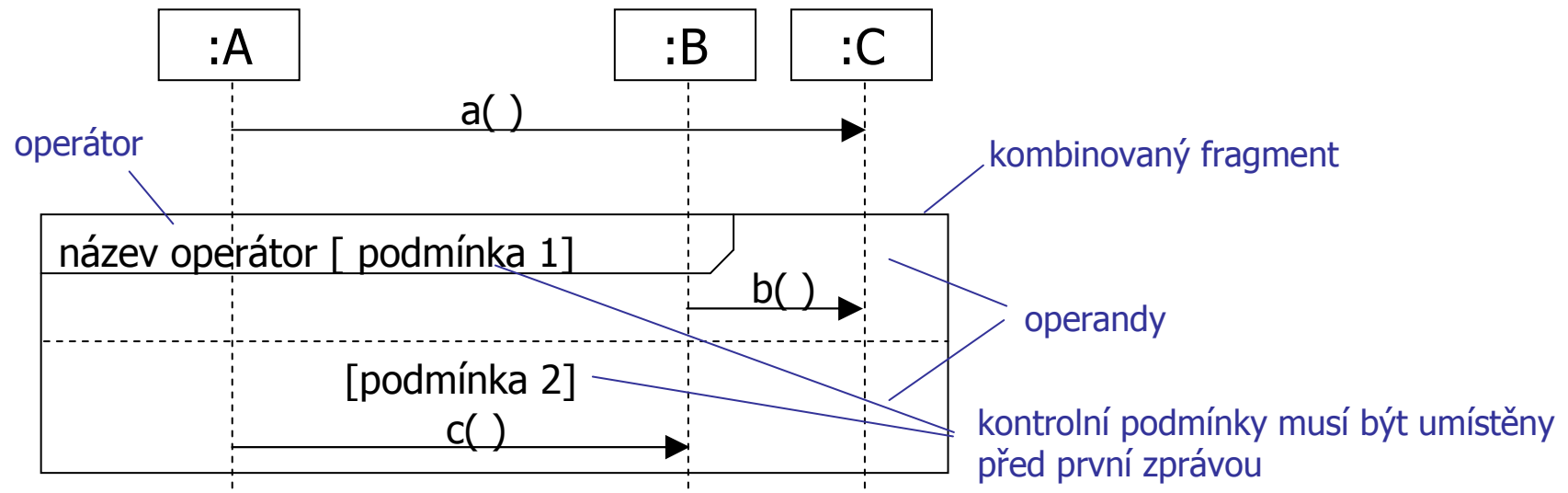


- Sebepověření nastává když čára života zasílá zprávu sama sobě
- Zrušení objektu se znázorňuje ukončením životní čáry velkým X

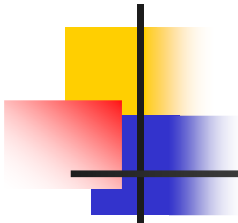
Invarianty a omezení stavu



Kombinované fragmenty



- Sekvenční diagramy mohou být rozděleny do polí, které nazýváme kombinované fragmenty
- Kombinované fragmenty mohou mít jeden nebo více operandů
- *Operátory* určují které operandy budou vykonány
- *Kontrolní podmínky* určují zda operandy budou vykonány.
- K vykonání dojde když je podmínka je vyhodnocena jako pravdivá



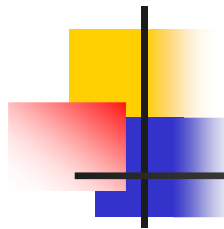
Seznam operátorů

operátor	dlouhé jméno	význam
opt	option	Pouze jeden operand, který se vykoná pouze tehdy, je-li podmínka splněna.
alt	alternatives	Provedou se operace s operandy, jejichž kontrolní podmínky jsou splněny. Místo logického výrazu lze použít klíčové slovo else.
loop	loop	Vyznačuje se zvláštní syntaxí: loop min, max [podmínka] Projít <i>min</i> -krát ale jen do <i>max</i> -krát dokud pokud je podmínka splněna
break	break	Po splnění podmínky je provedena pouze operace s operandem
ref	reference	Kombinovaný fragment odkazuje na další interakci

ref

```
najítStudenta(jméno):Student
```

ref má pouze jeden operand, který odkazuje na další interakci.



