

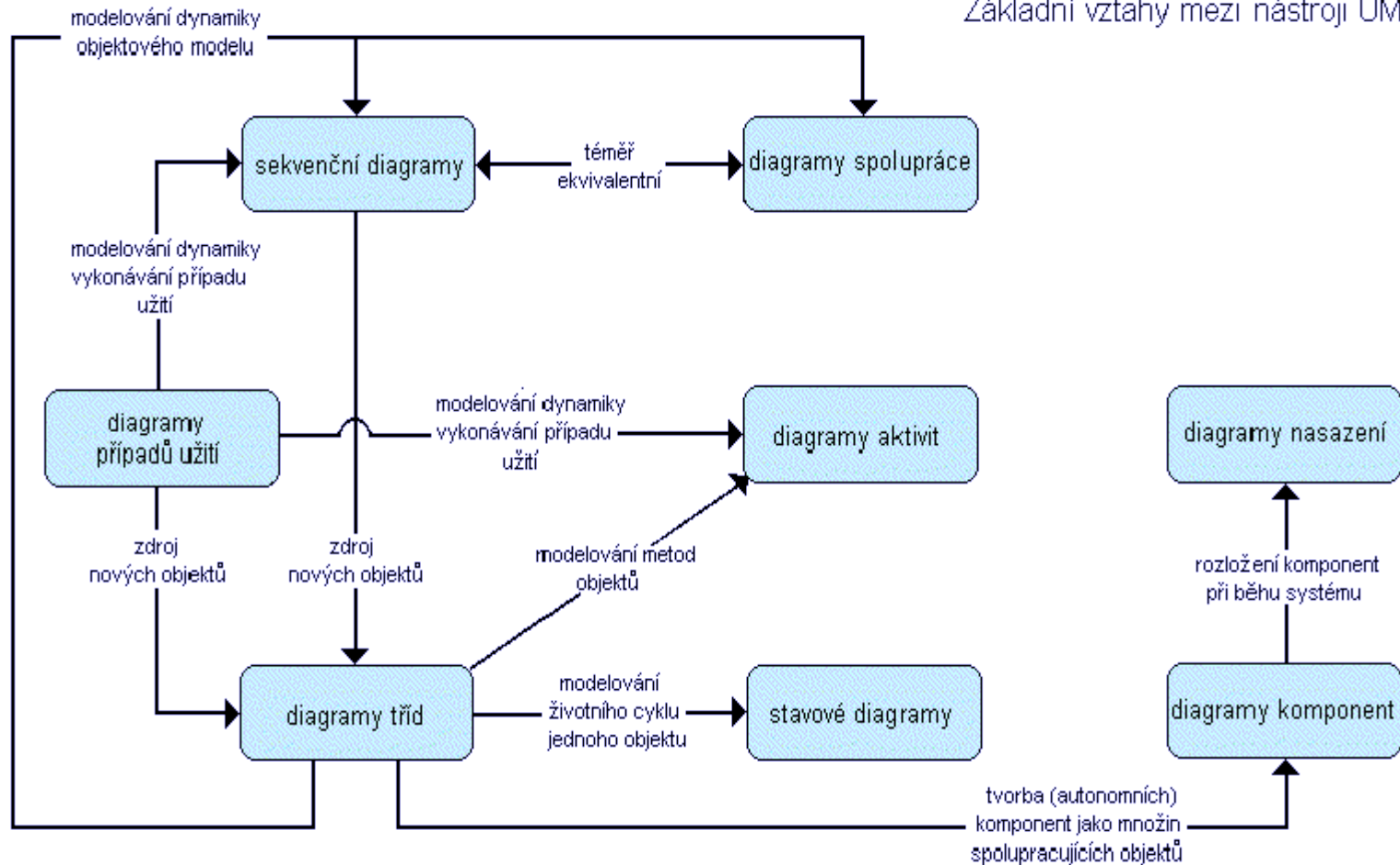
# UML

(diagram spolupráce)

# Začlenění:

- anglicky „collaboration diagram“
- Je jednou z možností jazyka UML, jak zobrazovat vzájemné vztahy mezi spolupracujícími objekty, jejich statickou strukturu a dynamické chování
- zachycuje interakci mezi objekty (struktura) s ohledem na časovou posloupnost
- popisuje komunikační vztahy a datové toky mezi jednotlivými objekty
- zvyšuje přehlednost mezi objekty

## Základní vztahy mezi nástroji UML



# Vztah k sekvenčnímu diagramu:

- Sekvenční diagram ukazuje interakci mezi objekty s ohledem na časovou posloupnost
- Diagram spolupráce zobrazuje tutéž informaci, ale zaměřuje se spíše na statickou strukturu komunikujících objektů.
- V sekvenčních diagramech je návaznost posílaných zpráv zřejmá z vertikálního uspořádání celého diagramu.
- V diagramech spolupráce je následnost zobrazena číslem, které vyjadřuje pořadí vykonávání.

