

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

---



## UML Fragmenty

Tím 4 - Bubbles

---

Predmet: *Tímový projekt I*

Vedúci projektu: *Ing. Ivan Polášek, PhD.*

Téma projektu: *3D UML*

Autor Dokumentu: *Lukáš Markovič*

Typ Dokumentu: *Analýza*

Akademický rok: *2014/2015, zimný semester*

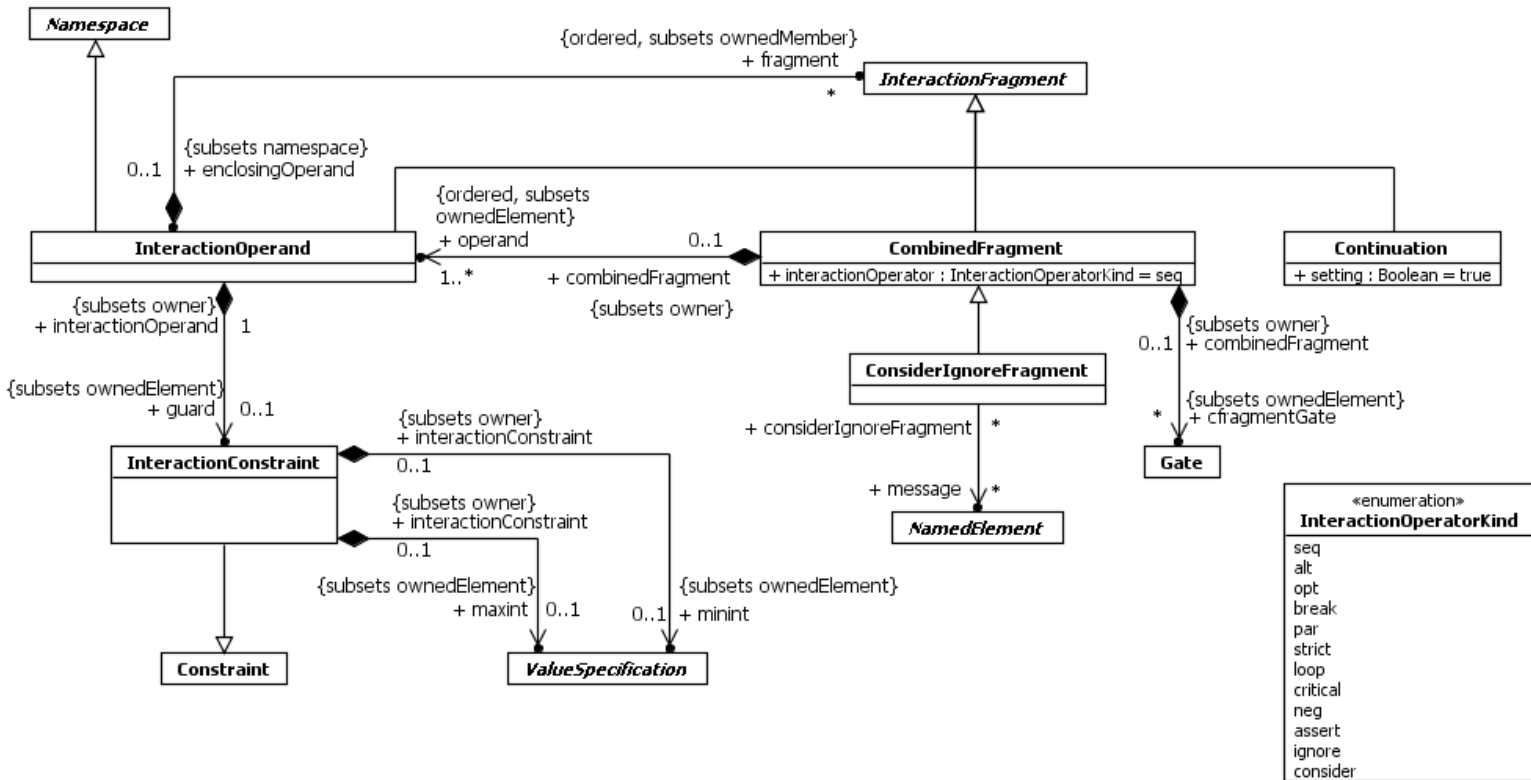
# Verzia dokumentu

<b>VERZIA</b>	<b>ZMENY</b>	<b>DÁTUM</b>
<b>1.0</b>	Vytvorenie dokumentu	13.10.2014
<b>1.1</b>	Rozšírenie kapitoly „Abstraktná syntax“	25.10.2014
<b>1.2</b>	Pridaná kapitola „notácia“	

