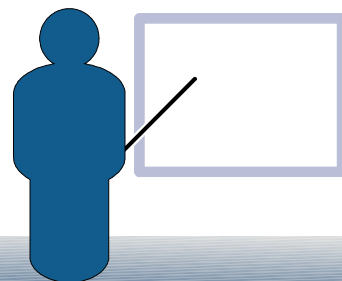


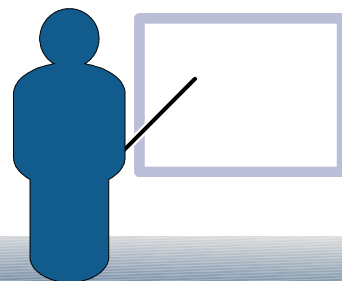
# Unified Modeling Language (UML)

Martin Matula  
martin.matula@sun.com



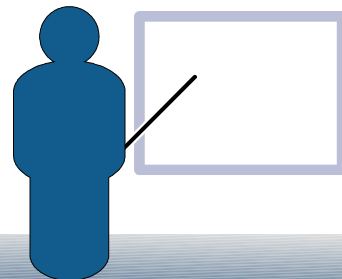
# Osnova

- Úvod (Co je UML?)
- Jednotlivé typy diagramů
- Minulost, současnost a budoucnost
- Nástroje
- Literatura



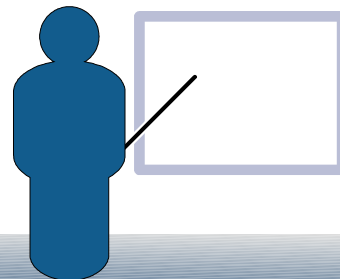
# Co je UML?

- Objektově orientovaný modelovací jazyk (nikoliv metoda)
- Skládá se z popisu notace (grafické syntaxe) a z metamodelu (popisu abstraktní syntaxe)
- Vytvořen sjednocením metod Booche, Rumbaugh a Jacobsona
- 1997 – přijat jako standard OMG (Object Management Group)



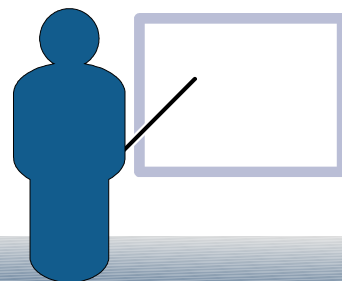
# Typy diagramů

- Diagram případů užití
- Diagram tříd
- Stavový diagram
- Interakční diagramy
  - Sekvenční diagram
  - Kolaborační diagram
- Diagram aktivit
- Fyzické diagramy
  - Diagram nasazení
  - Diagram komponent



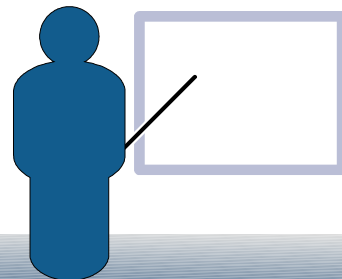
# Případ užití (use case)

- Zavedl Jacobson
- Souhrn alternativních scénářů/toků při snaze uživatele o dosažení nějakého cíle
- Scénář (scenario)/flow (tok) je sekvence kroků popisujících interakci mezi uživatelem a systémem.



# Případ užití - příklad

- Scénář pro „nákup produktu“:  
*Zákazník si prohlédne katalog a přidá požadované zboží do nákupního košíku. Když si zákazník přeje zaplatit, vloží dodací a platební informace (číslo kreditní karty) a potvrdí nákup. Systém ověří správnost platebních informací (zablokuje danou částku na platební kartě) a potvrdí prodej zobrazením potvrzující hlášky a zasláním kopie potvrzení e-mailem.*
- Popisuje pouze jednu z variant – tu nejočekávanější
- Může mít hodně variant



# Případ užití – příklad (2)

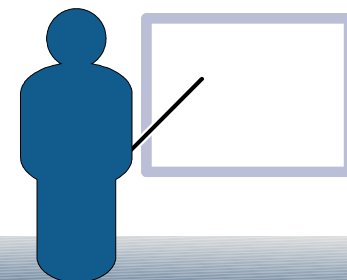
## Nákup produktu

### Hlavní scénář:

1. Zákazník prohlíží katalog a vybere si zboží k nákupu
2. Zákazník zvolí nákup
3. Zákazník vyplní dodací informace (adresa, expresní nebo standardní dodávka)
4. Systém zobrazí plnou cenu včetně ceny dodání
5. Zákazník vyplní platební informace (číslo kreditní karty)
6. Systém autorizuje platbu
7. Systém potvrdí prodej
8. Systém zašle potvrzovací e-mail zákazníkovi

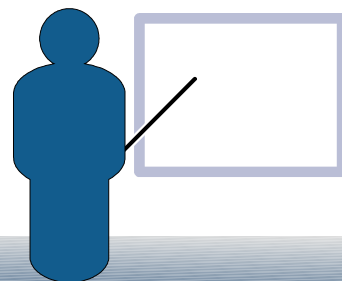
### Alternativy:

- 3a. Uživatel je pravidelným zákazníkem
  - 3a1. Systém zobrazí naposled zapamatované dodací a platební informace
  - 3a2. Uživatel může potvrdit, nebo změnit zobrazené informace a scénář pokračuje v kroku 6
- 6a. Systému se nepovedlo autorizovat platbu
  - 6a1. Zákazník může opravit platební informace, nebo zrušit nákup



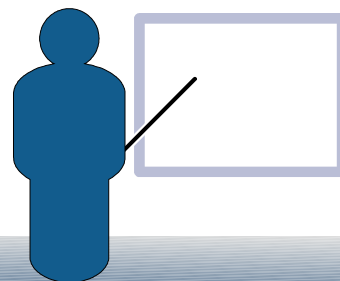
# Případ užití – další informace

- Popis případu užití může dále obsahovat:
  - Stručný popis případu užití
  - Rozdělení na alternativní a chybové toky
  - Úvodní předpoklady (pre-conditions)
  - ...