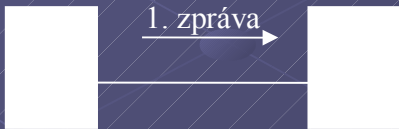
# Způsoby popisu grafických prvků diagramu spolupráce ve vývojových prostředích:

Objekt: Třída



- Objekt – má své jméno, můžeme uvést i jeho třídu

- Actor (terminator) – zdroj dat a požadavků, může být zobrazen také jako objekt

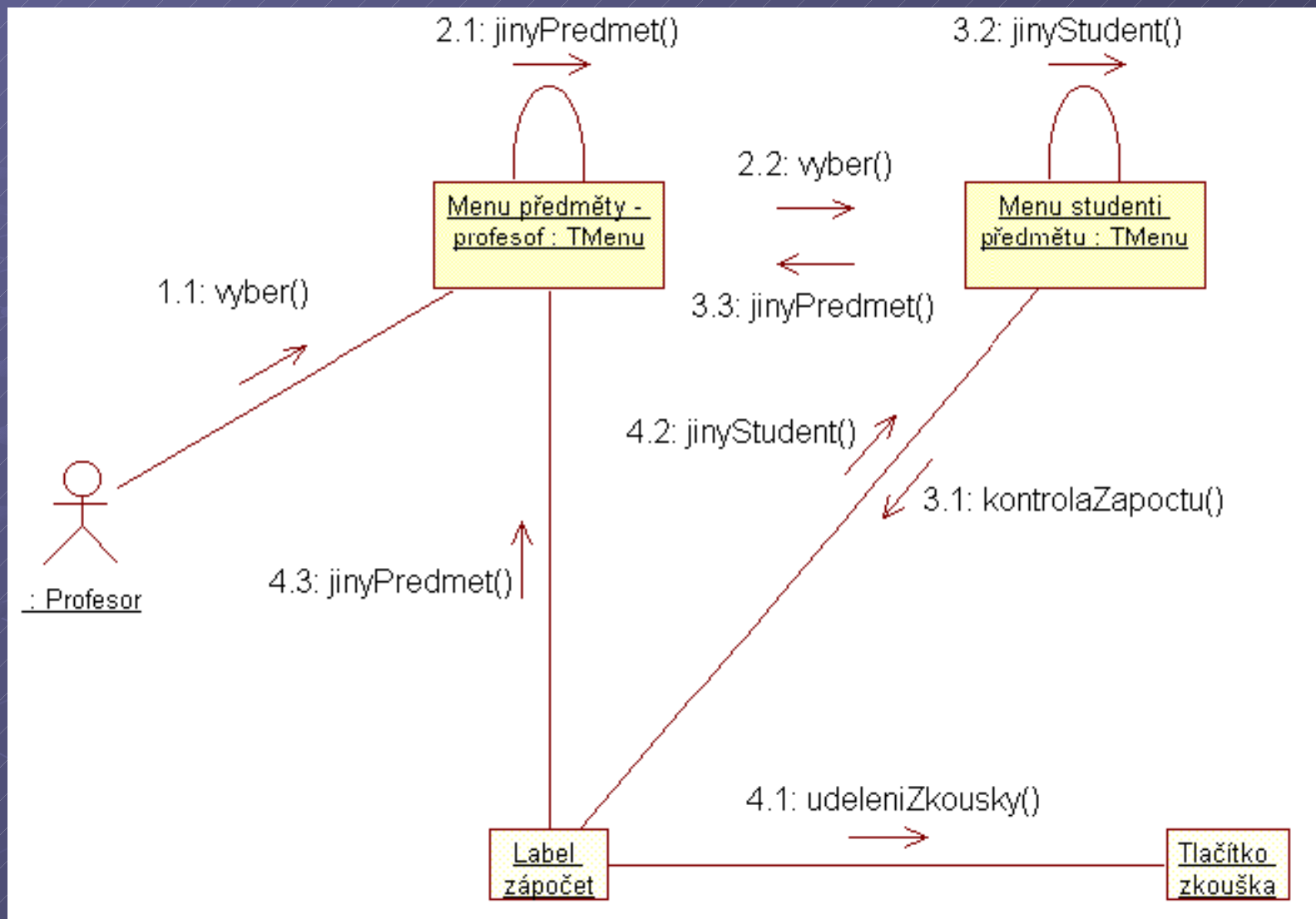


- Tok zpráv nebo dat se znázorněním orientace a druhu zprávy, každá zpráva má své pořadové číslo

Způsob zobrazení prvků diagramu závisí na vývojovém prostředí, ve kterém diagram vytváříme. V nenáročných případech vystačíme se slovním popisem objektů a druhů zpráv.

Více úrovní zpráv můžeme rozlišit pomocí teček tak, jak je označeno v následujícím příkladu, který zobrazuje část školního informačního systému pro správu informací o zkouškách a zápočtech.





# Seznam zpráv:

- 1.1: vyber() – výběr předmětu, který daný profesor učí
- 2.1: jinyPredmet() – výběr jiného předmětu
- 2.2: vyber() – výběr studena
- 3.1: kontrolaZapoctu() – provede se kontrola, zda student z příslušného předmětu má udělený zápočet
- 3.2: jinyStudent() – výběr jiného studenta
- 4.1: udeleniZkousky() – zapsání úspěšně složené zkoušky
- atd.