# Obsah

Verzia dokumentu.....	1
1. Abstraktná syntax .....	4
2. Kombinované fragmenty.....	5
3. Notácia .....	11
4. Príklady kombinovaných fragmentov zo špecifikácie UML .....	12
5. Zdroje .....	17

# 1. Abstraktná syntax



Obr. 1 Metamodel fragmentu

## Interakčný operand:

Predstavuje oblasť vyhradenú Kombinovaným fragmentom. Aby bol operand vykonaný, musí mať pravdivé ohraňenie. Ak ohraňenie nie je definované, automaticky sa berie, ako pravdivé.

## Interakčné ohraňenie:

Používajú sa v kombinácii s kombinovanými fragmentami.

## Kombinovaný fragment:

Sémantika je závislá na interakčnom operátore. Prvok „Gate“ reprezentuje syntaktické rozhranie medzi kombinovaným fragmentom a jeho okolím.

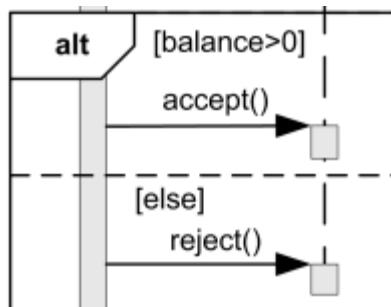
## 2. Kombinované fragmenty

Kombinované fragmenty sú fragmenty, ktoré definujú výraz, na základe interakčných fragmentov. Kombinačné fragmenty sú definované pomocou interakčného operátora a interakčných operandov.

Medzi interakčné operátory patria:

### Alternatives:

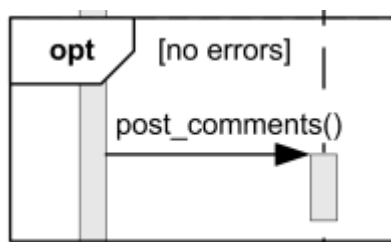
Operátor označuje možnosť výberu medzi viacerými tokmi. Výber operandu závisí na definovanom ohraňení, ktoré musí byť explicitne uvedené



Obr. 2 Príklad alternatív

### Option:

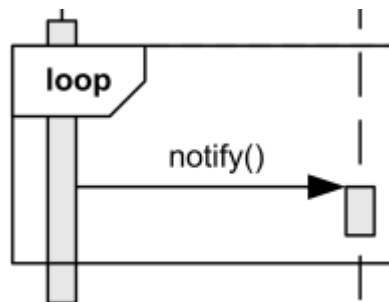
Operátor označuje možnosť, medzi vykonaním operadnu, alebo nevykonaním. Rozhodnutie o vykonaní závisí na vyhodnotení ohraňujúcej podmienky, ktorá musí byť explicitne stanovená.



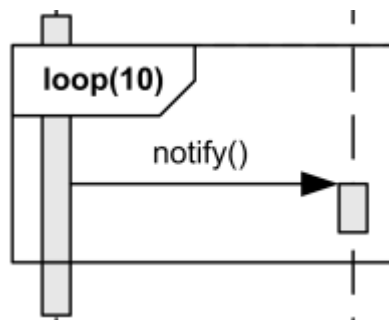
Obr. 3 Príklad možnosti

## Loop:

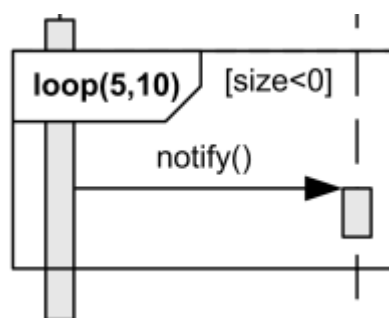
Operand v operátore „loop“ sa vykoná stanovený počet krát. Počet, koľo krát sa operand vykoná závisí na stanovenom ohraničení. V prípade, že ohraničenie nie je stanovené, ide o nekonečný cyklus. V prípade zadanie jednej hranice sa cyklus vykoná presne stanovený počet krát. Je možné taktiež zadať hornú aj dolnú hranicu. Okrem toho môže byť uvedené aj ďalšie ohraničenie, ktoré podmieňuje samotné spustenie cyklu.