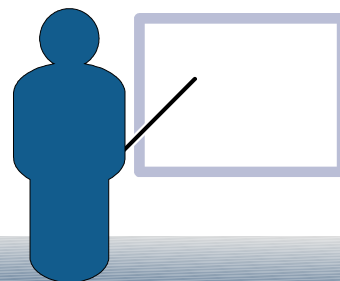
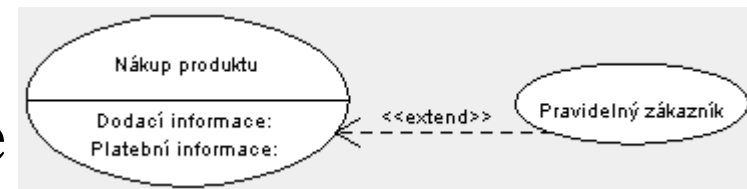
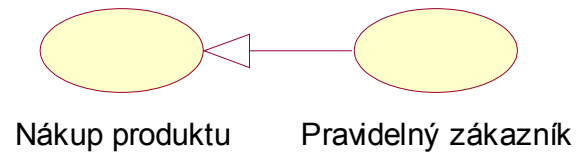
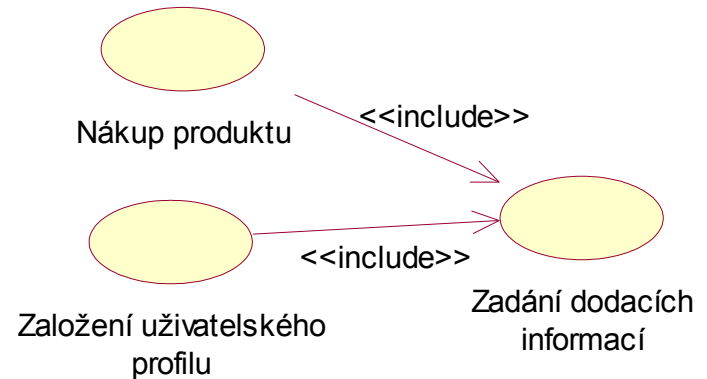
# Diagram případů užití

- Actor – role, ve které se uživatel nachází při „provádění“ případu užití
  - Nemusí nutně reprezentovat živou bytost (může se např. jednat o externí systém)
  - Jeden uživatel může vystupovat v několika rolích



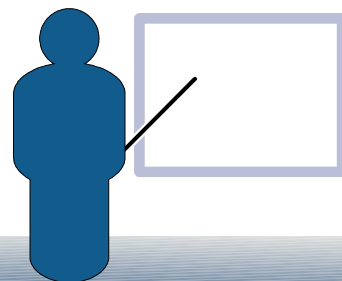
# Vztahy mezi případy užití

- Obsažení (include)
  - Více případů užití může jako svou součást obsahovat jiný případ užití
- Dědičnost (generalization)
  - Když jeden případ užití pozměňuje/rozšiřuje scénáře v jiném případě užití
- Rozšíření (extend)
  - Podobný jako dědičnost, ale může rozšiřovat pouze „extension points“ rozšiřovaného případu užití

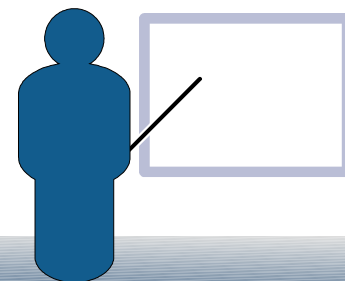
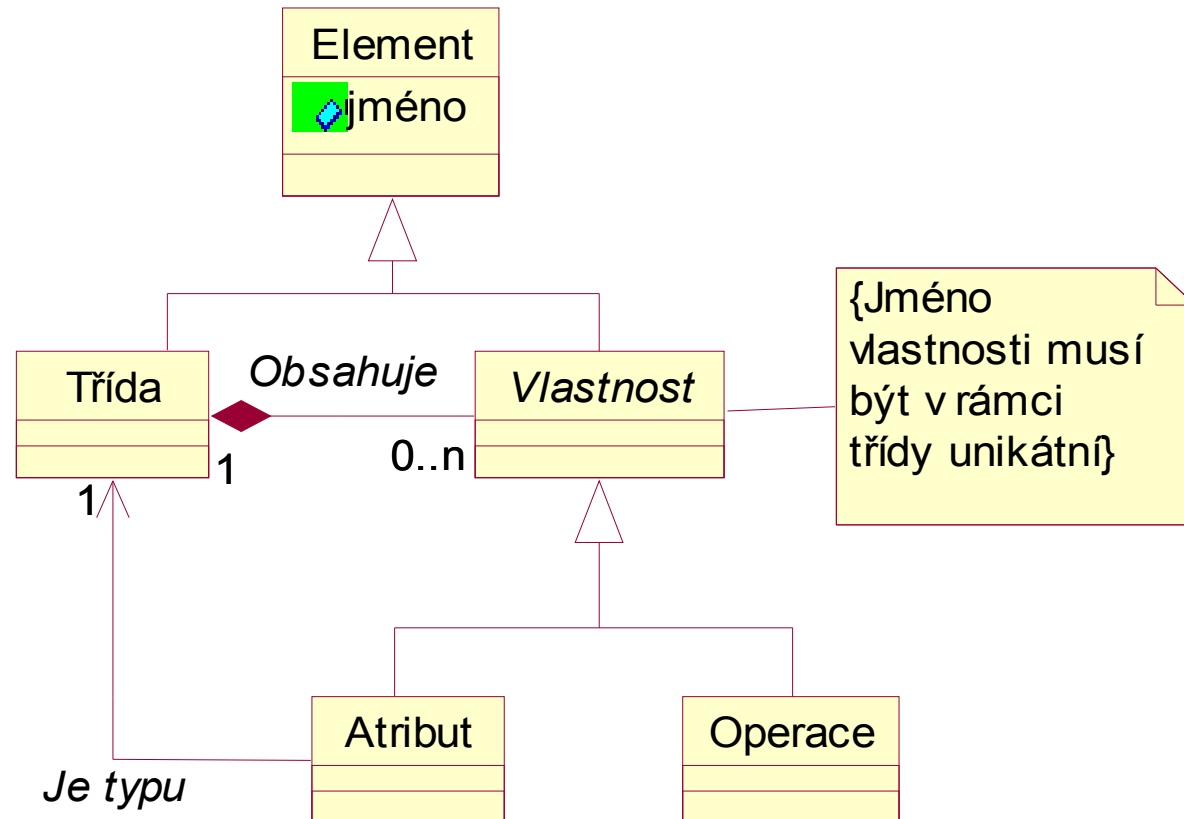


