

Abstrakce

- Abstrakce představuje nejdůležitější princip pro opětovnou použitelnost.
- Jedná se o nahrazení konkrétní datové entity, abstraktnější datovou entitou.
- Uvedená abstrakce nabývá většinou jedné ze dvou následujících forem:
 - generalizace
 - typování

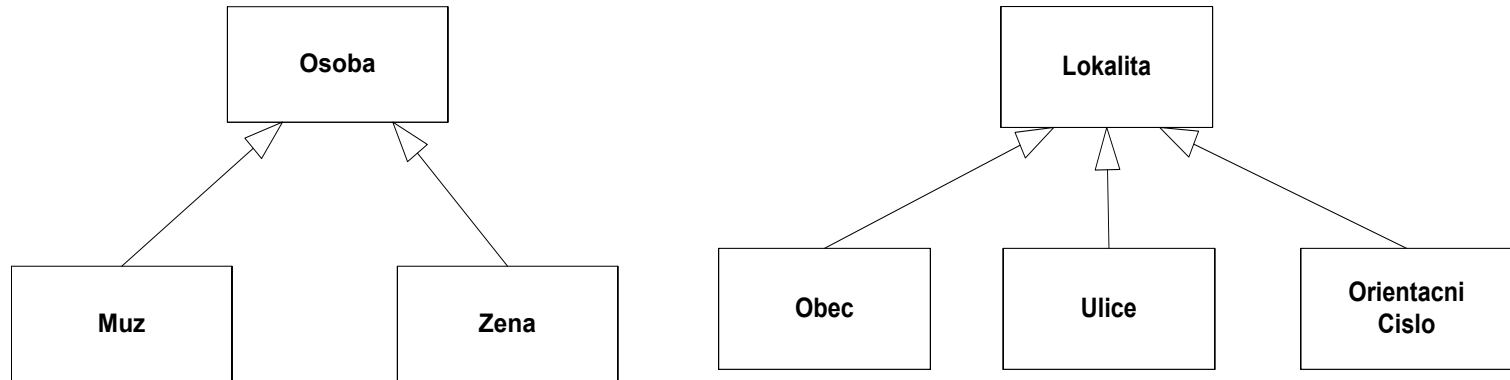
Generalizace

- Generalizací se rozumí zobecnění specializované datové entity do obecnější datové entity; mezi těmito entitami se stanoví relace **generalizace**.
- Specifické vlastnosti relace generalizace:
 - existuje jen mezi datovými entitami (třídami), nepřenáší se na instance,
 - způsobuje dědění datových atributů a relací z obecnější do specializovanější datové entity,
 - každá instance od obecnější entity dědí pouze **strukturu** datových atributů, **nikoli však obsah**,

Generalizace

- je to relace typu nebo – existuje-li k jedné obecnější datové entitě více specializovanějších entit, daná instance se vytvoří **právě k jedné z nich**.

Diagramy Osoba a Lokalita

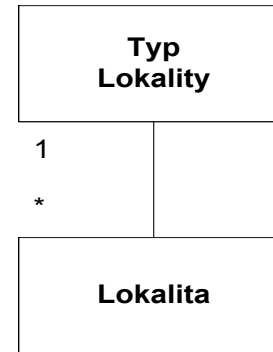
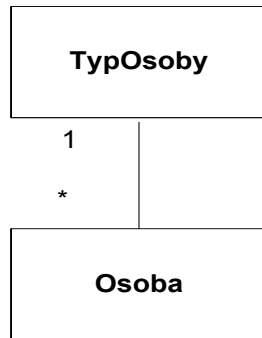


- Osoba atributy: jméno, rok narození
- Muž atributy: počet manželek (zdědí jméno, rok narození)
- Žena atributy: počet narozených dětí (zdědí jméno, rok narození)

Typování

- Při typování se místo obecnější datové entity definuje datová entita, která reprezentuje typ této datové entity.
- Tím, že se při typování obecnější datová entita specializuje ne datovou entitou jako při generalizaci, ale **instancemi typové datové entity**, vytvořená instance dědí **nejen strukturu datových atributů** obecnější datové entity, ale zároveň i **jejich obsah**.
- Uvedené datové entity jsou spojeny asociací.
- První příklad – typování osob, druhý příklad – typování lokalit.