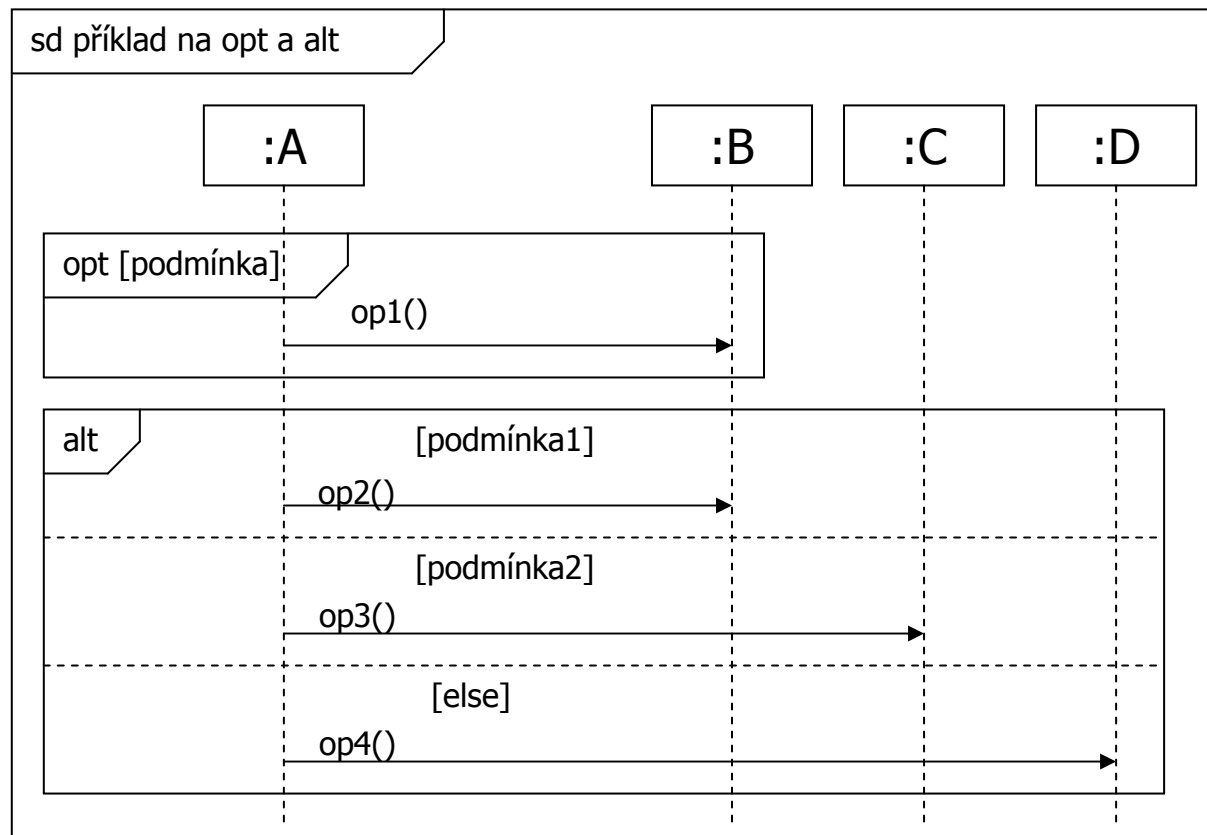
Seznam zbytku operátorů

- These operators are less common

operator	long name	semantics
par	parallel	Both operands execute in parallel
seq	weak sequencing	The operands execute in parallel subject to the constraint that event occurrences on the <i>same</i> lifeline from <i>different</i> operands must happen in the same sequence as the operands
strict	strict sequencing	The operands execute in strict sequence
neg	negative	The combined fragment represents interactions that are invalid
critical	critical region	The interaction must execute atomically without interruption
ignore	ignore	Specifies that some message types are intentionally ignored in the interaction
consider	consider	Lists the message types that are considered in the interaction
assert	assertion	The operands of the combined fragments are the only valid continuations of the interaction