# Diagramy tříd

- Popisují typy objektů v systému a jejich vzájemné vztahy
- Dva druhy vazeb/vztahů
  - Asociace
  - Generalizace
- Třídy mohou mít atributy, operace a omezení (constraints)

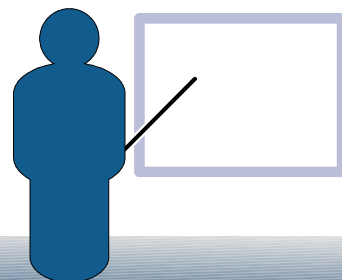


# Diagram tříd - příklad



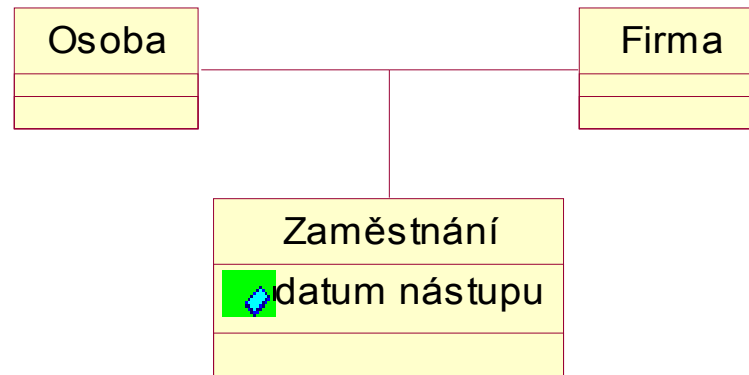
# Diagram tříd - pojmy

- Třída
  - Viditelnost, abstrakce
  - Atribut
    - *viditelnost jméno : typ = hodnota*
    - Viditelnost - *+public, -private, #protected*
  - Operace
    - *viditelnost jméno(seznam-parametrů):návrátový-typ*
    - Seznam parametrů – *směr jméno : typ = hodnota*
      - Směr – *in (default), out, inout*
- Asociace
  - Multiplicita konců, agregace, navigabilita
- Omezení

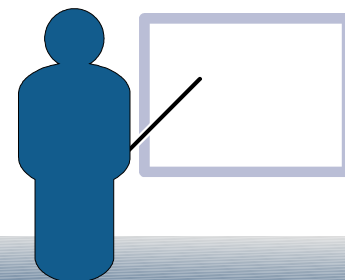
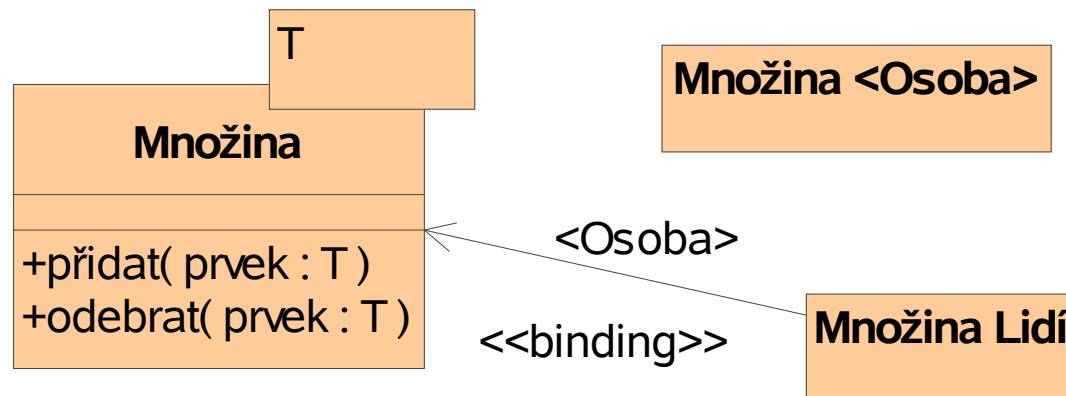


# Diagramy tříd – pokročilejší koncepty

- Asociační třídy

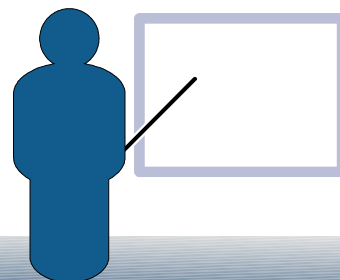
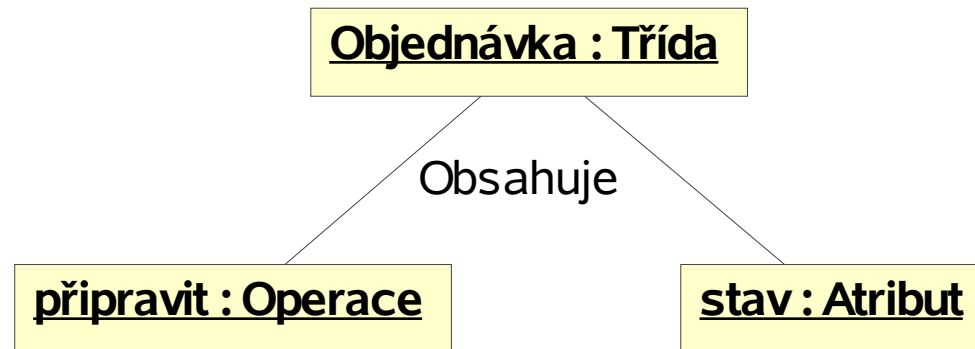


- Parametrizovatelné třídy (šablony)