Obr. 4 Nekonečný cyklus



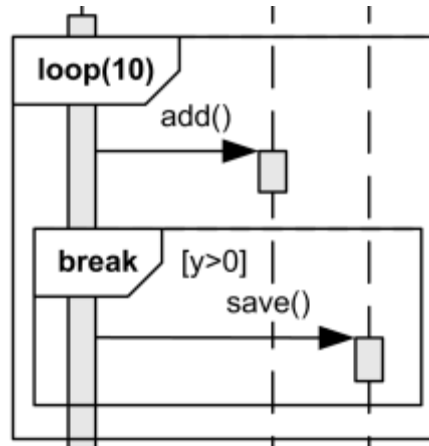
Obr. 5 Cyklus opakujúci sa 10 krát



Obr. 6 Cyklus s minimálnym aj maximálnym ohraničením

**Break:**

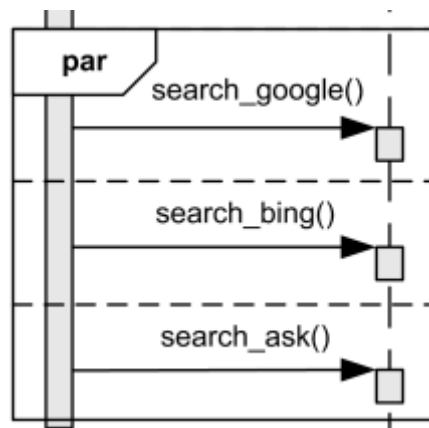
Operátor sa používa na ukončenie fragmentu, do ktorého je vnorený. Ukončenie nastáva, ak je stanovené ohraničenie splnené. V prípade, že podmienka nie je uvedená ide o nedeterministické správanie.



Obr. 7 Příklad Ukončenia

**Parallel:**

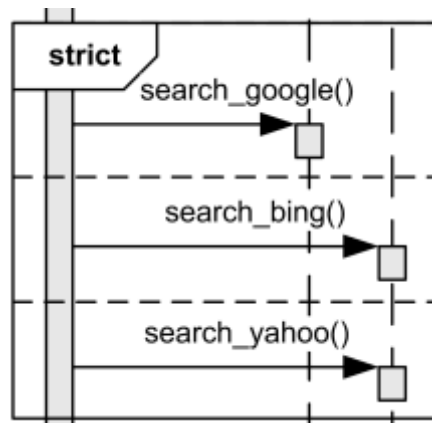
Označuje paralelizmus. Každá časť sa vykoná paralelne s ostatnými časťami operátora.



Obr. 8 Příklad paralelizmu

### Strict Sequencing:

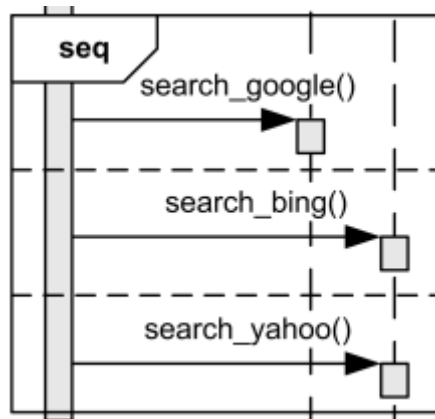
Operátor označuje, že operátory musia byť vykonané v presne stanovenom poradí.



Obr. 9 Prísna sekvencia

### Weak Sequencing:

Operátor označuje sekvenciu, ktorá, na rozdiel od predchádzajúceho prípadu, nemusí byť prísne po poradí. Konkrétne to znamená to, že operandy na jednej línii sa musia vykonať v poradí, no na rôznych líniiach na poradí nezáleží.

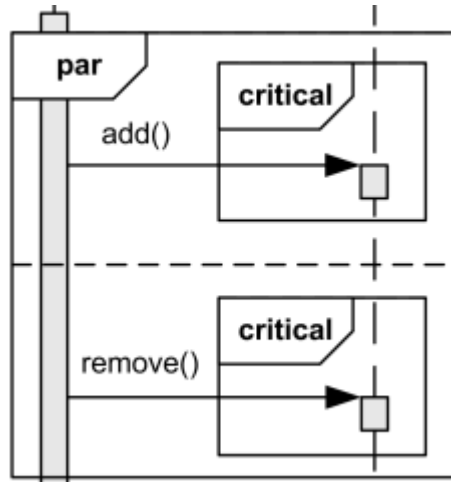


Obr. 10 Sekvencia



### Critical Region:

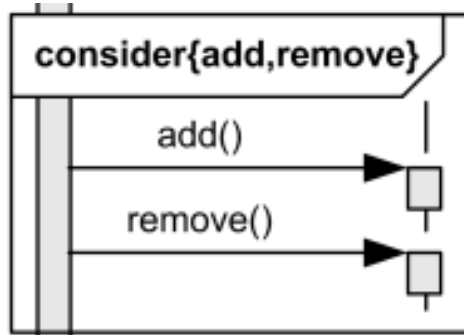
Operand špecifikuje kritický región, ku ktorému je nutné pristupovať atomicky.