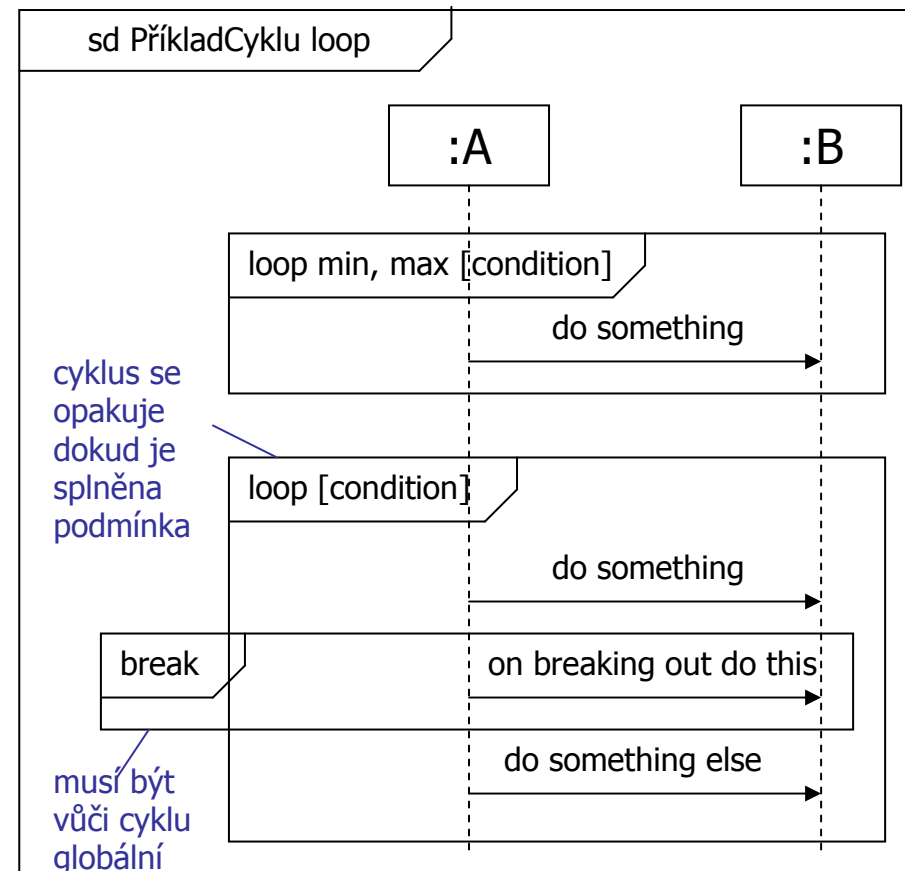
Větvení pomocí operátorů opt a alt

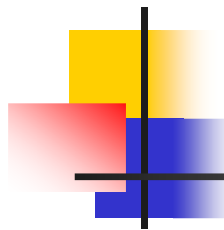
- význam opt:
 - jednoduchý operand, který se vykoná když je podmínka splněná
- význam alt:
 - dva nebo více operátorů, kdy každý má svoji podmínku
 - každý operand se vykoná když je splněná podmínka
 - operand s podmínkou **else** bude vykonán když nebude splněná žádná z ostatních podmínek



Iterace s operátory loop a break

- význam loop:
 - Opakuj min krát, dokud (max – min) krát je splněná podmínka
- syntaxe loop:
 - Cyklus bez min, max nebo podmínky je nekonečný cyklus
 - Když je pouze min je max = min
 - podmínka je
 - Boolean výraz
- Break specifikuje místo kde po splnění podmínky se přeruší vykonávání cyklu