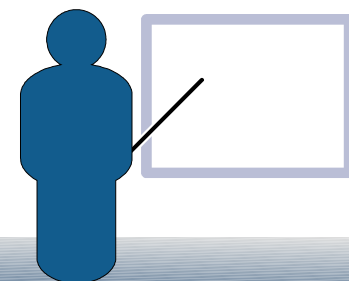
# Diagram objektů

- Zachycuje objekty v systému v daném časovém okamžiku

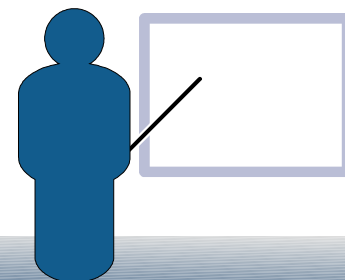
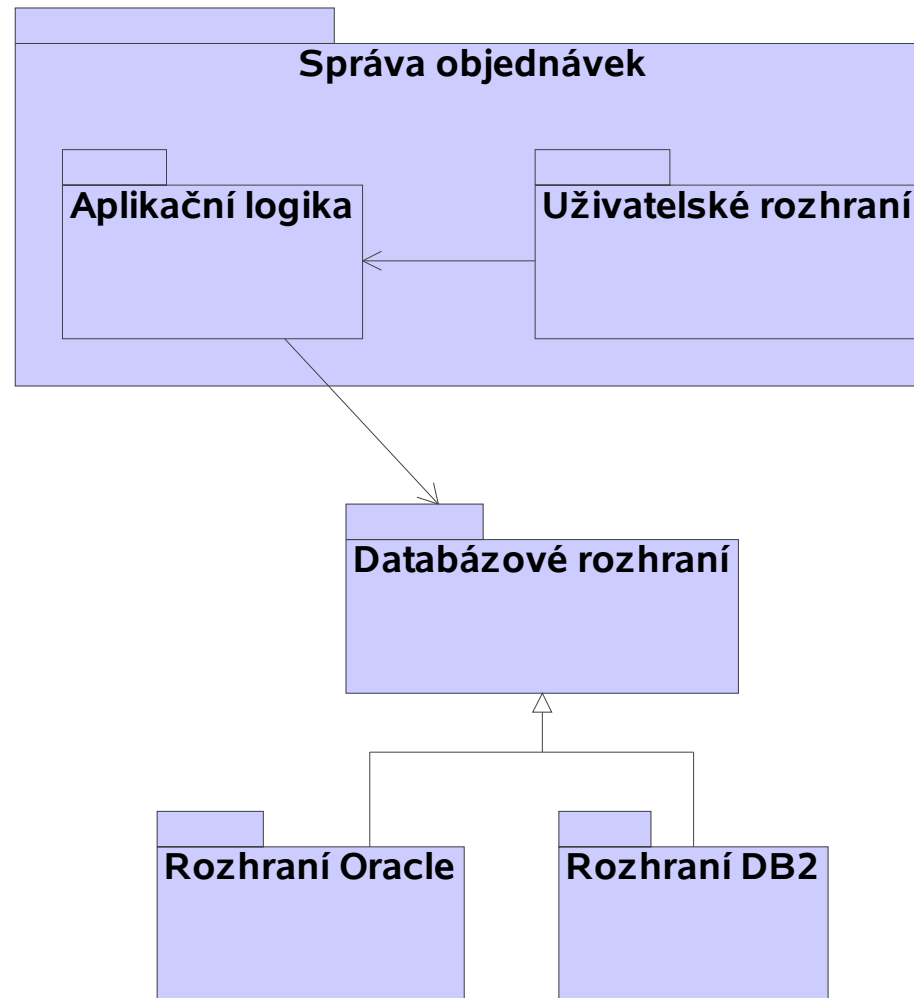


# Diagram balíků

- Diagram tříd, který obsahuje pouze balíky a závislosti mezi nimi
- Balík – umožňuje modelované elementy seskupovat do větších logických celků
- Konstrukce:
  - Závislost
  - Dědičnost
  - Vnoření

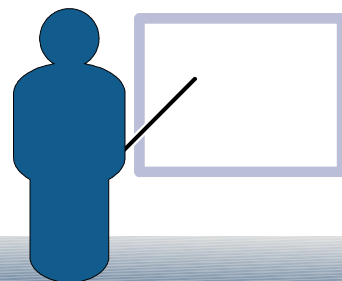


# Diagram balíků - příklad

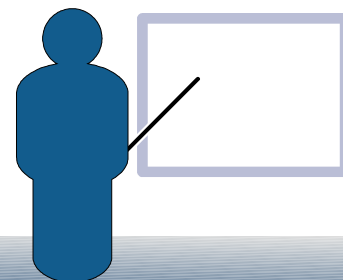
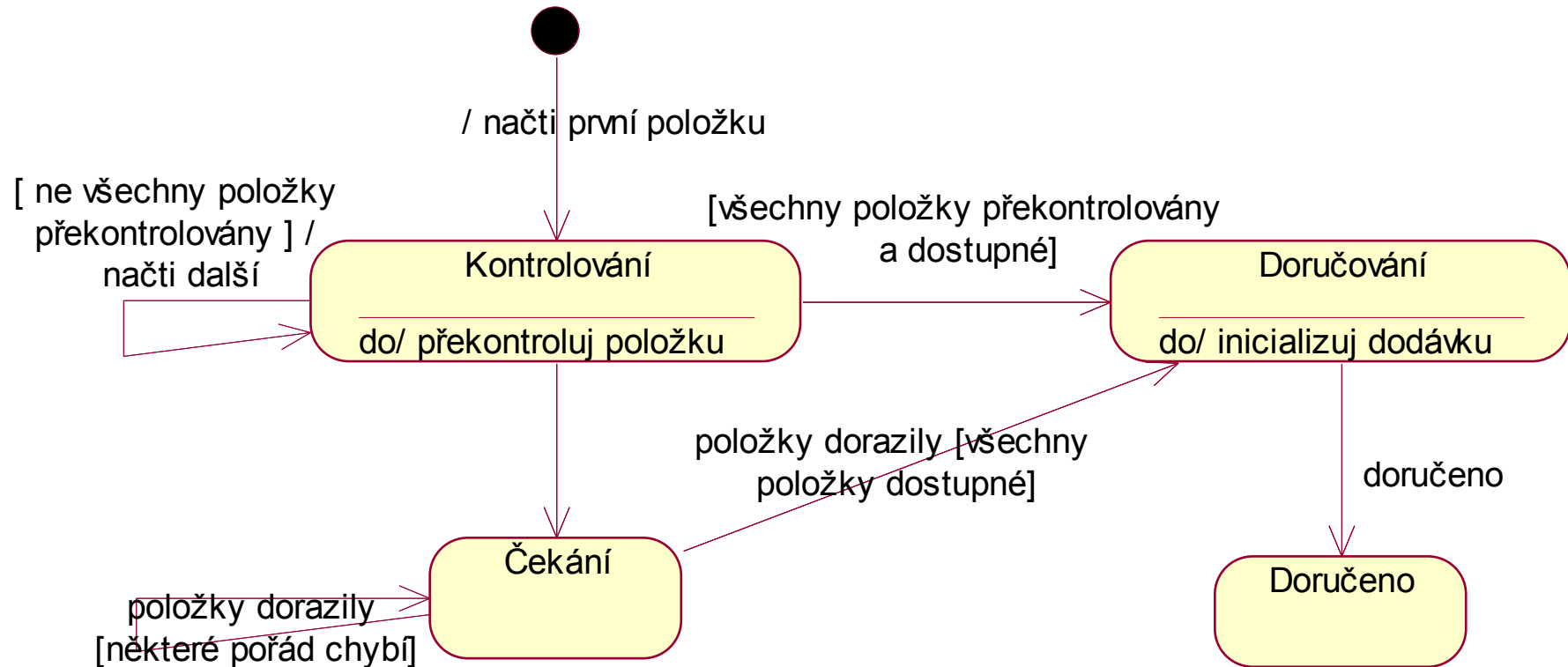