Obr. 11 Príklad kritickej oblasti

### Consider:

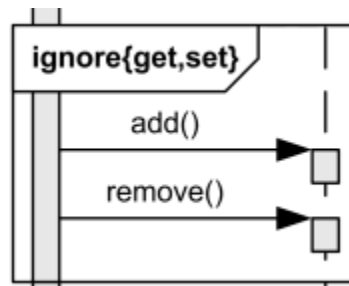
Operátor označuje oblasť, v ktorej musí byť zvážené, ktorá z poskytnutých správ bude zvolená. Zoznam poskytnutých správ na výber sa musí nachádzať v „{}“ zátvorkách. Iba jedna správa bude volaná, ostatné sa ignorujú.



Obr. 12 Príklad zváženia

### Ignore:

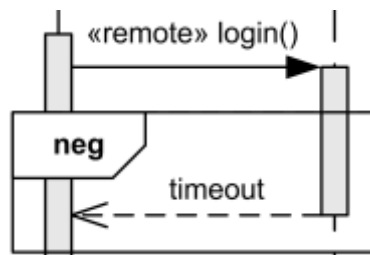
Označuje časti, ktoré nemajú byť zobrazené.



Obr. 13 Príklad Ignorovania

### Negative:

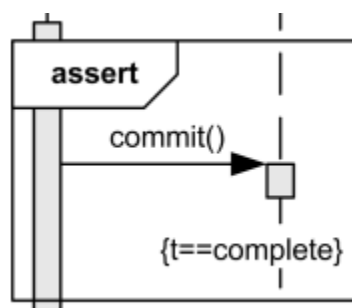
Operátor označuje oblasť, ktorá sa vykonáva v prípade neúspešného volania. Napríklad v prípade pádu systému. Všetky ostatné typy fragmentov sú považované za pozitívne.



Obr. 14 Príklad negative

### Assertion:

Operátor označuje oblasť, ktorá musí byť úspešne ukončená pre korektné pokračovanie programu.



Obr. 15 Príklad Assertu

## 3. Notácia

### **Interakčný operand“**

Interakčné operandy sú od seba oddelené vodorovnou prerušovanou čiarou. Spolu tvoria orámovaný kombinovaný fragment. V sekvenčnom diagrame je poradie operandov dané vertikálnou polohou operandu.

### **Interakčné ohraňenie:**

Interakčné ohraňenie sa uvádza v hranatých zátvorkách. Majú tvar:

$$\langle \text{interactionconstraint} \rangle ::= \text{'[ (} \langle \text{Boolean-expression} \rangle \text{ | 'else'} \text{' ) ']'}$$

Pokiaľ ohraňenie nie je uvedené, predpokladá sa pravdivé tvrdenie.