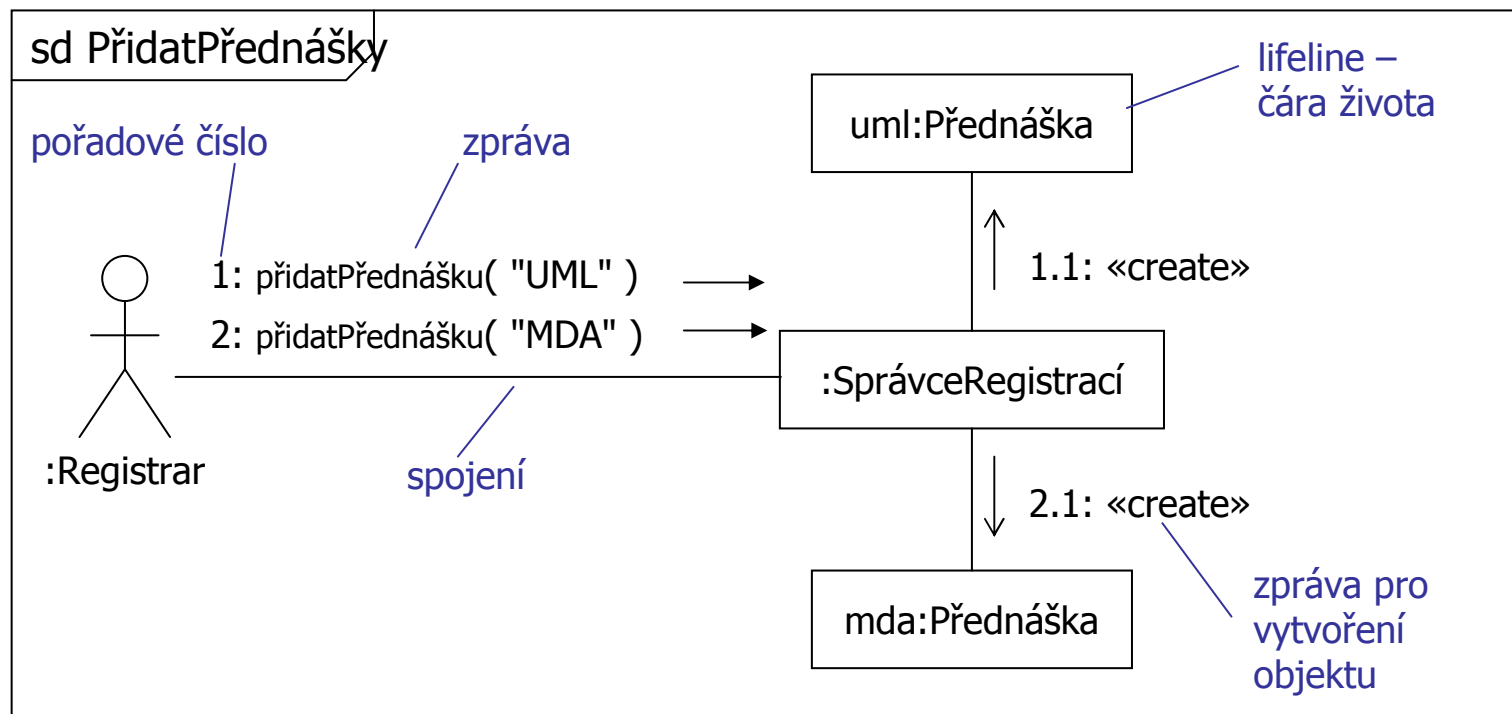


Různé cykly

typ cyklu	význam	výraz cyklu
while (true) {tělo}	nekonečný cyklus	loop *
for i = 1 to n {body}	opakovat (n) krát	loop n
while(booleanExpression) {body}	opakovat, dokud je podmínka splněna	loop [booleanExpression]
repeat {body} while(booleanExpression)	opakovat jednou a pak tak dlouho, dokud je podmínka splněna	loop 1, * [booleanExpression]
forEach object in set {body}	provést příkazy cyklu jednou pro každý objekt v kolekci objektů	loop [for each object in objectType]