# Stavový diagram

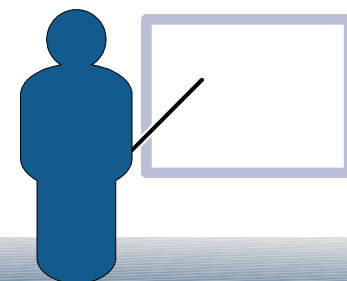
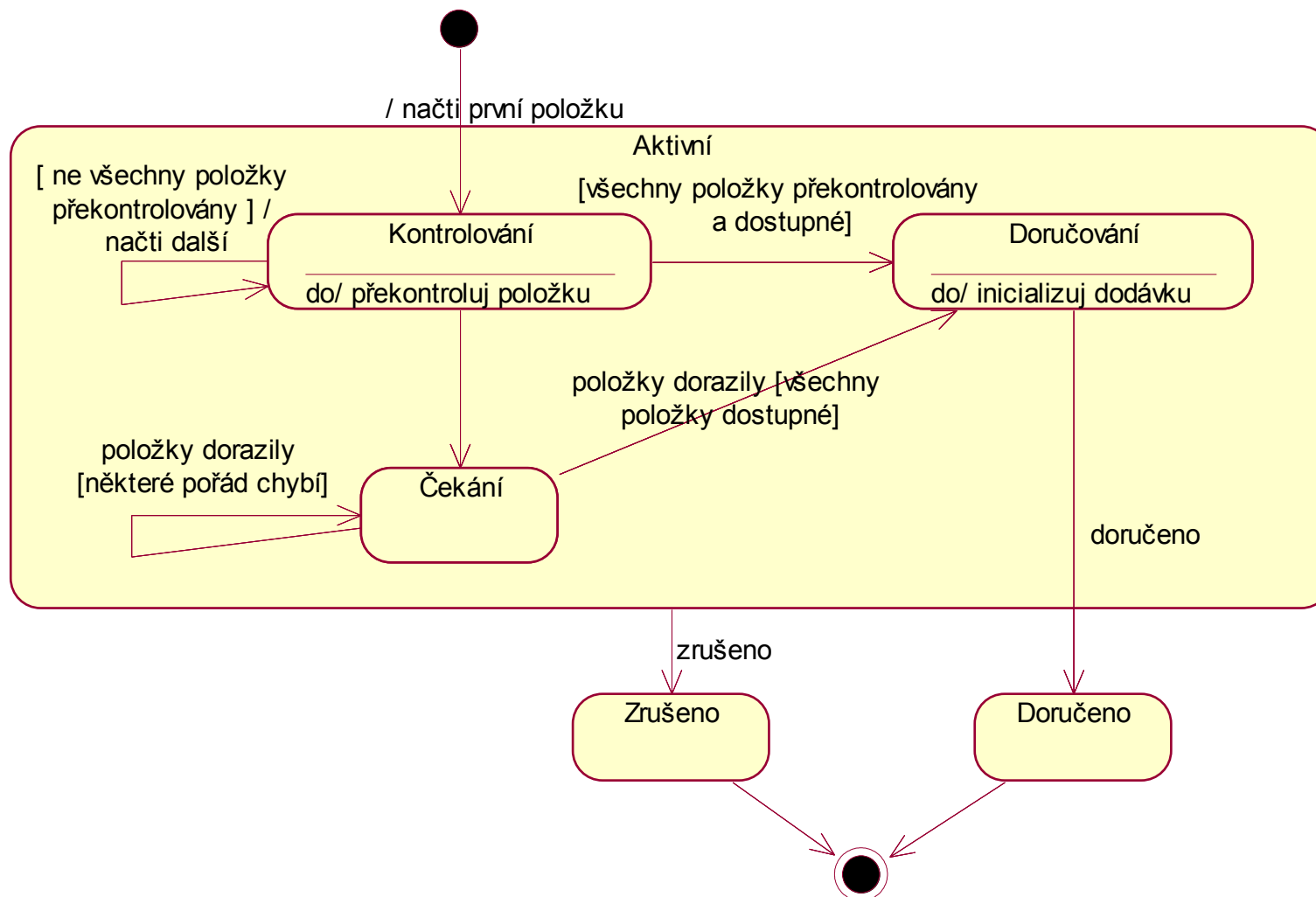
- Popisuje všechny stavy daného objektu a přípustné přechody mezi nimi
- Pojmy:
  - Událost [podmínka]/Akce
  - Aktivita (entry/do/exit/událost)
  - Stav
  - Přechod
  - Počáteční/koncový stav
- Paralelní/konkurenční stavy



# Stavový diagram - příklad

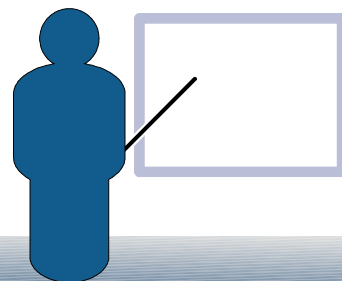


# Složený stav - příklad



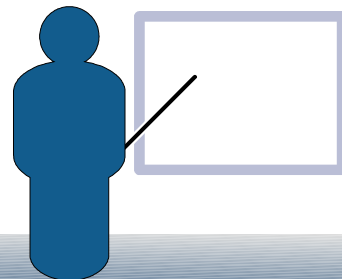
# Interakční diagramy

- Popisují, jak mezi sebou komunikují/spolupracují skupiny objektů
- Typicky interakční diagram zachycuje chování jednoho případu užití
- Dva druhy
  - Sekvenční diagramy
  - Kolaborační diagramy