### **Kombinovaný fragment:**

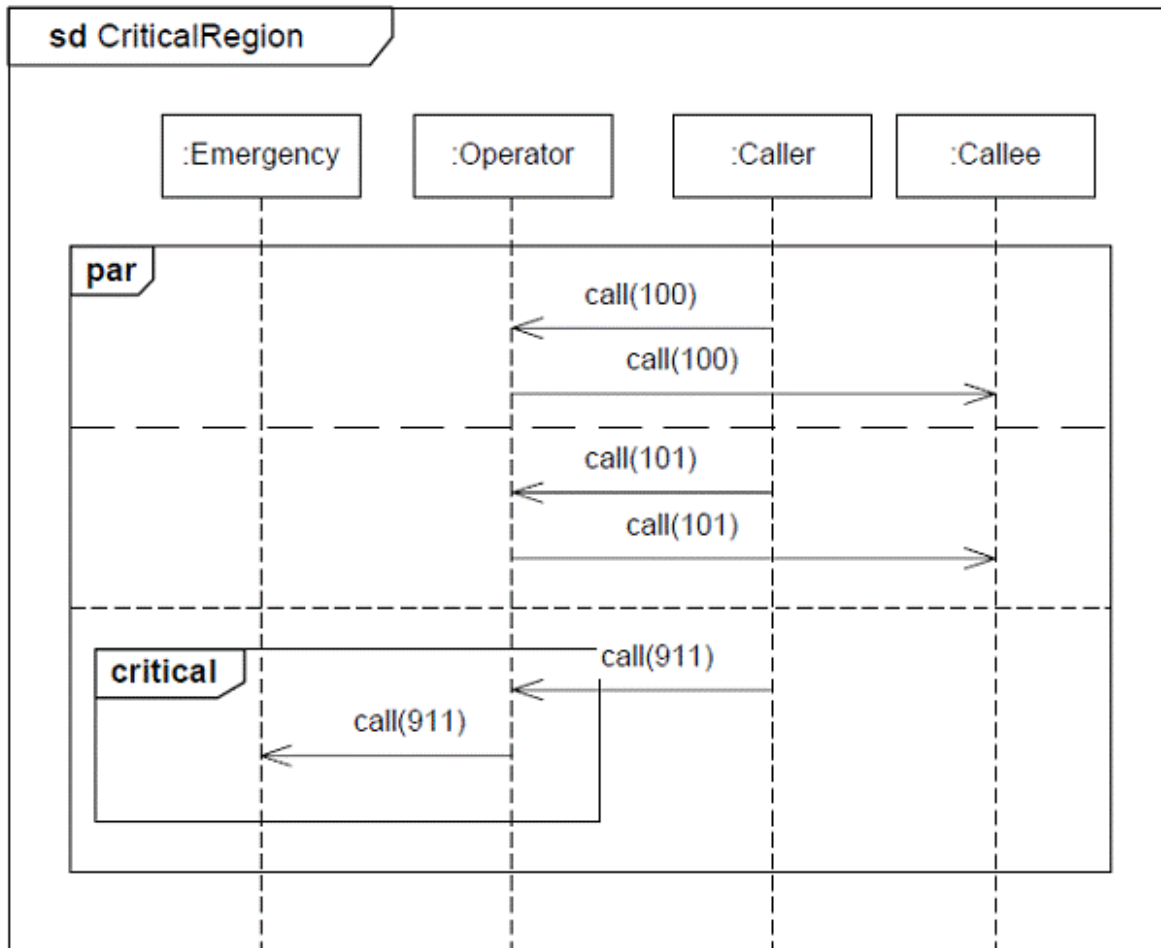
V sekvenčnom diagrame sa uvádza, ako obdĺžnik. Operátor je uvedený v päťuholníku v ľavom hornom rohu.