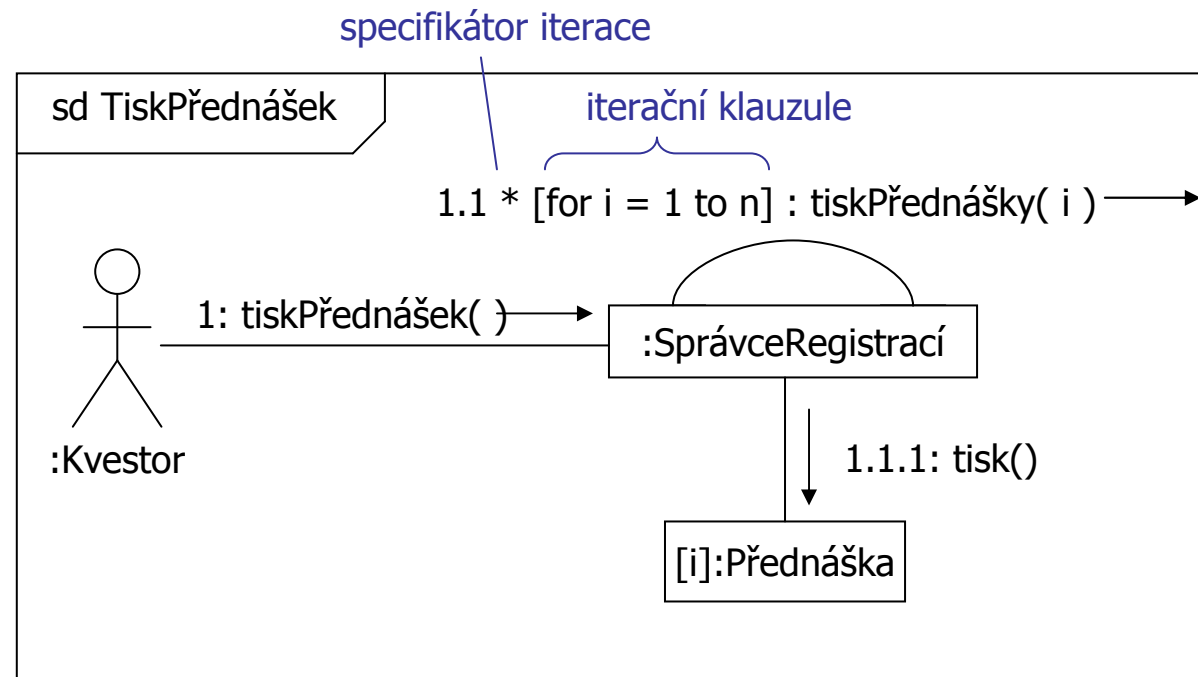
Syntaxe komunikačního diagramu

- Komunikační diagramy zdůrazňují strukturní aspekty každé iterace – jak jsou spojeny čáry života
 - V porovnání se sekvenčními diagramy jsou sémanticky slabé
 - Objektové diagramy je speciální případ komunikačních diagramů

