

SOFTVÉROVÉ MODELOVANIE

6.

Ján Ružbarský
Marek Tavač

Obsah 6. prednášky

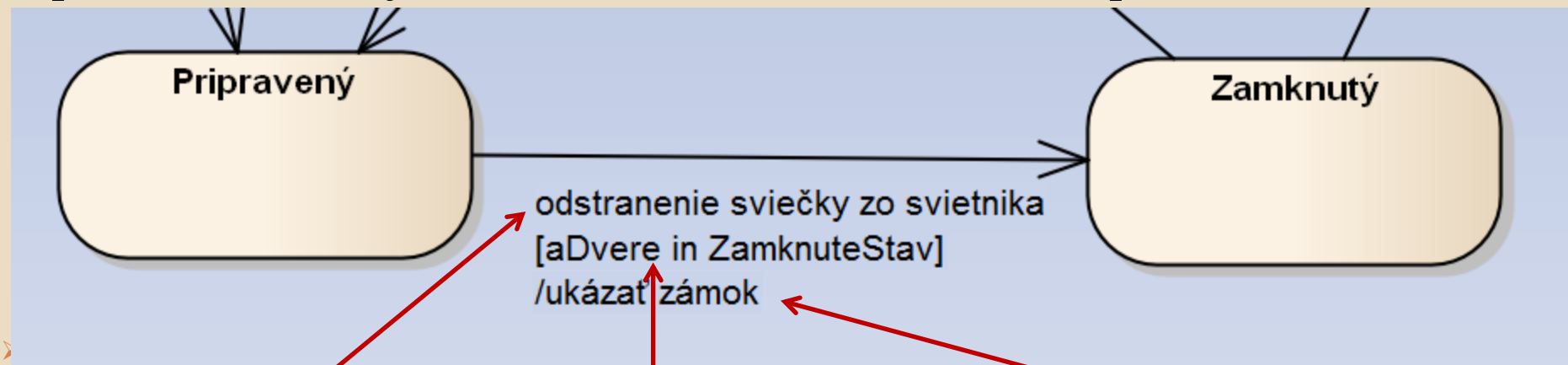
- *Opakovanie*
- *Interakcie*
 - Prvky
 - Správy
- *Sekvenčný diagram*
- *Komunikačný diagram*
 - Prvky v diagramoch
 - Príklady
- *Vaše otázky*

Opakovanie - Stavové diagramy

- *modelovanie stavov - chovanie systému v čase, časové zmeny UML modelu*
- *popisujú **životný cyklus prvkov***
- ***všetky stavy***
- *podmienky **prechodu medzi stavmi***
- **STAV** - *konkrétna situácia alebo podmienka, v ktorej sa prvok nachádza počas svojho životného cyklu*

Opakovanie - Prechody

- vzťahy medzi stavmi
- *plná čiara od zdrojového stavu k cieľovému zakončená šípkou*