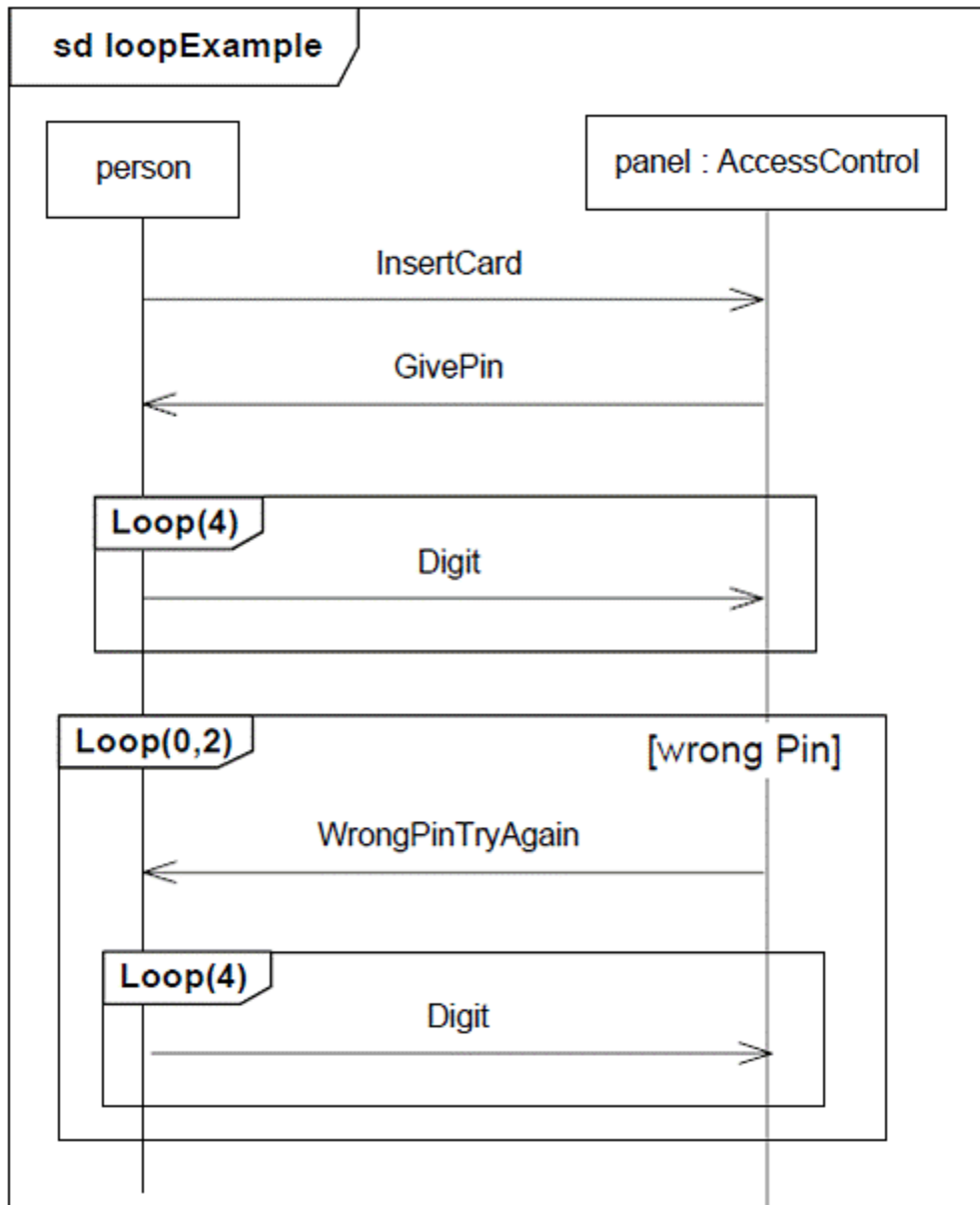
### **Consider / Ignore Fragment:**

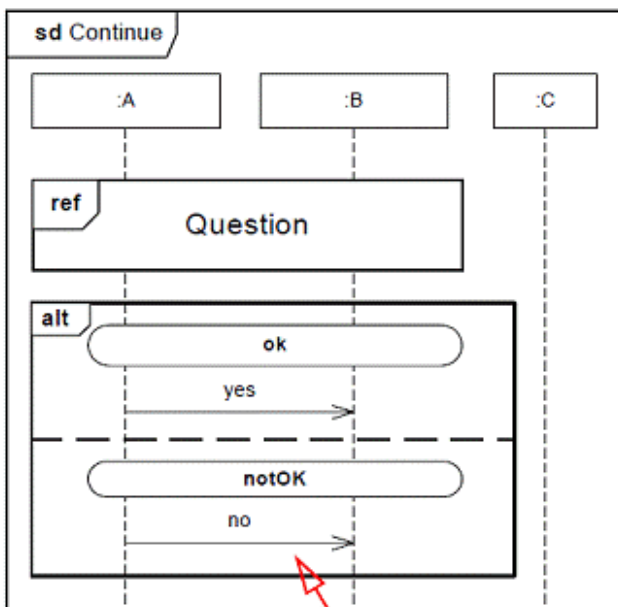
