

# UML – úvod

---

Kapitola má seznámit se základy modelovacího jazyka UML.

**Klíčové pojmy:** UML, CASE nástroje, procesní modelování, případy užití, role, diagram tříd, diagram objektů, sekvencní diagramy, diagram stavů, diagram aktivit

**Zdroje:**

- Kanisová Hana, Müller Miroslav: UML srozumitelně, Computer Press 2007

## 1 Co to je UML

- Unified Modeling Language
- jazyk pro popis objektově orientované analýzy a návrhu aplikací
- slouží k vzájemné komunikaci mezi zadavatelem a návrhářem systému
- má několik částí, druhů, není nutné použít všechny části, ani je použít v určitém pořadí

### 1.1 CASE nástroje

- nástroje pro podporu analýzy a návrh aplikací
- ideální pro návrh IS
- vycházejí z jazyka UML (např. Power Designer firmy Sybase)

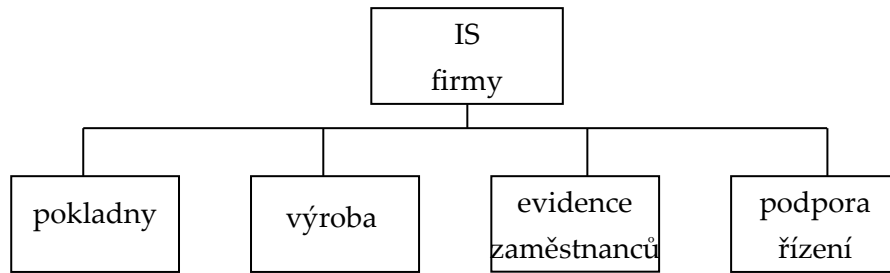
### 1.2 Příklad

Jednotlivé části si ukážeme na příkladu: Zákazník donese do sběrný elektrospotřebič, ta ho přijme, vyřídí formality, zašle do patřičné opravny a pak zase předá zpět zákazníkovi. Před tím, než navrhne databázi na uchování patřičných dat a obslužné programy, budeme se snažit celý problém rozebrat, abychom pak při tvorbě aplikací na nic nezapomněli. Postupně můžeme použít tyto nástroje (některé z nich nejsou přímo součástí jazyka UML):

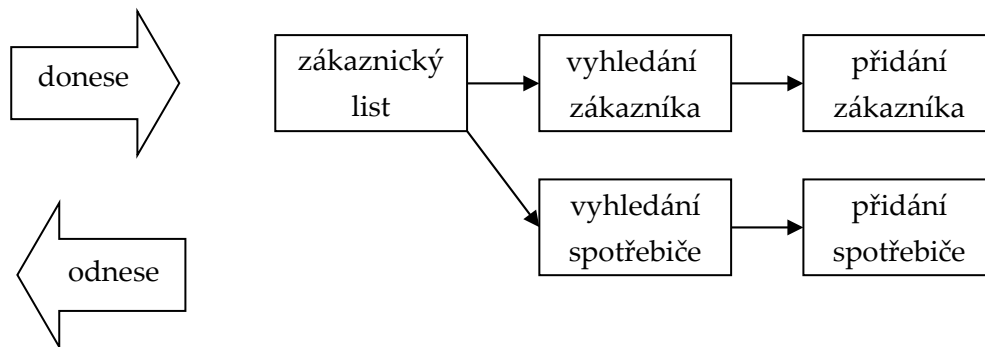
## 2 Modelování firemních procesů = PROCESNÍ MODELOVÁNÍ

- **firemní proces** = sekvence činností vytvářejících výsledný produkt (smysl činnosti organizace - výroba, obchodní činnost, ...)
- bývá úvodním krokem zahájení analýzy problému
- není součástí UML

## 2.1 Diagram hierarchie procesů



## 2.2 Diagram procesních vláken



## 3 PŘÍPADY UŽITÍ

- navrženo 1992 Iivarem Jacobsonem, dnes základním nástrojem
- popisují **typické interakce uživatelů se systémy**, zachycují požadavky na systém
- to, co má budoucí systém dělat (ne jak!)
- každý případ užití popisuje **jeden ze způsobů** použití systému
- **aktér** = role, ve které vystupuje uživatel v rámci komunikace se systémem
  - někdo, kdo si vyměňuje informace se systémem, externí objekt
  - jeden fyzický uživatel může vystupovat ve více rolích
  - jeden případ užití může být prováděn více aktéry
- nejdříve definujeme aktéry, pak pro každého z nich případy užití
- snaží se zachytit všechnu komunikaci aktéra
- všechny akce v systému jsou vyvolány aktérem
- případy užití se kreslí elipsou