Spúšťacia udalosť[ochranná podmienka]/efekt

Spúšťacia udalosť

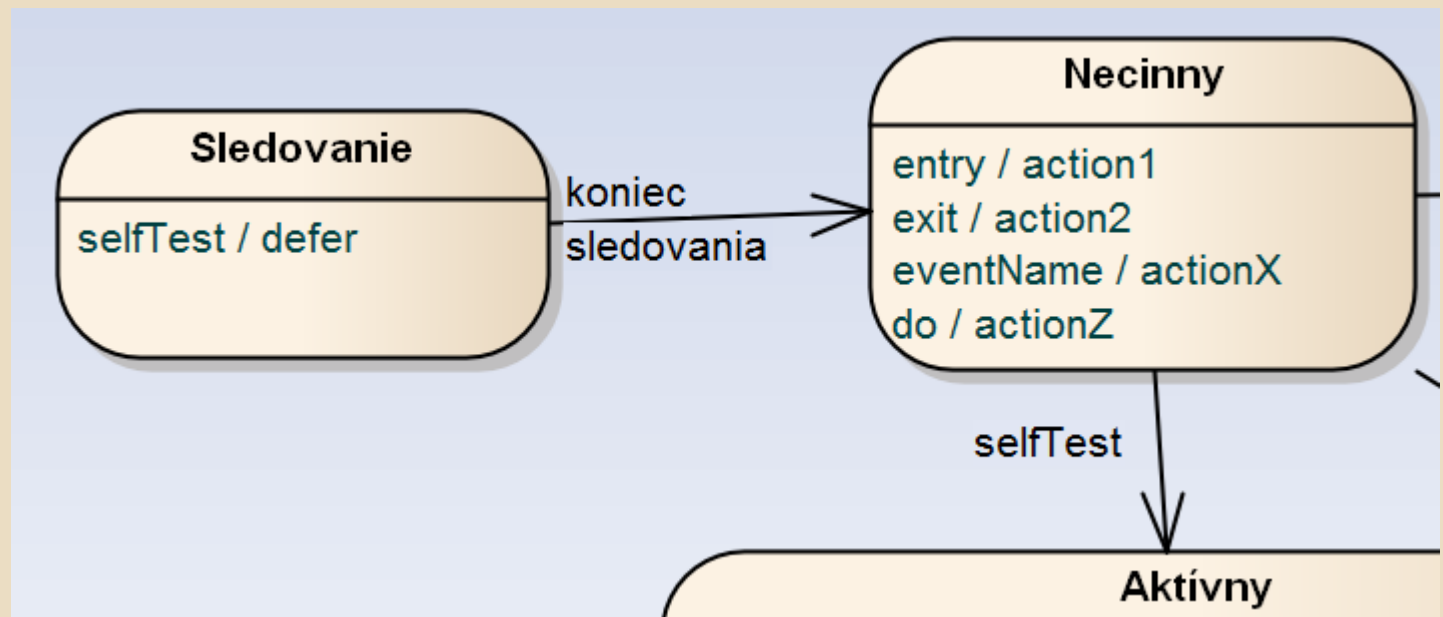
- *meno_udalosti (zoznam_parametrov)*

Efekt-akcia

- *návratová_premenná :=
cieľový_prvok.meno_akcie(zoznam_argumentov)*

Ďalšie možnosti

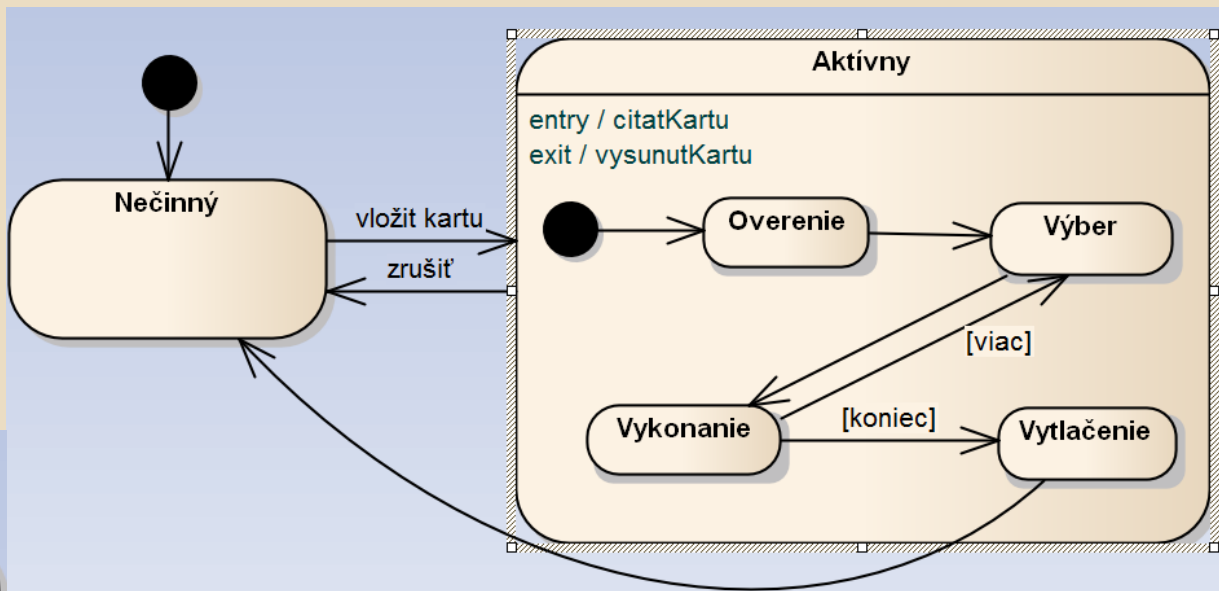
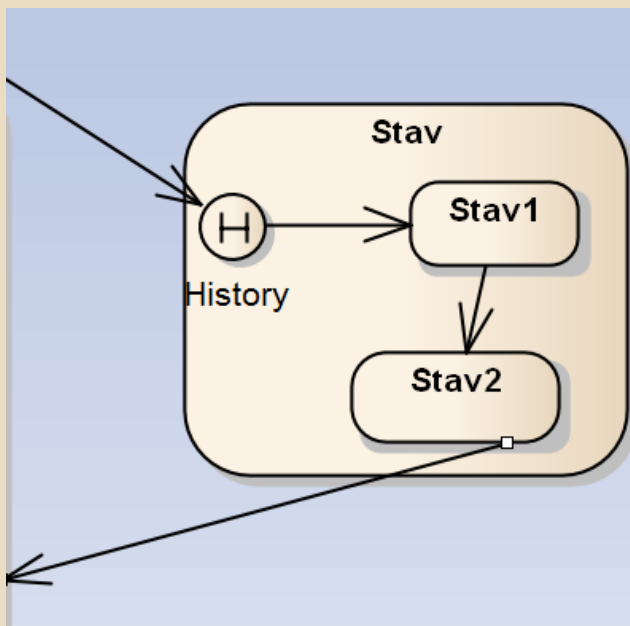
- *Vstupné/výstupné akcie (entry/exit effects)*
- *Interné prechody (internal transitions)*
- *Do-aktivity*
- *Odložené udalosti (deferred events)*