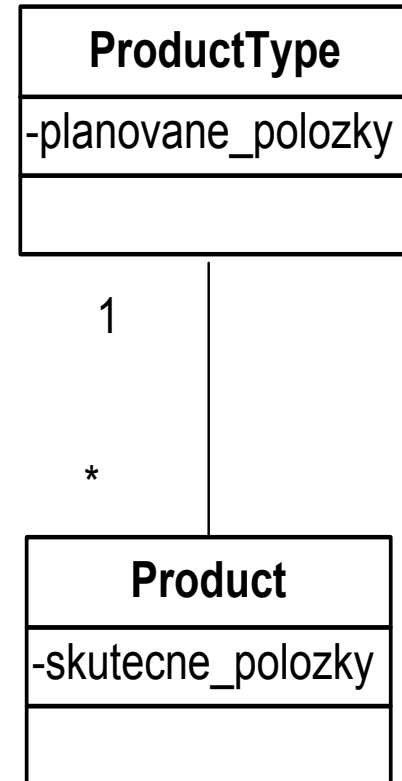
Typování osob a lokalit



- **TypOsoby** jsou instance **Muž** (jméno, rok narození, počet manželek) a **Žena** (jméno, rok narození, počet narozených dětí) a instance od třídy **Osoba** pak patří buď pod **Muže** nebo **Ženu**.
- Hodnoty datových atributů zůstávají stejné pro všechny specializované instance.

Objednávka / Produkt (lepší příklad)

- ProductType (Objednávka) obsahuje všechny plánované položky (plánovaná váha, plánované složení příměsí).
- Instance od **ProductType** pak představují různé objednávky lišící se různou hodnotou plánovaných položek.
- Instance od třídy **Product** jsou vždy svázány s konkrétní instancí třídy **ProductType** a liší se v hodnotách **skutečných položek** (skutečná váha, skutečné složení příměsí).



Typování & generalizace 1/2

- Typování je podobné generalizaci. V obou případech je využit princip **obecnější datové entity**.
- Při generalizaci se tato obecnější datová entita specializuje pomocí **konkrétnějších datových entit**.
- Při typování se obecnější datová entita specializuje **instancemi typové datové entity**.
- Při typování tak instance typové datové entity odpovídají konkrétnějším datovým entitám u generalizace.

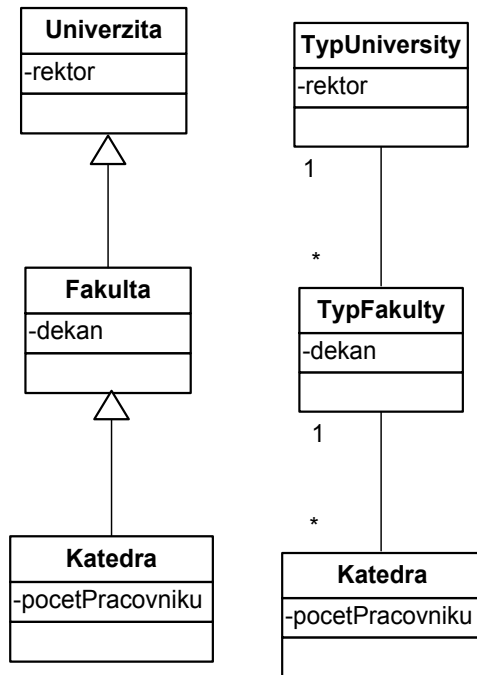
Typování & generalizace 2/2

- Typování se používá v případě, že seznam typů se mění a je malá důležitost dědění.
- Generalizace se používá v případě, že seznam specializovanějších entit statický a je větší důležitost dědění.
- Při generalizaci instance specializovanější datové entity dědí strukturu datových atributů od obecnější datové entity.
- Při typování specializovanější instance dědí od obecnější instance nejen strukturu datových atributů, ale zároveň i jejich obsah.