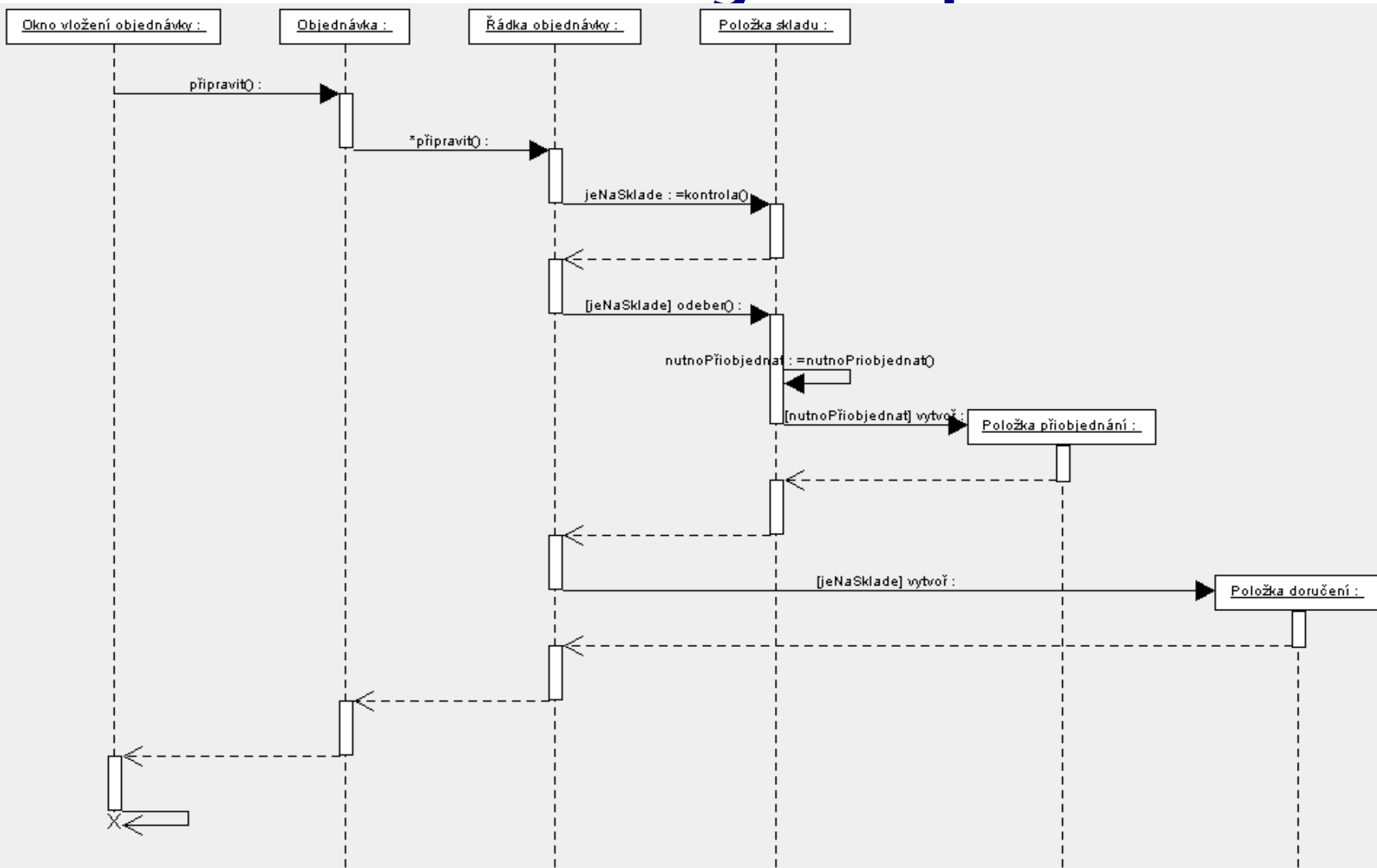


# Příklad popisované funkcionality

- Okno „Vložení objednávky“ pošle zprávu „připravit“ objektu typu „Objednávka“
- „Objednávka“ poté pošle zprávu „připravit“ každé „Řádce objednávky“
- Každá „Řádka objednávky“ překontroluje danou „Položku skladu“
  - Pokud kontrola vrací hodnotu „true“, „Řádka objednávky“ odečte příslušné množství „Položky skladu“ ze skladu
  - Pokud „Položka skladu“ dosáhla limit pro přioobjednání, vyžádá přioobjednání