### 3.1 Základní prvky grafů případů užití

- **postavička** = aktér
- **obdélník** = systém
- **elipsa** = případ užití
- **plná čára** = spojnice = interakce mezi uživatelem a případem užití

- čárkovaná čára se šipkou = odkaz = interakce mezi jednotlivými případy užití

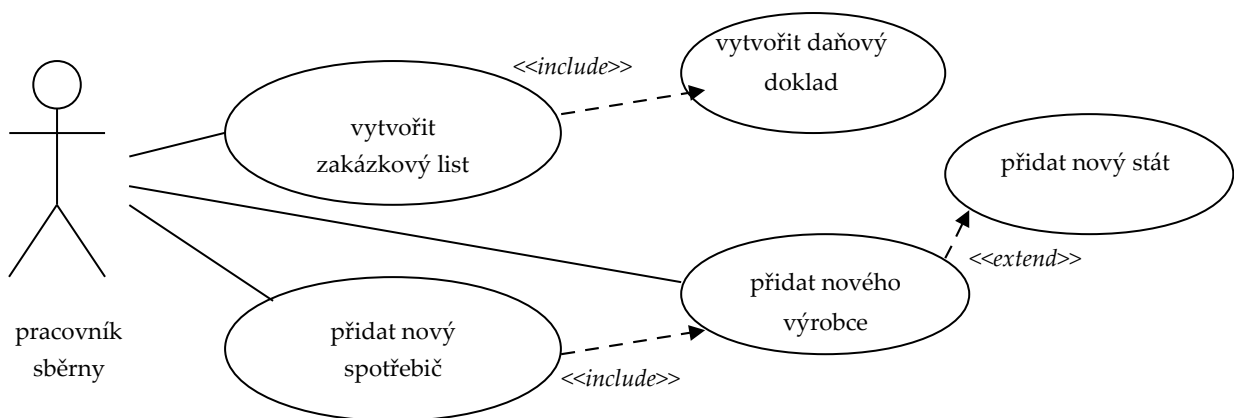
### 3.2 Vztahy mezi jednotlivými případy užití

- relace `<<include>>` = části scénáře se opakují ve více případech užití, např. vytvořit daňový doklad, vyhledat zakázku, zákazníka, ... ('procedura'), šipka směrem ke vloženému
- relace `<<extend>>` = doplňuje chování do základního případu užití, scénář je soběstačný, ale existuje možnost rozšíření (př. u případě Přidat nového výrobce může být možnost Přidat nový stát), šipka směrem od rozšiřujícího

### 3.3 Vztahy mezi aktéry

- zobecňování – dědičnost

### 3.4 Příklad



### 3.5 Případy užití jako sada scénářů

Scénář případů užití = sekvence kroků popisujících interakci mezi aktérem a systémem

**Příklad:** Přijmout spotřebič do opravy - popis případu užití

krok	role	akce
1	uživatel	spustí volbu <i>Založit zakázku/nová zakázka</i>
2	system	zobrazí formulář a zpřístupní údaje pro vstupy
3	uživatel	pořídí vstupní informace o zakázce
...		

- pojmenujeme podle akce
- číslovaná sekvence kroků
- rozšíření = alternativní scénáře (opět číslované), např. Nový zákazník, větvení pomocí "pokud-jinak-konec\_pokud", ... (pseudojazyk)
- můžeme přidat vstupní a výstupní podmínky

## 4 MODELOVÁNÍ TŘÍD A OBJEKTŮ

### 4.1 Třída a její znázornění

- jméno
- vlastnosti, atributy
- metody=operace, to, co může třída vykonávat

Příklad:

- jméno: Pračka

atributy:

- značka
- kapacita
- spotřeba vody
- metody:
  - vložitPradlo()
  - přidatPrašek()
  - prat()
  - vyjmoutPradlo()
  - zapnout()

Pračka
značka kapacita spotřeba vody příkon
vložitPradlo() přidatPrašek() prat() vyjmoutPradlo() zapnout()