Zložené stavy a substavy

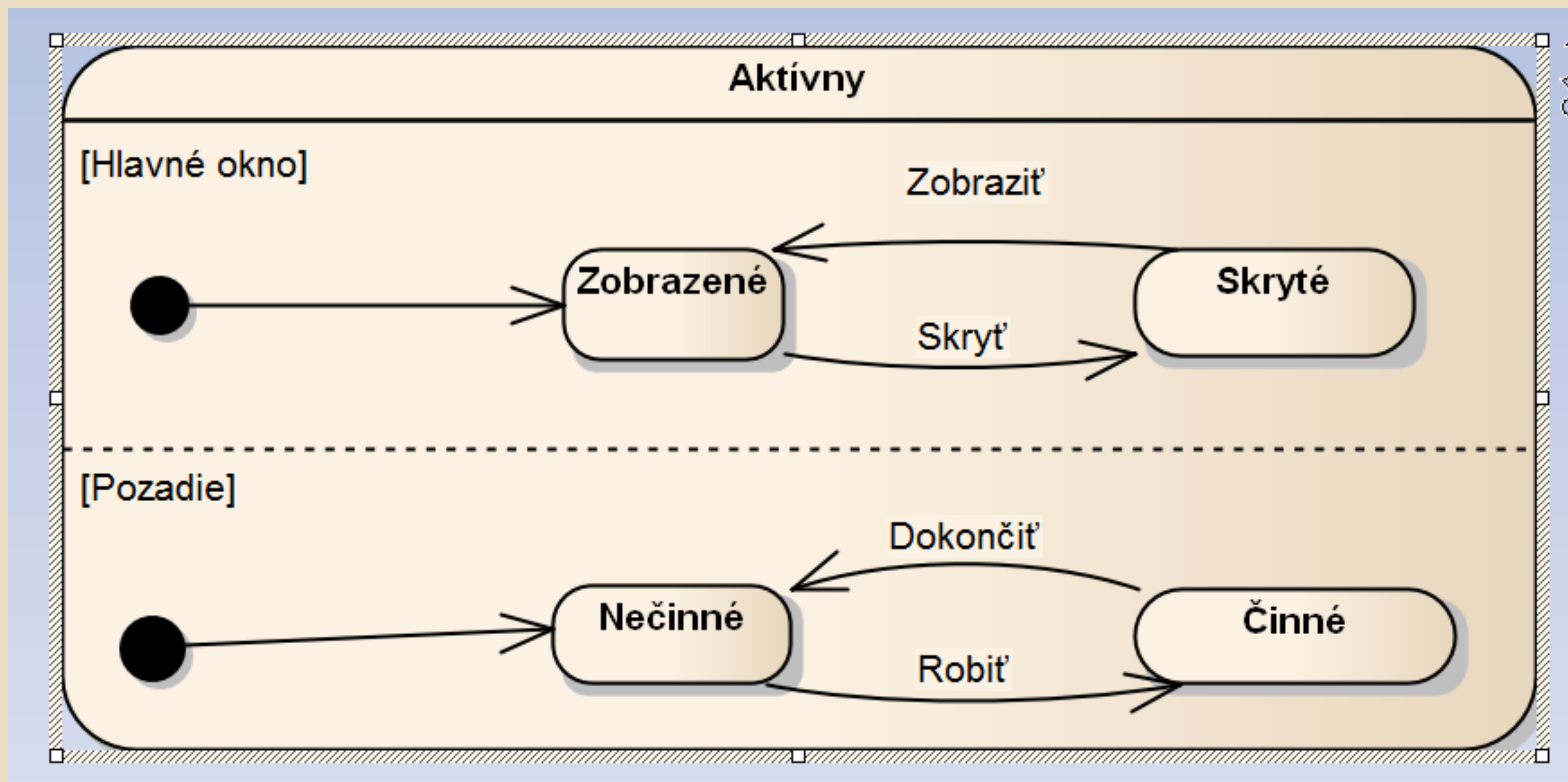
➤ *Sekvenčné – nonorthogonal, sequence*

➤ *History stavy*



Zložené stavy a substavy

➤ Súbežné – orthogonal, concurrent



Ako na to ?

- *Určiť čo modelujeme stavovým automatom ?*
 - *Trieda, use case alebo systém ako taký*
- *Počiatočný a koncový(-é) stav(-y)*
- *Hľadanie udalostí – interface*
- *Hľadanie top-level stavov*
- *Rozšírenie o ďalšie možnosti – ak je potrebné*
- *Kontrola*

Modelovanie interakcií

- *dynamické chovanie prvkov systému v čase*
- *vzájomné pôsobenie – **interakcia***
 - **správanie**
 - **množina správ**

- **modelovanie interakcie**
 - špeciálny prípad modelovania chovania
 - ako sa prvky ovplyvňujú v čase
- *2 spôsoby*
 - zameranie na časový priebeh
 - zameranie na štrukturálne vzťahy
- *vo fáze analýzy a návrhu*
- *začína – ak máme dost' požiadaviek*
- *pokračuje paralelne s modelovaním tried a objektov*

Interakcie

- *pojem **Správa***

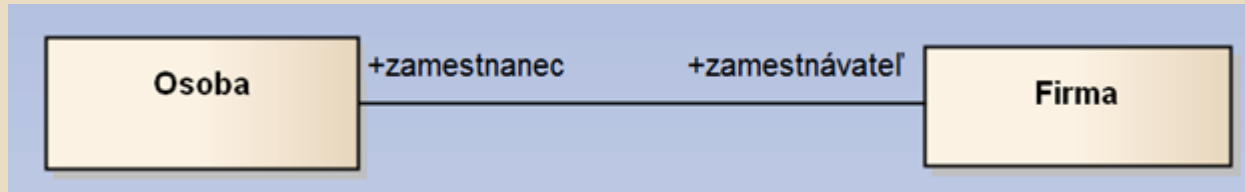
Kde stretneme interakcie:

- *Spolupráca objektov v systéme*
- *Operácie*
- *Trieda*

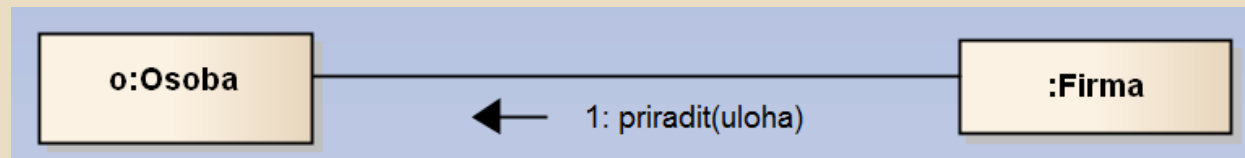
- **Objekty a roly**
 - **Konkrétne veci** – objekt popisuje konkrétnu inštanciu skutočného sveta
 - **Typické veci** – objekt popisuje hocakú inštanciu triedy
 - hrajú **ROLU** – častejšie používané
- **Väzby a konektory**
 - Spojenie medzi objektmi – inštancia asociácie
 - Typická väzba – **KONEKTOR**
- Obmedzenie na konci väzby
 - association, self, global, local, parameter