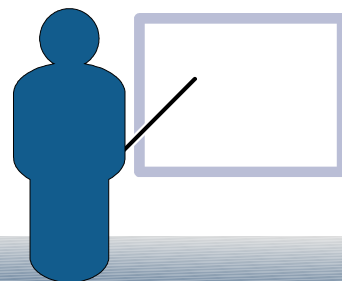


# Sekvenční diagram - příklad

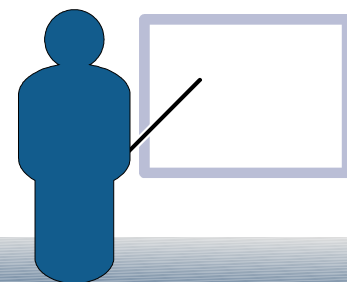
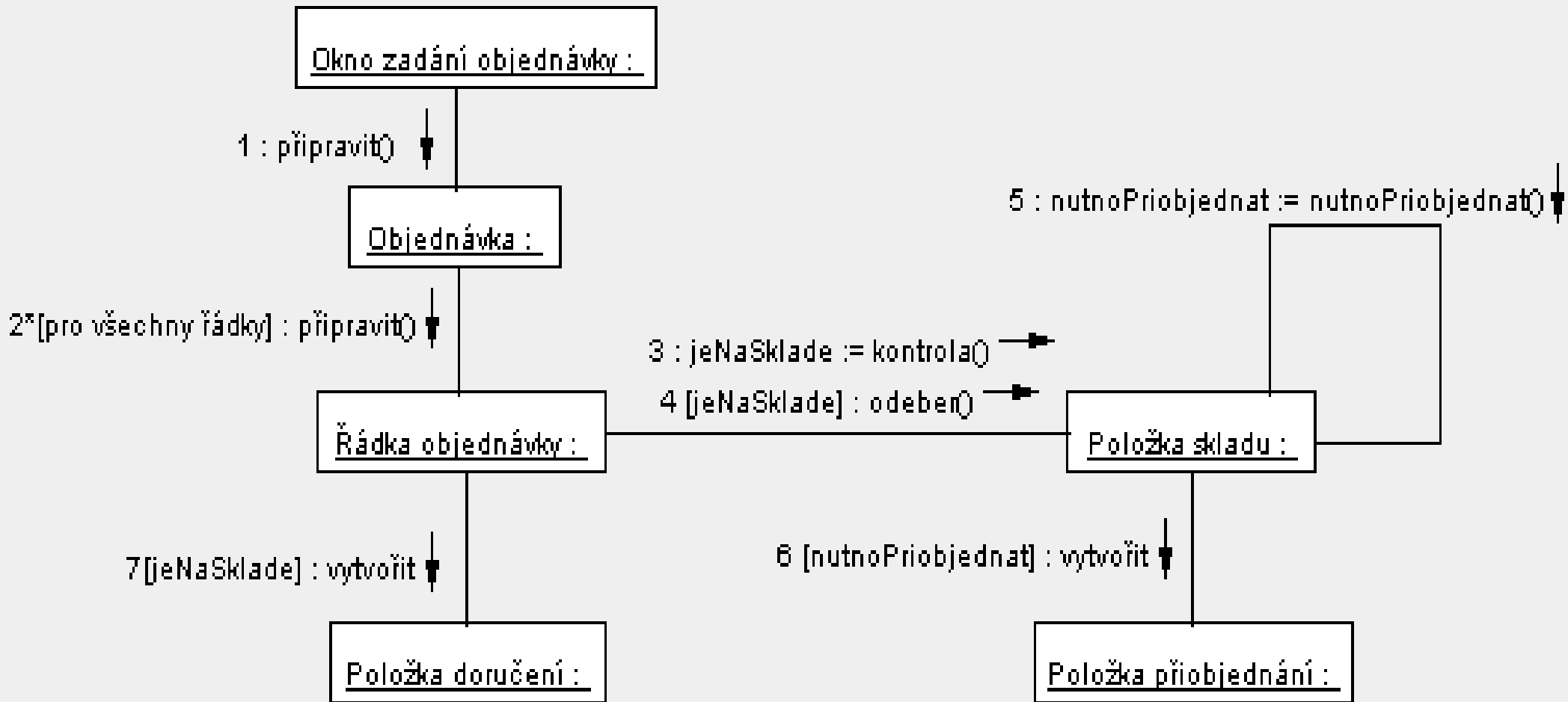


# Sekvenční diagram - pojmy

- Objekty
- Lifeline (Jacobson) – „životní čára“ objektu
- Zprávy (messages)
- Activation box
- Podmínka (condition)
- Iterační značka (iteration marker)
  - \*[pro všechny řádky objednávky]
- Návrat
- Asynchronní zpráva
- Vytvoření/smazání objektu

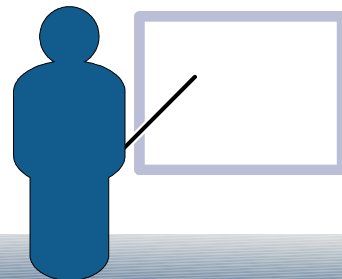


# Kolaborační diagram - příklad

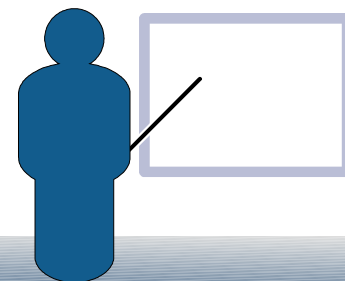
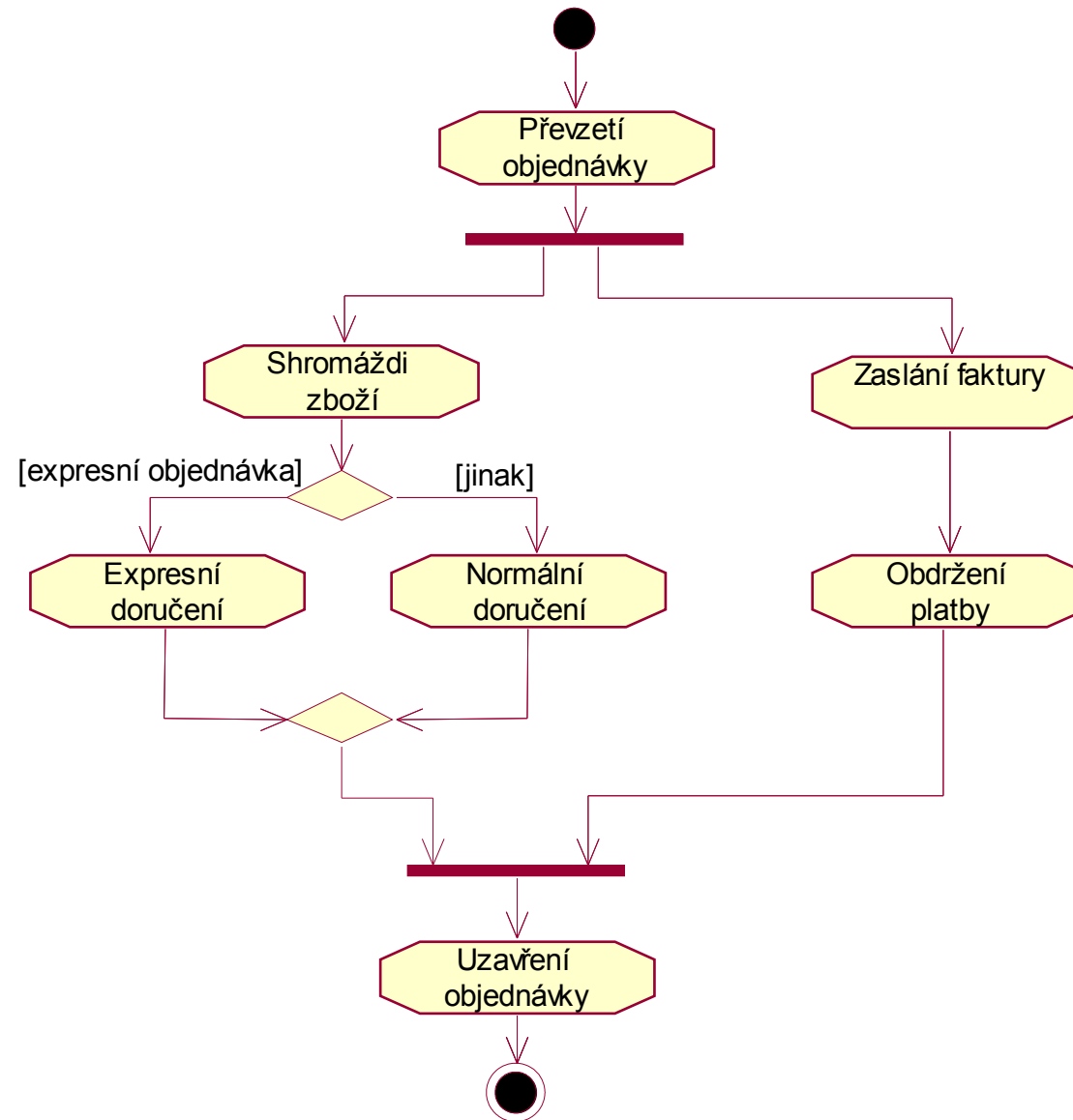