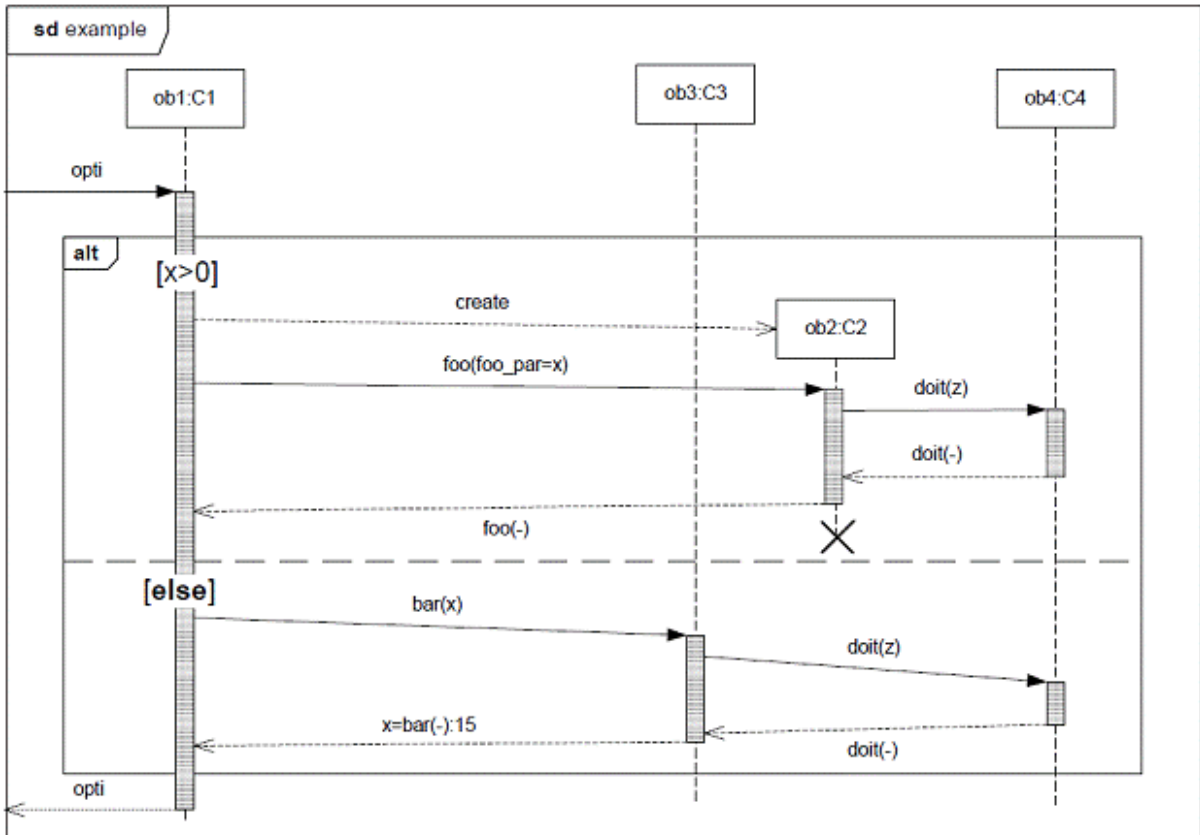
Na rozdiel od ostatných kombinovaných fragmentov sa za operátorom ešte uvádza zoznam parametrov uvedený v „{ }“.