Príklad

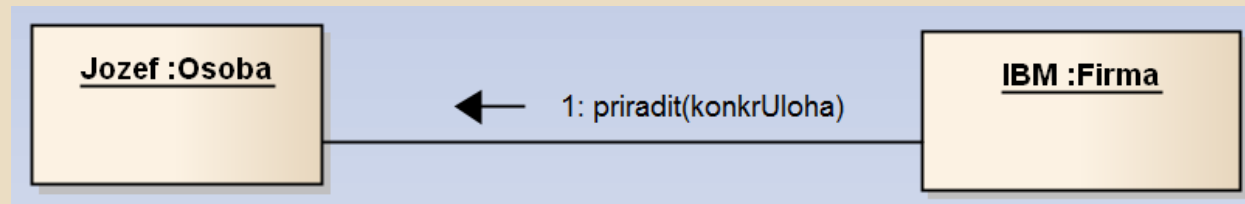
➤ Deklarácia tried



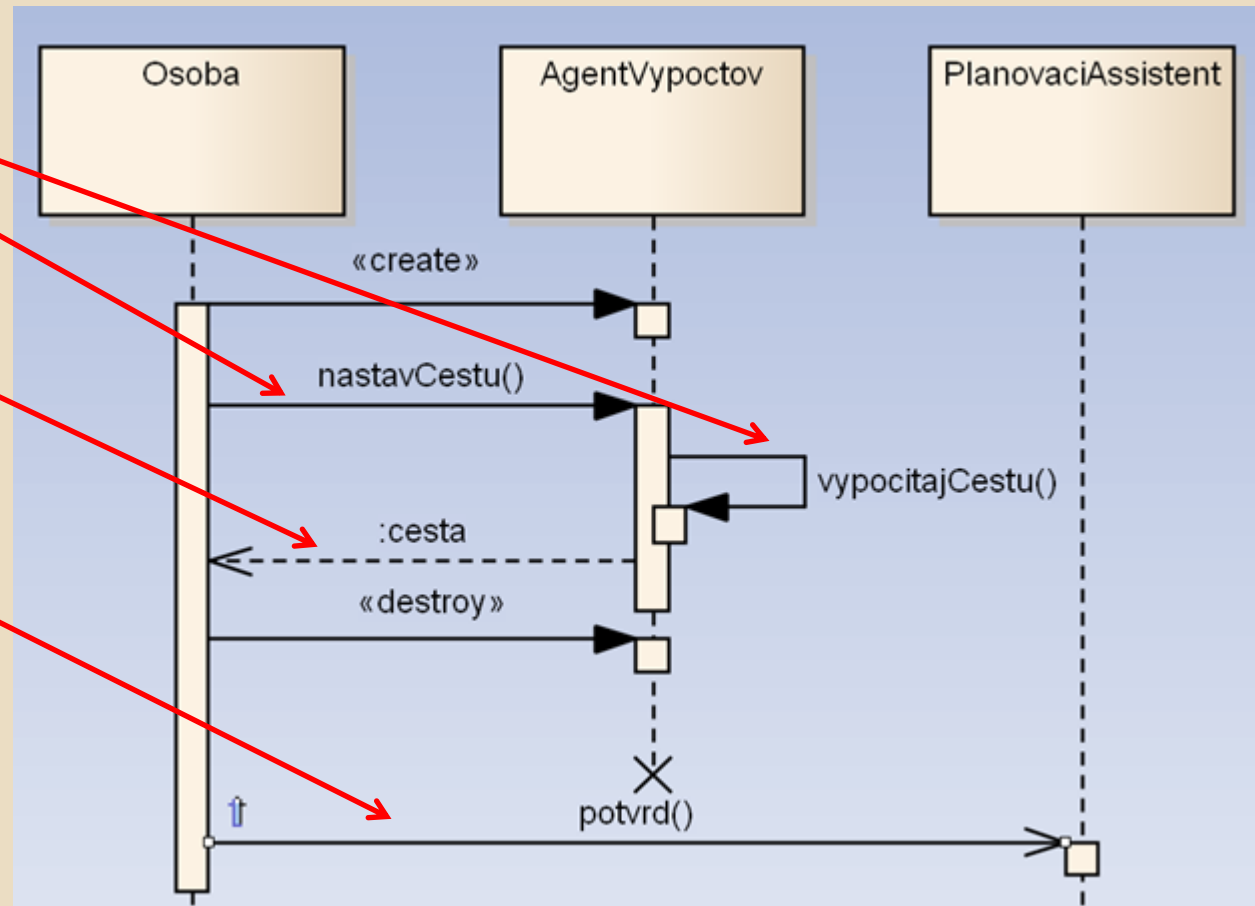
➤ Role a konektory



➤ Objekty a väzby



- *Call*
- *Return*
- *Send*
- *Create*
- *Destroy*



- *Informácia posielená asynchrónne*
- *Po poslaní – posielaajúci objekt pokračuje*
- *Pri prijatí – prijímajúci objekt sa nezávisle rozhoduje čo urobí*
 - Spúšťanie prechodov pri zmene stavu

Sekvencia správ

- *Postupnosť, prúd správ*
- *Začiatok – proces, vlákno*
- *Koniec – životnosť procesu, vlákna*
- *Nonstop systém*
- *Číslo poradia správy - nepovinné*

- *Vytvorenie prvku*
- *Zrušenie prvku*
- *Modifikácia prvku*
- *Reprezentácia*
 - Interakčné diagramy