Někdy je vhodné označit zprávu jako funkci s uvedením typu dat, která jsou jejím výsledkem. Syntaxe je závislá na implementaci, zápis může vypadat například takto:

```
Dolt(a,b,c): status  
status:=Dolt(a,b,c)
```

Počet opakování množiny stejných zpráv můžeme vyjádřit pomocí hvězdičky s obvyklou syntaxí:

```
*(j=1..n)
```

```
*(1..n)
```

Samotná \* značí neznámý počet opakování.

Zaslání zprávy může být omezeno splněním podmínky, kterou je možno zapsat matematickým výrazem nebo slovně, například:

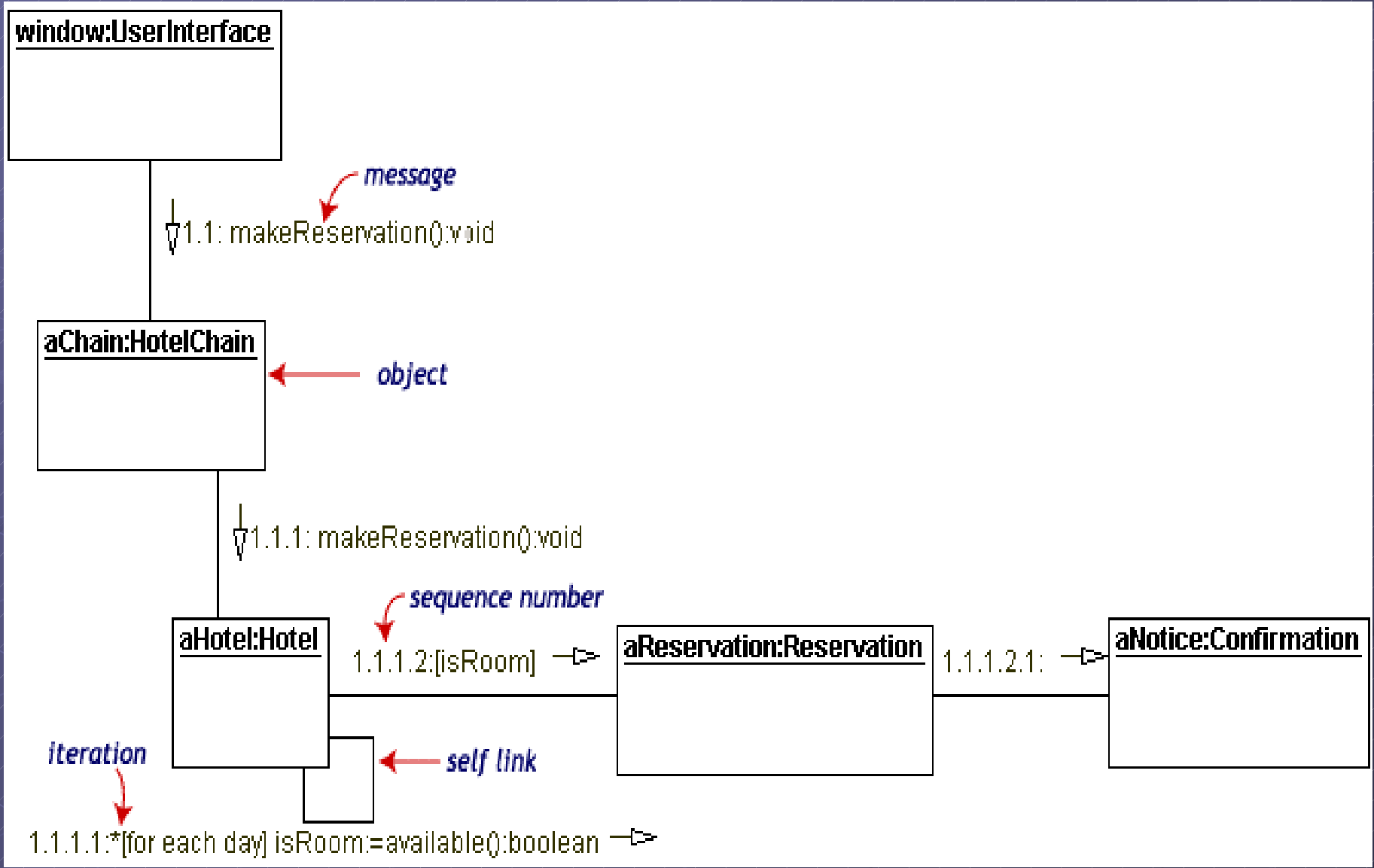
`Dolt[x>0]`

`DoltAgain[isOk()]`

V případě příjmu zpráv od více objektů, je nutné specifikovat případnou závislost na předchozích zprávách, například:

`[B6,A3]C13`

Znamená to, že zpráva C13 se zašle až po příchodu zpráv B6 a A3



# Pro jaké účely je vhodné diagram spolupráce použít:

- Popis spolupráce mezi objekty
- Pro definici datových a komunikačních vazeb mezi objekty
- Zpřehlednění struktury spolupracujících objektů