### 4.2 Vlastnosti Objektivě orientovaného návrhu

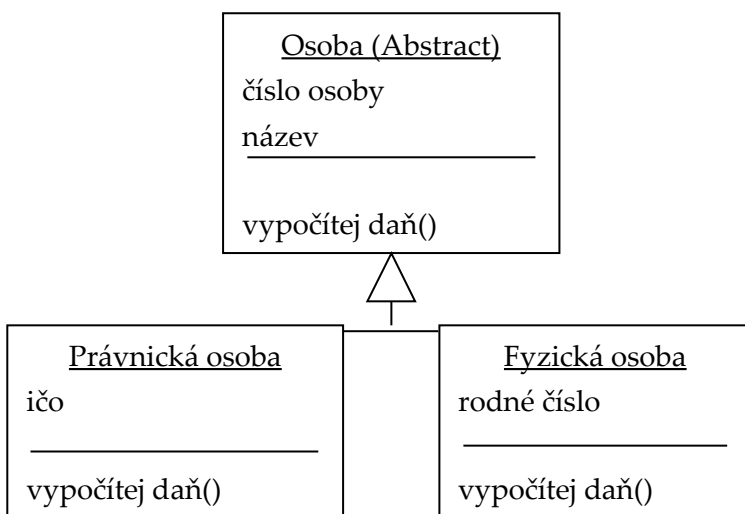
- **zapouzdření** – objekt skrývá informace před ostatními objekty, je potřeba vědět, ŽE to funguje, ne JAK to funguje (soukromý, chráněný, veřejný)
- **rozhraní** – tlačítka na ovladači, na pračce, ...
- **polymorfismus** – otevřít dveře, okno, krabici, ...
- **agregace** – složení objektu z jiných objektů
- **kompozice** – silný vztah mezi agregátem a komponentami, komponenty objektu mají smysl jen jako součást jiného objektu (košile – dírky, rukávy, límeček, ...), komponenty zanikají s celkem
- abstrakce
- objekty spolupracují - **zasílání zpráv** (ovladač-televize)

### 4.3 Asociace

- vztah mezi objekty – kreslí se jako čára
- asociace má **směr a jméno**
- asociace má směr, jednosměrná (zapínám televizi) i obousměrná (ženatý/vdaná)
- pro 2 objekty může být více asociací (spolupracovník, přítel | hrát za oddíl, být předsedou oddílu | učitel, výchovný poradce, ...)
- třída hraje v asociaci určitou **rolí** (hráč-zaměstnanec ... hraje za ... zaměstnavatel – mužstvo)
- **omezení asociace** – pravidlo, vztah nebo
- **asociační třída** – např. smlouva, asociace může mít **atributy a operace**, kreslí se čárkovaně k asociaci

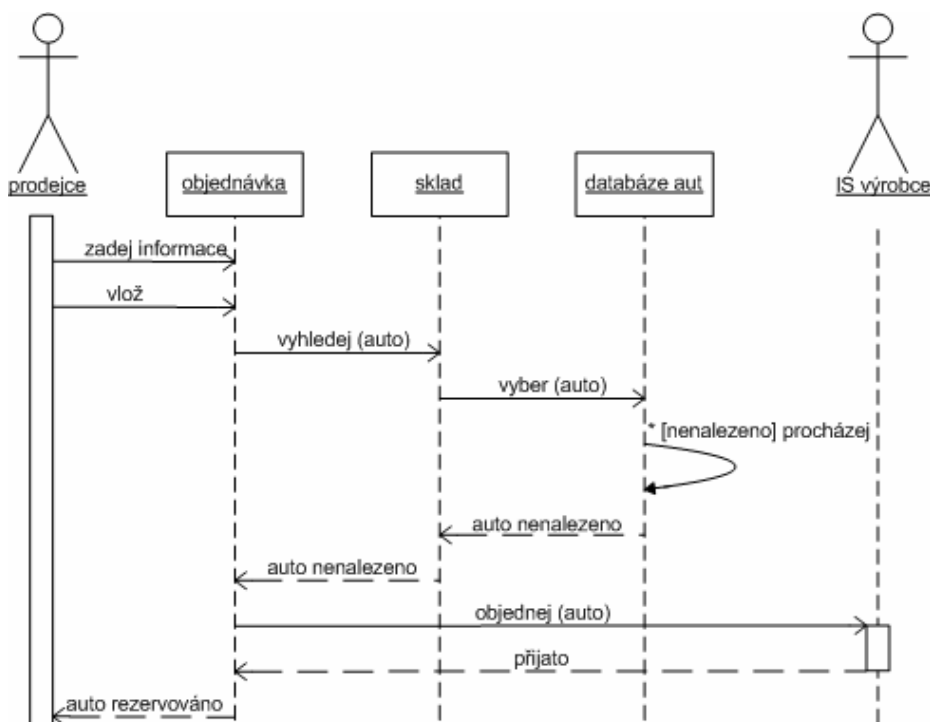
#### 4.4 Vazby mezi objekty, instancemi

- **instance** asociace, asociace mezi konkrétními **objekty** (instancemi třídy)
- instance tříd se podtrhávají
- násobnost asociace (př. od-do)
- příklad: dědičnost