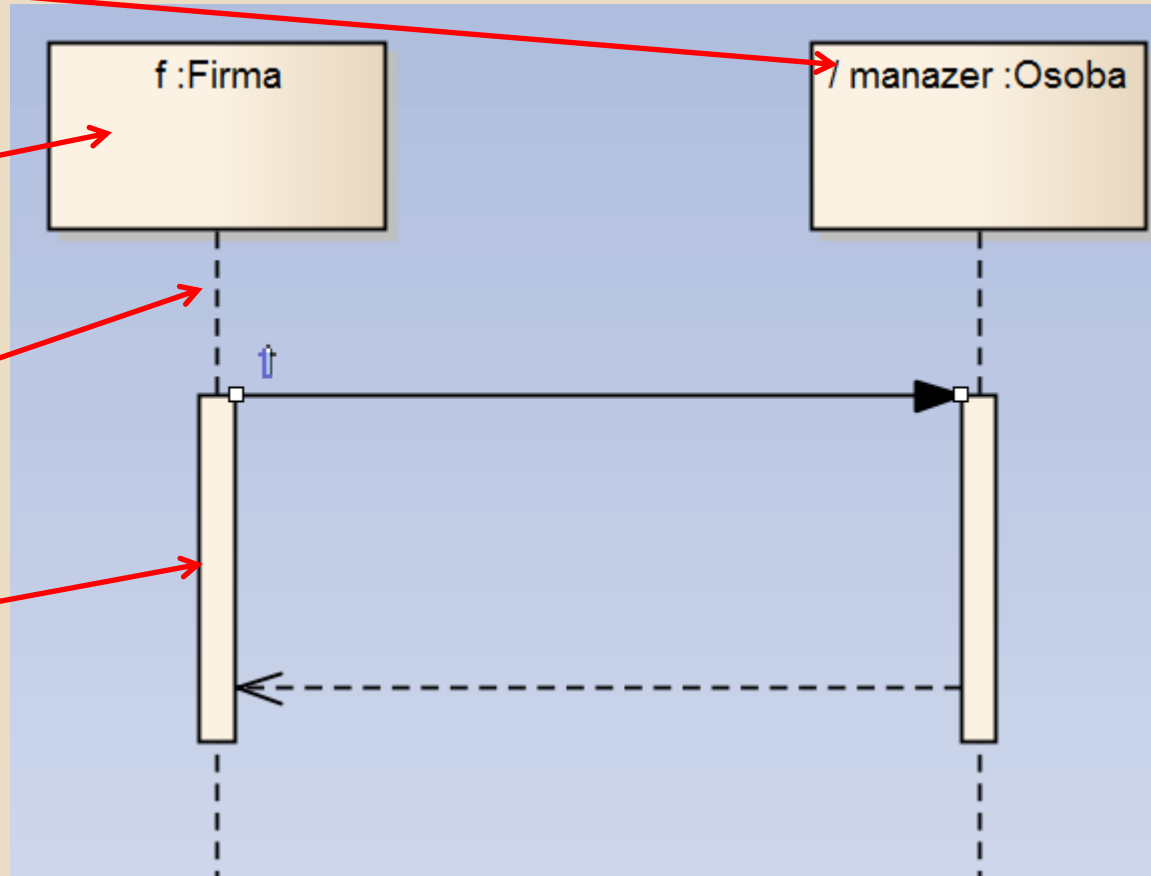
Interakčné diagramy

- *Ukazujú vzájomné pôsobenie prvkov*
- *Roly alebo objekty*
- *Komunikácie alebo väzby*
- *Správy*
- *Obmedzenia a poznámky*
- ***Sekvenčný (sequence) diagram***
- ***Komunikačný (communication) diagram***
- ***Časový (timing) diagram***

Sekvenčné diagramy

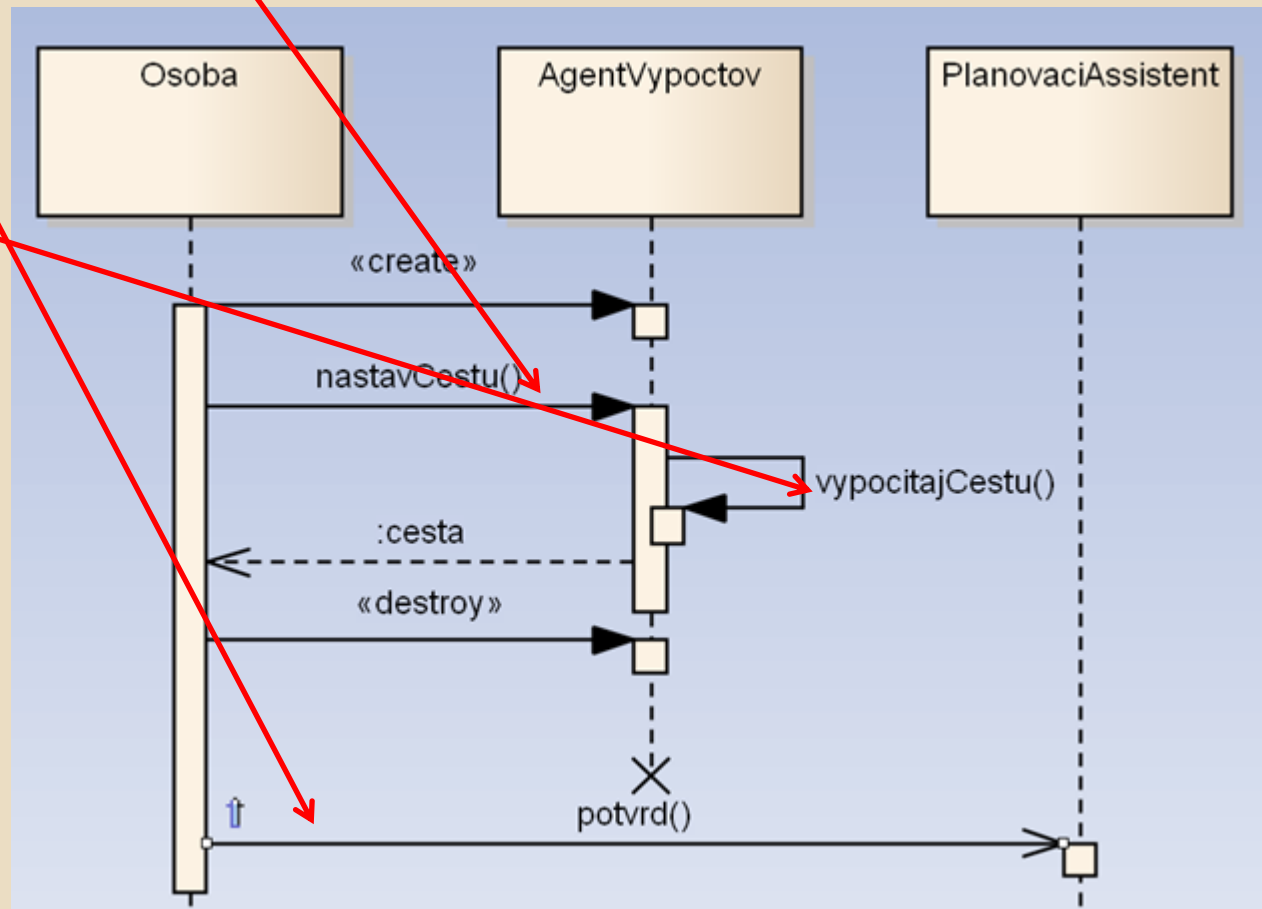
- ako na seba prvky vzájomne pôsobia **v čase**
- **horizontálna osa** ukazuje **prvky**, ktoré sú zahrnuté v interakcii
- **vertikálna osa** reprezentuje **čas**