$$\text{'(ignore' | 'consider')' '{' \langle \text{message-name} \rangle \text{' , ' } \langle \text{message-name} \rangle \text{'* '}'}$$

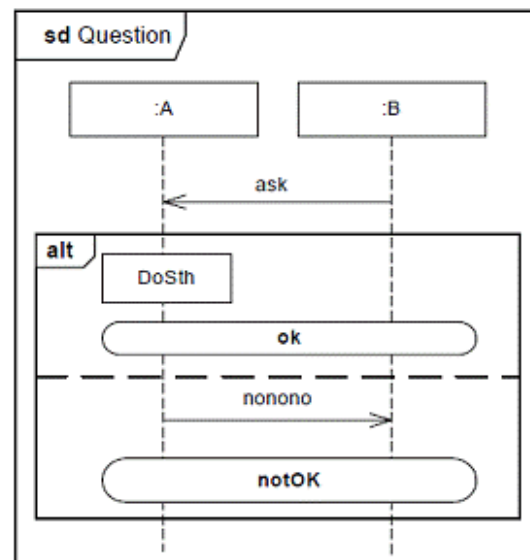
## 4. Príklady kombinovaných fragmentov zo špecifikácie UML



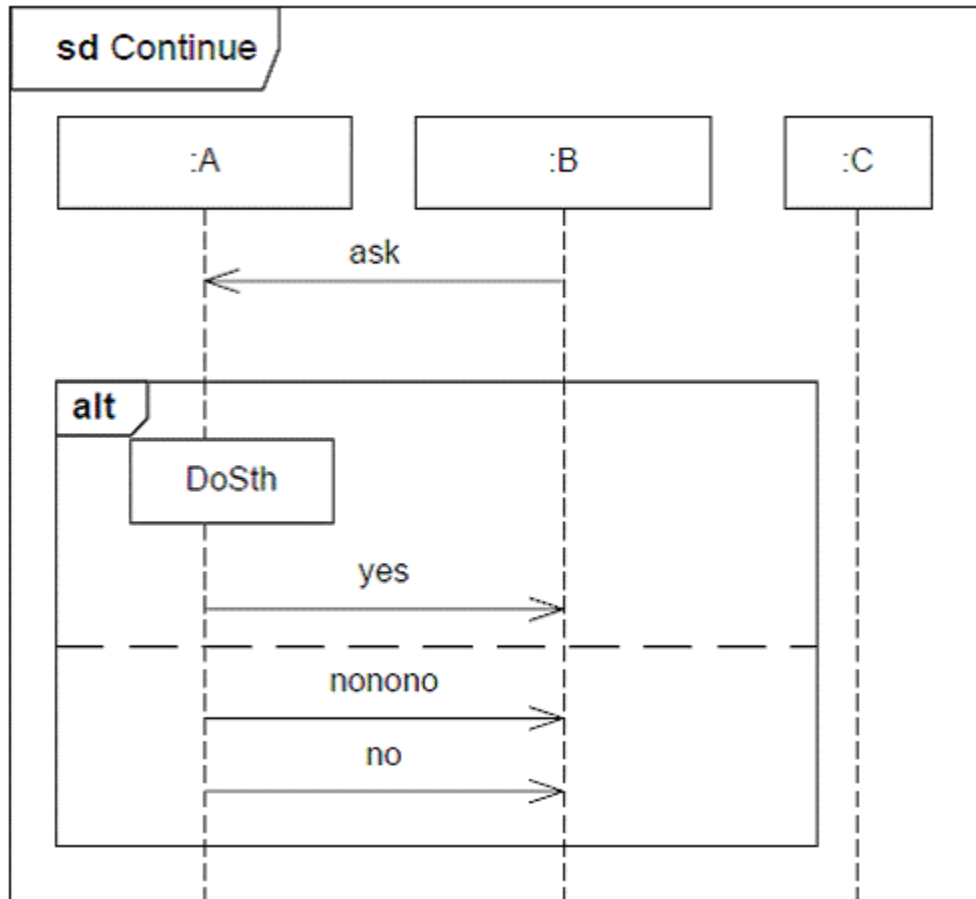


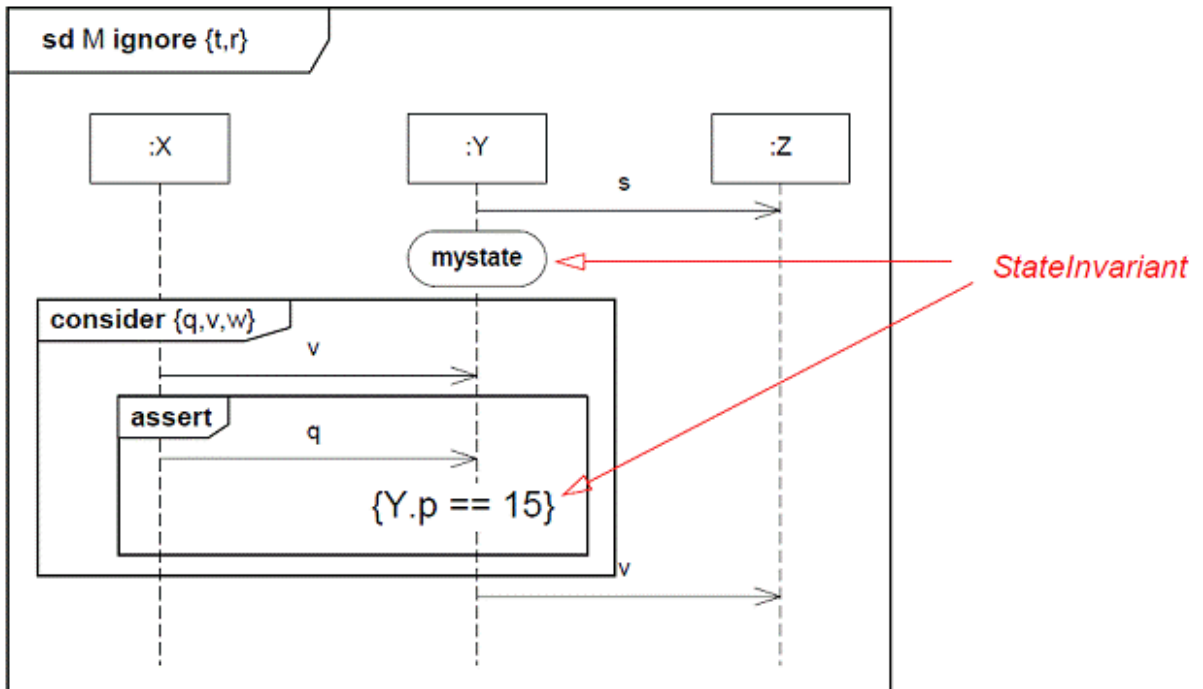


Continuation (setting==False)



Continuation (setting==True)









## 5. Zdroje

OMG Unified Modeling Language TM (OMG UML) Version 2.5

Combined Fragment - <http://www.uml-diagrams.org/sequence-diagrams-combined-fragment.html>