#### 5 MODEL OBJEKTOVÉ SPOLUPRÁCE – Sekvenční diagramy

- interakce tříd navzájem
- svislá osa je osa času, čas roste směrem dolů

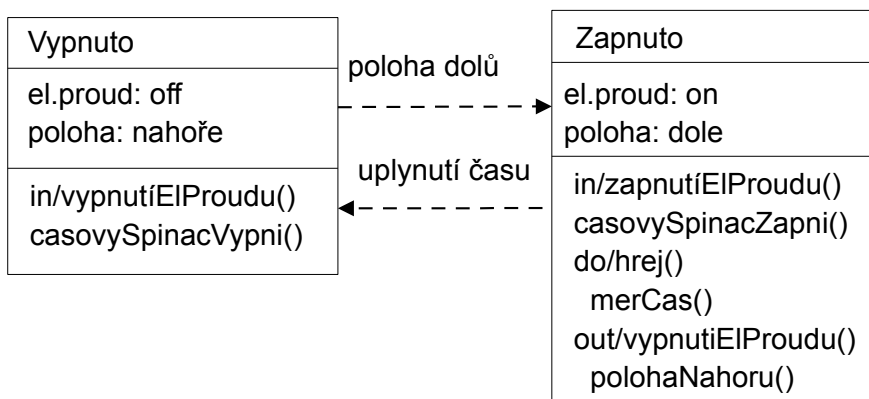


## 6 SESKUPENÍ TŘÍD

- seskupit třídy, které spolu souvisejí
- ohraničit společným obdélníkem (např. Zákazník, Zakázka, Montáž, ...)

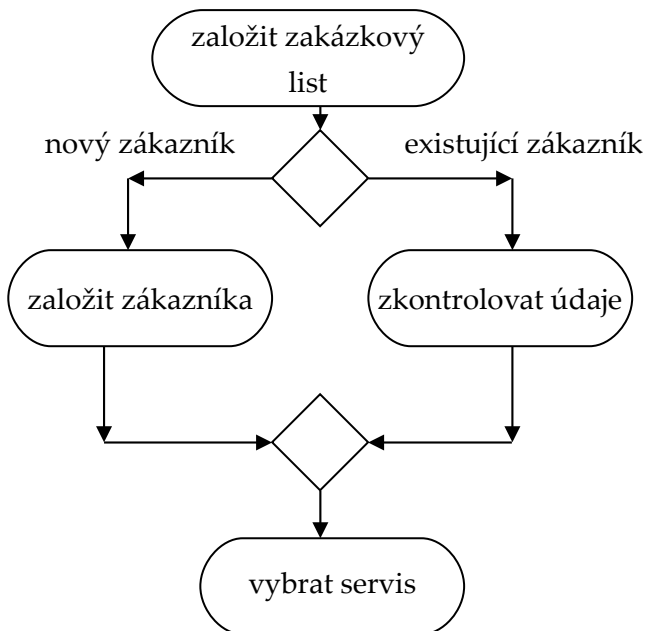
## 7 STAVOVÉ DIAGRAMY

- popisují stav jednoho objektu
- každý stav je popsán obdobně jako třída:
  - jménem
  - atributy
  - metodami
- objekt přechází mezi 2 stavy po splnění nějaké podmínky (zapnutí, vypnutí, kritická teplota, uplynutí určitého času, ...)
- metody jsou rozděleny na metody, které se provádějí:
  - když objekt vstupuje do daného stavu
  - když je objekt v daném stavu
  - když objekt daný stav opouští
- můžeme si ukázat na **příkladu topinkovače**, který přechází mezi 2 stavy: vypnuto a zapnuto
  - atributy: čas, el. proud on/off, poloha topinky nahoře/dole
  - metody: po změně polohy dolů začne topným tělesem protékat proud, protéká do té doby, dokud neuběhne nastavený čas, poté el. proud přestane protékat a poloha topinky se změní ze spodní na horní



## 8 DIAGRAM AKTIVIT

- obdoba stavových diagramů
- popisují sekvenční i paralelní chování



## 9 DATOVÉ MODELOVÁNÍ

- převod tříd na tabulky
- **logický** datový model (entita, atribut, vazba, klíče, ...)
- **fyzický** datový model (tabulka, řádek, sloupec, normalizace, ...)