Typování

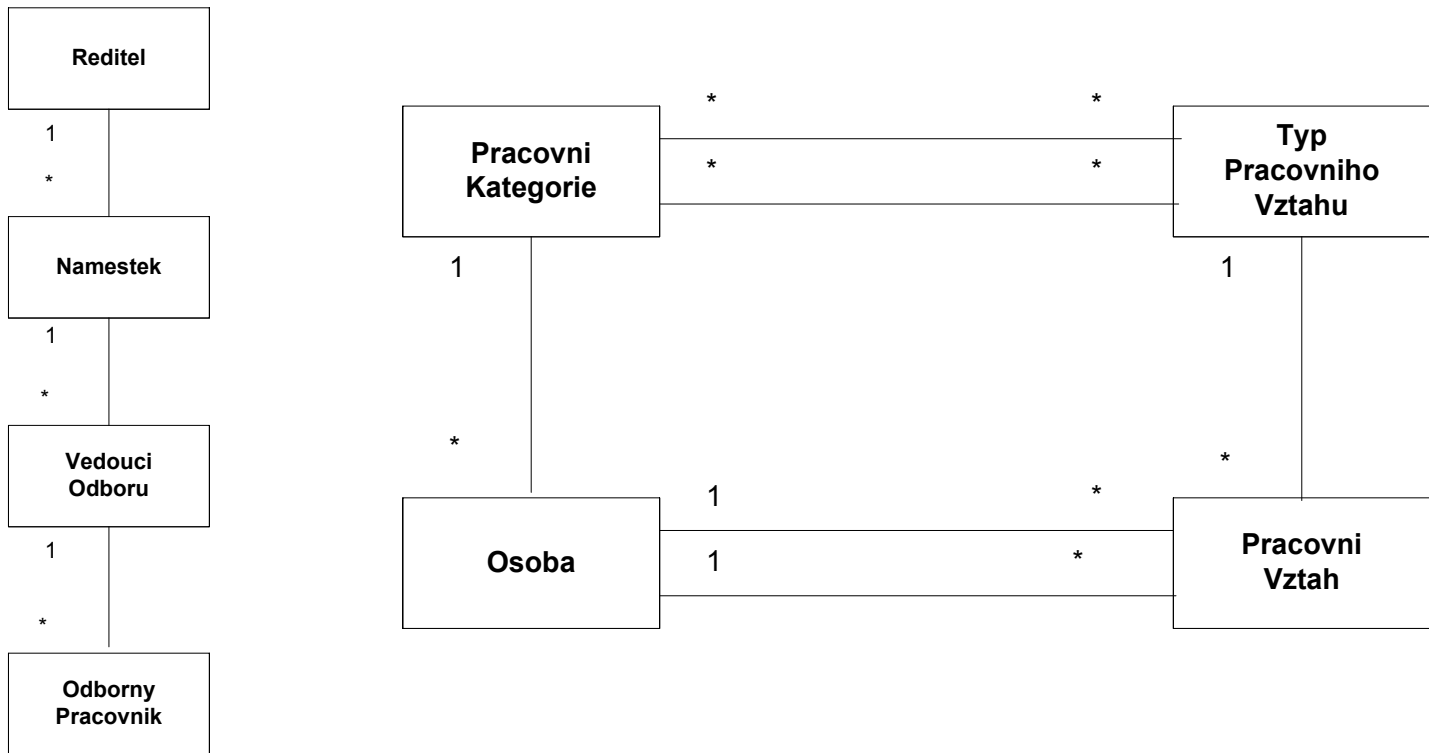
- Datové entity typ se vzájemnými relacemi patří do metaúrovně (směrnice, obecné zákony).
- Standardní entity vyjadřují operační úroveň (každodenní skutečnost).
- Mezi operativní úrovní a metaúrovní existuje vzájemný vztah – operativní úroveň zachycuje konkrétní případy, které musí (měly by být) splňovat pravidla předepsaná metaúrovní.