- *Roly*
- *Objekty*
- *Čiary života*
- *Aktivácie*



Komunikácia

- Synchronná správa
- Asynchrónna správa
- Reflexná (self)



[podmienky] *[iterácia] číslo_sekvencie :
návratová_premenná := meno_operácie
(list_argumentov)

Podmienky

voliteľný parameter, indikuje podmienky, ktoré musia byť splnené pre komunikáciu.

Hranaté zátvorky nepíšeme, ak nie je žiadna špecifikovaná

Číslo sekvencie

voliteľné celé číslo, indikuje poradie komunikácie.

Nasledujúca dvojbodka sa nepíše, keď nie je špecifikované.

*[podmienky] *[iterácia] číslo_sekvencie :
návratová_premenná := meno_operácie
(list_argumentov)*

návratová premenná

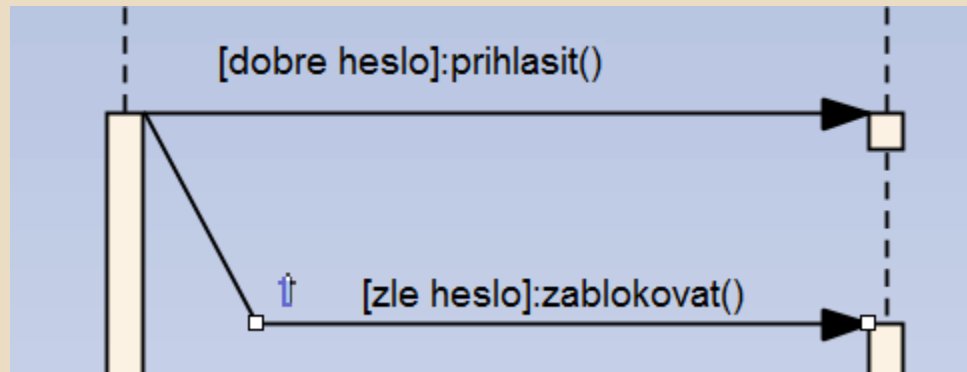
- nepovinná, meno pre hodnotu, ktorá je vrátená operáciou

meno operácie - meno vyvolanej operácie

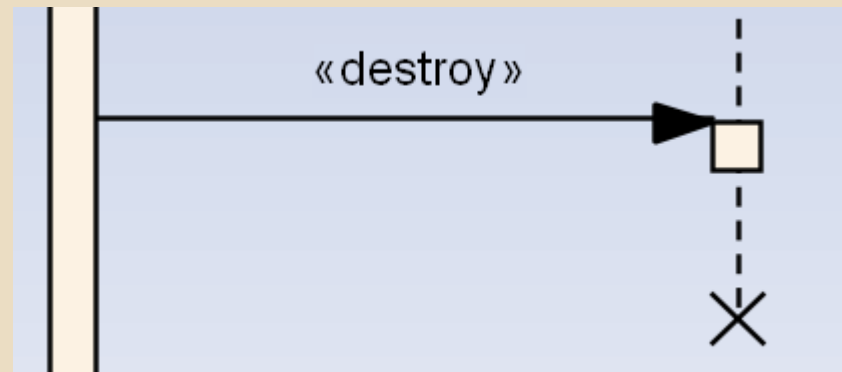
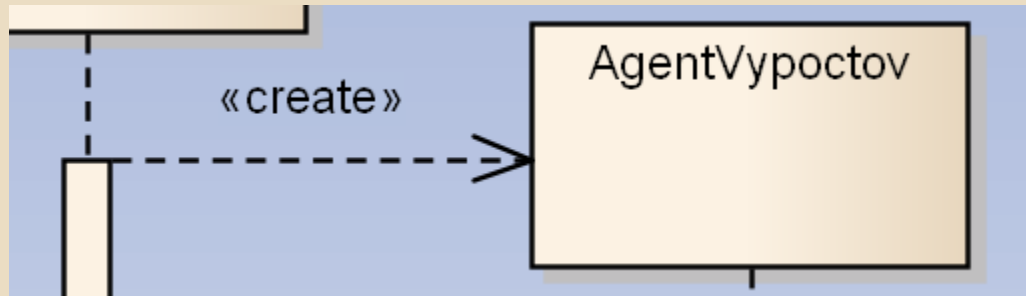
list argumentov

- nepovinný, čiarkou oddelený zoznam, ktorý indikuje argumenty pre operáciu.
- každý parameter môže byť hodnota alebo návratová premenná z predchádzajúcej komunikácie.