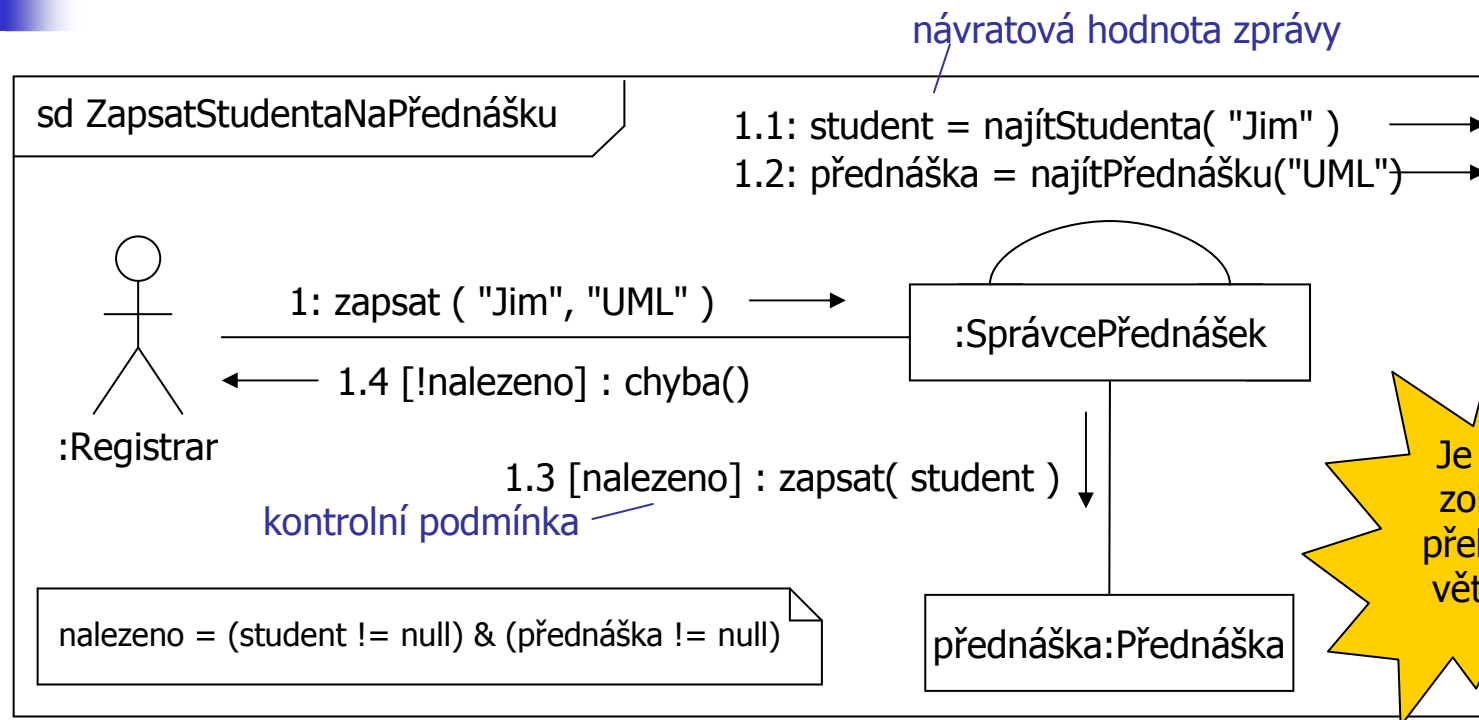


Iterace

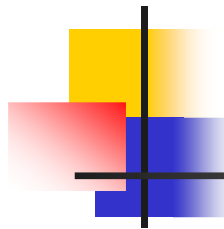
- Iterace je znázorněna specifikátorem iterace (*) a volitelně i iterační klauzulí
 - UML nepředepisuje syntaxi iterační klauzule
 - Lze užít kód nebo pseudokód



Větvění



- Větvení se modeluje předčísáním a kontrolní podmínkou
 - Syntaxe kontrolní podmínky není předepsána UML!
 - Zde je například použita pomocná proměnná **nalezeno**.



Souhrn

- V této přednášce jsme si ukázali realizaci případů užití použitím iteračních diagramů
- Ze čtyř typů diagramů jsem si ukázali dva:
 - Sekvenční diagramy – důraz na časové řazení sekvencí odesílaných zpráv
 - Komunikační diagramy – důraz na strukturální vazbu mezi čárami života (objekty)
- Ostatní iterační diagramy si ukážeme příště.