



# PROJEKTOVÁNÍ IS

**P1**  
**2007-10-04**

## PROJEKTOVÁNÍ INFORMAČNÍCH SYSTÉMŮ:

### Cíl předmětu:

- ✓ Objektový přístup k tvorbě systémů
- ✓ Principy a techniky objektového modelování
- ✓ Metodologie OMT – nástroj analýzy a návrhu
- ✓ Porovnání metodologií

### Seznam literatury:

- ✓ Vrana – Projektování IS
- ✓ Vrana, Richta – Zásady a postupy při zavádění podnikových informačních systémů
- ✓ Duben – Metodika analýzy a návrhu systému pomocí Objektové Modelovací Techniky

### Zápočet:

- ✓ Maximálně 1 neomluvená neúčast
- ✓ V průběhu semestru (v polovině) test – alespoň 5 bodů z 10, podkladem přednášky
- ✓ Odevzdat semestrální projekt

## CO JE TO OBJEKTOVÁ ORIENTACE:

- ✓ Nový způsob myšlení o problémech
- ✓ Objekt – základní stavební prvek
- ✓ Sdružuje datový a funkční pohled
- ✓ Základní **charakteristiky objektu**:
  - Identita
  - Klasifikace
  - Polymorfismus
  - Dědičnost

### Identita:

- ✓ Identita znamená, že data jsou kvantována do rozlišitelných entit – objektů.
- ✓ Každý objekt má svou vlastní identitu – 2 objekty jsou rozdílné přesto, že hodnoty všech jejich atributů jsou identické (jméno, velikost...)
- ✓ Tátův kufr, mámin kufr, kráva, kotva.

### Klasifikace:

- ✓ Objekty se stejnou datovou strukturou (atributy) a stejným chováním (operacemi) jsou seskupeny do tříd.
- ✓ Třídy zohledňují vlastnosti podstatné pro aplikaci a ostatní ignorují.
- ✓ Třída je (nekonečná?) množina jednotlivých objektů.
- ✓ Každý jednotlivý objekt je výskytem (instancí) své třídy.
- ✓ (Třída je abstrakce, předmět má identitu.)
- ✓ Každá instance třídy má své vlastní hodnoty všech atributů, ale sdílí společnou strukturu (jména) atributů a operací s ostatními instancemi této třídy.
- ✓ **Objekty X třídy**:
  - Objekt – kufr, třída – kufr, atributy – výška, šířka, zobecněné na operace – ulož, naplň, oprav
  - Objekt – šálek, třída – šálek, atributy – obsah, účel, materiál, zobecněné na operace...

### Polymorfismus:

- ✓ Stejná operace se může chovat odlišně v různých třídách.
- ✓ Specifická implementace operace nad určitou třídou se nazývá metoda.



### Dědičnost:

- ✓ Sdílení atributů a operací mezi třídami na hierarchickém principu.
- ✓ Každá podtřída přebírá (dědí) všechny vlastnosti své nadtřídě a přidává své vlastní unikátní vlastnosti.
- ✓ Tento princip je jedním ze základních výhod OOS a velmi zmenšuje opakování v návrhu.

### Poznámky k objektové orientaci – několik principů dobře podporovaných OO technologií:

- ✓ **Abstrakce:**
  - Soustředění se na podstatné a přirozené vlastnosti objektů a ignorování okrajových vlastností.
  - Zajímáme se co objekt je a co dělá a ne o implementaci.
  - Abstrakce vždy sleduje určité **hledisko**, které určuje co je a co není důležité.
- ✓ **Zapouzdření** (skrytí informace) – odlišujeme externí aspekty objektu, které jsou dostupné jiným objektům od interních (implementačních) detailů, které nejsou viditelné z jiných objektů.
- ✓ **Sdílení** – dědičnost umožňuje sdílet společné datové struktury a chování několika podobnými podtřídami (bez redundance)
- ✓ **Důraz na strukturu objektů** – ne na strukturu procedur:
  - Zajímáme se spíš co objekt je, než jak se používá.
  - Systémy postavené na struktuře objektů jsou dlouhodobě stabilnější.
- ✓ UML, OMT

### Metodologie OMT (Object Modeling Technique) – fáze životního cyklu:

- ✓ **Analýza:**
  - Počíná formulací problému – budování modelu reálného světa (podstatných vlastností).
  - Model je stručná a přesná abstrakce toho **co** musí dělat očekávaný systém (ne **jak** se to provede).
  - Model má být představou aplikační oblasti, nikoliv počítačové implementace.
  - Nesmí obsahovat implementační rozhodování.
- ✓ **Návrh systému:**
  - Celková architektura
  - Rozdělení do subsystémů
- ✓ **Objektový návrh:**
  - Návrh již obsahuje implementační detaily.
  - Předmětem zájmu jsou datové struktury a algoritmy jednotlivých tříd objektů.
- ✓ **Implementace:**
  - Třídy objektů a jejich vztahy jsou převedeny do příslušného programovacího jazyka, databáze, resp. HW implementace.

### Tři modely:

- ✓ **Objektový model:**
  - Statická struktura objektů a jejich vztahů.
  - Objektový diagram – objekty = uzly, vztahy = souvislosti.
- ✓ **Dynamický model:**
  - Změna aspektů systému během času
  - Stavový diagram – stavy = uzly, události = souvislosti
- ✓ **Funkční model:**
  - Transformace dat uvnitř systému
  - Data Flow Diagram – procesy = uzly, datové toky = souvislosti
- ✓ => tyto 3 modely se vzájemně doplňují a jsou propojeny.
- ✓ => základní je objektový model, protože je nutné popsat **co** se mění, dříve než **kdy a jak**.
- ✓ Důraz na strukturu objektů spíše než na strukturu procedur => dlouhodobá stabilita.
- ✓ Hlavní přednost OO vývoje není úspora času.