Podmienka



Vytvorenie a zrušenie prvku

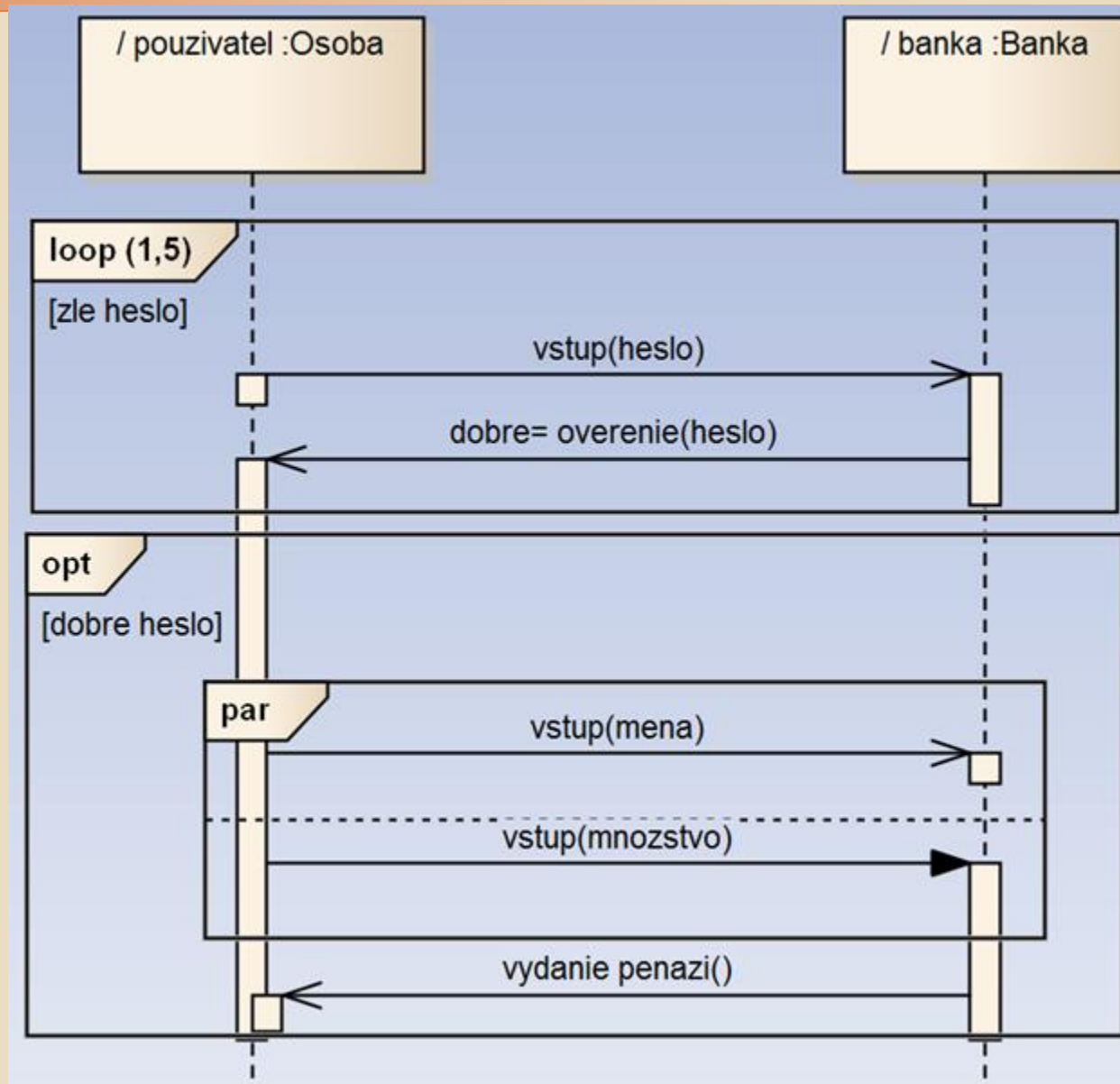


Operátory riadenia (control operators)

- *Postupnosť správ spojená podmienkou, cyklom*
- *Tag označuje druh operátora*
- *Telo operátora – čiary života a správy, ktoré ho pretínajú*

Operátory riadenia

- *Optional*
 - opt
- *Parallel*
 - par
- *Conditional*
 - Alt
- *Loop – cyklus*
 - loop