Stratifikované typování



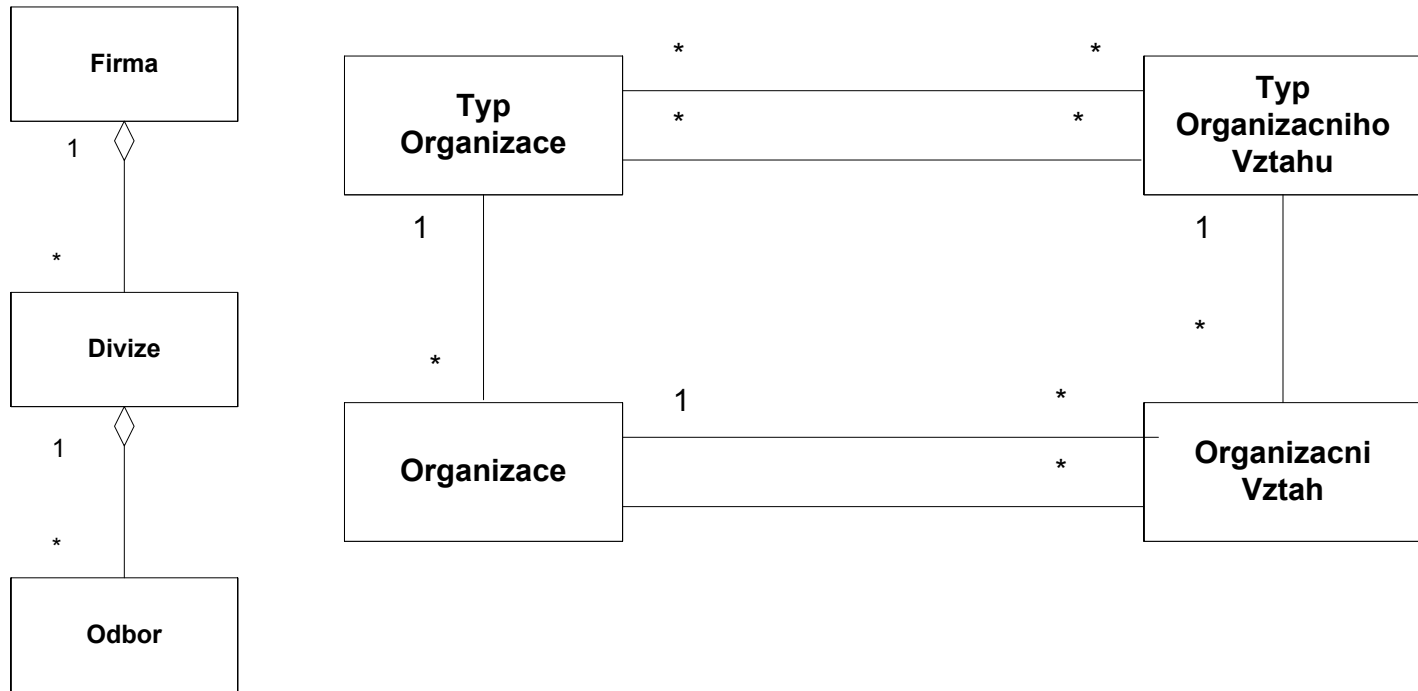
- Instance od třídy `TypUniversity` jsou např. `VŠB-TU`, `SU` nebo `OU`. Ty mají konkrétní různé hodnoty jmen rektorů.
- V rámci instancí `VŠB-TU` se pak vytváří instance `EkF`, nebo `FEI` se svými děkany a společným rektorem. Pro `SU` a `OU` je to podobné.
- V rámci instancí fakult se vytváří konkrétní instance kateder např. `katedra matematických metod`, `informatiky v ekonomii` pro `EkF` nebo `informatiky a matematiky` pro `FEI`.

Relace mezi pracovními kategoriemi a pracovními vztahy



- Generalizace – nepružné řešení
- Typování – vhodné řešení

Flexibilní reprezentace organizační struktury



- Podobné řešení platí i pro organizační strukturu firem.
- Výsledné řešení odpovídá vzorům z článku o politikách

Typování a generalizace



František Huňka