# Diagram aktivit

- Nepochází od „three amigos“
- Kombinuje událostní diagramy (Jim Odell), SDL techniky pro modelování stavů, modelování workflow a Petriho sítě
- Užitečné k popisu chování, které obsahuje hodně paralelního zpracování
- Obsahuje aktivitní stavy (activity states)/aktivity
  - Stav dělání něčeho – vykonávání softwarové rutiny, psaní znaku, atd.
- Popisuje seřazení aktivit s podporou jak podmíněného, tak paralelního chování

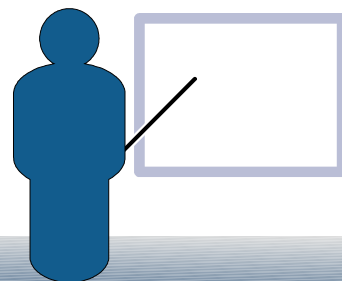


# Diagram aktivit - příklad



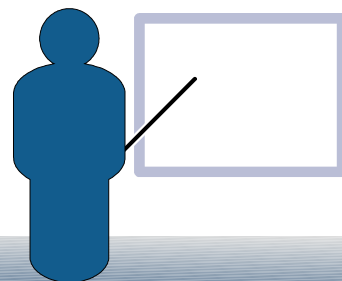
# Diagram aktivit - pojmy

- Počáteční/koncový stav
- Aktivita
- Vidlička (fork) / sjednocení (join)
- Větvení (branch) / spojení (merge)
- Podmíněná vlákna
- Složené aktivity
- Plavecké dráhy (swimlanes)



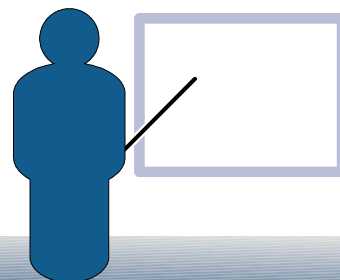
# Fyzické diagramy

- Dva druhy
- Diagram nasazení (deployment diagram)
  - Zobrazuje fyzické vztahy mezi SW a HW komponentami v dodaném systému
  - Skládá se z uzlů (výpočetních jednotek) a komponent, které jsou na daných uzlech nasazeny (instalovány)
- Diagram komponent (component diagram)
  - Zobrazuje jednotlivé komponenty v systému a jejich vzájemné vztahy
  - Komponenta reprezentuje fyzický softwarový modul



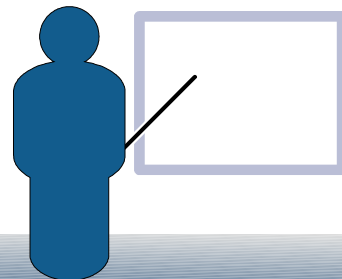
# Uživatelská rozšíření - Profily

- Stereotypy – mechanismus rozšíření metamodelu UML o nové metatřídy
- Např. „Tabulka“ pro modelování relačních tabulek



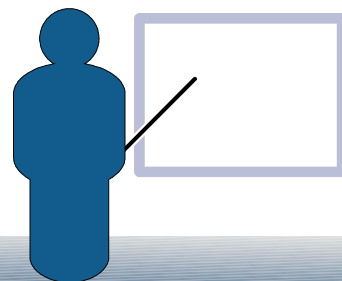
# Minulost, současnost, budoucnost

- UML 1.3 -> 1.4
  - Formální definice profilu
  - Nový element „Artifact“ (fyzická reprezentace komponenty – např. všechny kopie dané komponenty na disku)
  - „package“ viditelnost (~)
- 1.4 -> 1.5
  - Action semantics language
- 2.0
  - Kompletní přepracování metamodelu
  - Rozdělení na několik úrovní



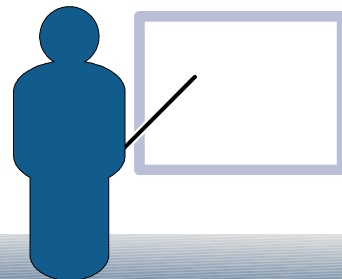
# Nástroje

- MagicDraw ([www.magicdraw.com](http://www.magicdraw.com))
- Poseidon for UML ([www.gentleware.com](http://www.gentleware.com))
- ArgoUML ([www.argouml.org](http://www.argouml.org))
- Rational Rose ([www.rational.com](http://www.rational.com))



# Literatura

- Fowler, M.: UML Distilled 2<sup>nd</sup> Edition, Addison-Wesley, 2000
- Booch, Rumbaugh, Jacobson: UML User Guide, Addison-Wesley, 1999
- Booch, Rumbaugh, Jacobson: UML Reference Manual, Addison-Wesley, 1999
- Alstair Cockburn: Writing Effective Use Cases, Addison-Wesley, 2001
- Specifikace UML ([www.omg.org](http://www.omg.org))



# Dotazy