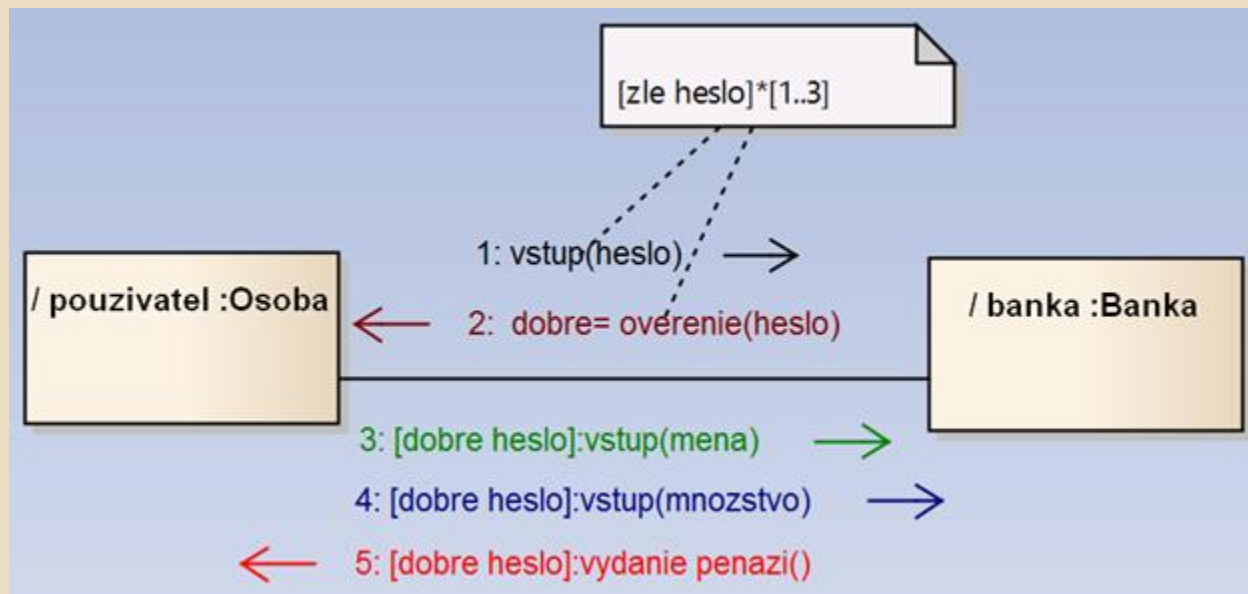
- *predtým diagram spolupráce (**collaboration** diagram)*
- *podobné informácie ako v sekvenčnom diagrame*
 - prvky na seba pôsobia v čase
 - **dôraz – vzťahy medzi prvkami**

➤ ***Sekvenčný***

- časovo orientovaný, zobrazuje celkový prúd interakcií
- vhodný pre komplexný popis interakcií – číta sa zhora dole

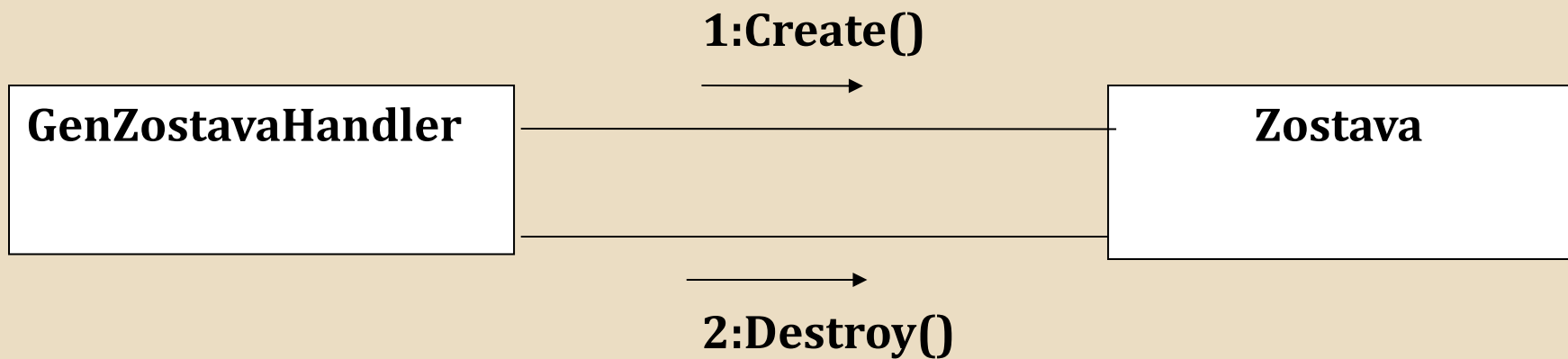
➤ ***Komunikačný***

- časovo aj priestorovo orientovaný, obsahuje prvky a ich vzťahy
- vhodný na vizualizáciu dopadov interakcií na rôzne prvky – je vidno prvky, s ktorými spolupracuje



- *správy, stimuly – šípka vedľa vzťahu od odosielateľa k príjemcovi*
- *rovnaká notácia ako v sekvenčnom diagrame*
- *požadované číslo sekvencie*
- *dot notácia – pre rôzne úrovne komunikácie*
- *Reflexná komunikácia*

Vznik a zánik prvku



- *výber nejakej množiny správ alebo stimulov*
- *použitie dot notácie*
- *guard výraz – podmienka*
 - ak je podmienka splnená – môže sa spustiť aj komunikácia na nižšej úrovni – vid' príklad

- *v interakčných diagramoch sú možné problémy, ak majú popisovať alternatívy*
- *možné riešenie – **CRC Cards***
- ***Class Responsibility Collaboration**
(trieda zodpovednosť spolupráca)*
- *použitie kariet nie diagramov na znázornenie modelu*
- *zodpovednosť – popis cieľa triedy*

CRC Cards

Organizácia	
Zodpovednosť	Spolupráca
Zistiť druh práce	Projekt
Zistiť produkt práce	Projekt

➤ <http://c2.com/doc/oopsla89/paper.html>

➤ *Rebecca Wirfs-Brock – Alan McKean:*

Object Design“ Roles Responsibilities and Collaborations

Ako na to ?

- *Čo modelujeme interakciou – systém, triedu, operáciu ?*
- *Identifikácia objektov, rolí, pomenovanie rolí*
- *Ak modelujeme štruktúru – väzby*
- *Postupnosť správ*

Ako na to ?

- *Dva typy diagramov , ale to isté inak*
- *Radšej viac diagramov jednoduchších ako jeden zložitý*
- *Dôraz buď na poradie správ alebo postupnosť správ v kontexte štruktúry – nie obe naraz*
- *Vetvenie a podmienky – striedmo*
- *Používajte poznámky, farby*
- *Minimum krížení čiar*

Otázky ?

Ďakujem za pozornosť