

OOT

Objektově orientované technologie

Historické souvislosti
Objekt, vlastnosti

Daniela Szturcová
Institut geoinformatiky, HGF

Objektově orientované technologie

- Historické souvislosti
- Vývoj používaných technik
- Objekt – definice
- Objekt – stav, chování, zprávy
- Zapouzdření
- Polymorfismus
- Dědičnost

Historické souvislosti

- Lenost -> podpůrné prostředky pro ulehčení práce.
- Dnes počítačová podpora na úrovni přesahující možnost zvládnout jedním člověkem.
- Tvorba systému vychází z výše uvedených aspektů a zahrnuje
 - prostředky CASE, metodiky, techniky apod.
 - rozklad systému, rozdělení práce mezi tým

Historické souvislosti (2)

- Specializace v oblasti práce – analytik, designér, programátor (od lovců přes zemědělce a průmyslníky k IT profesím).
- Je to výsledek postupného narůstání úloh v oblasti IT, jak z hlediska velikosti řešených úloh, tak jejich složitosti a pokrývaných oblastí.
- Určité úlohy se opakují – princip znovupoužitelnosti již ověřeného řešení.
- Neustále se objevují nové – demo verze novinek, nestabilita, vývojové trendy.

Historické souvislosti (3)

Pomůcky:

- Prsty, kamínky, kresby do písku, ...
- Stabilnější záznamy – papyrus, kamenné bloky – opakovatelnost postupů.
- Kuličková počítadla, mechanické strojky – technická podpora.
- Užití „počítačů“ – výpočet síly atomové bomby za II. světové války – kancelářské strojky IBM.
- Internet – celosvětová provázanost: úloh, řešení, prostředků, dat, výsledků.

Používané techniky

- Vytváření programu v programovacím jazyce - „vyšší“ programovací jazyky – menší úkoly – syntéza do větších systémů.
- Vytvořené programy jsou pomocí překladače přeloženy do strojového kódu.
- Problémy:
 - přenositelnost
 - rychlost použitých algoritmů
 - znovupoužitelnost
 - údržba – chyby, vylepšování

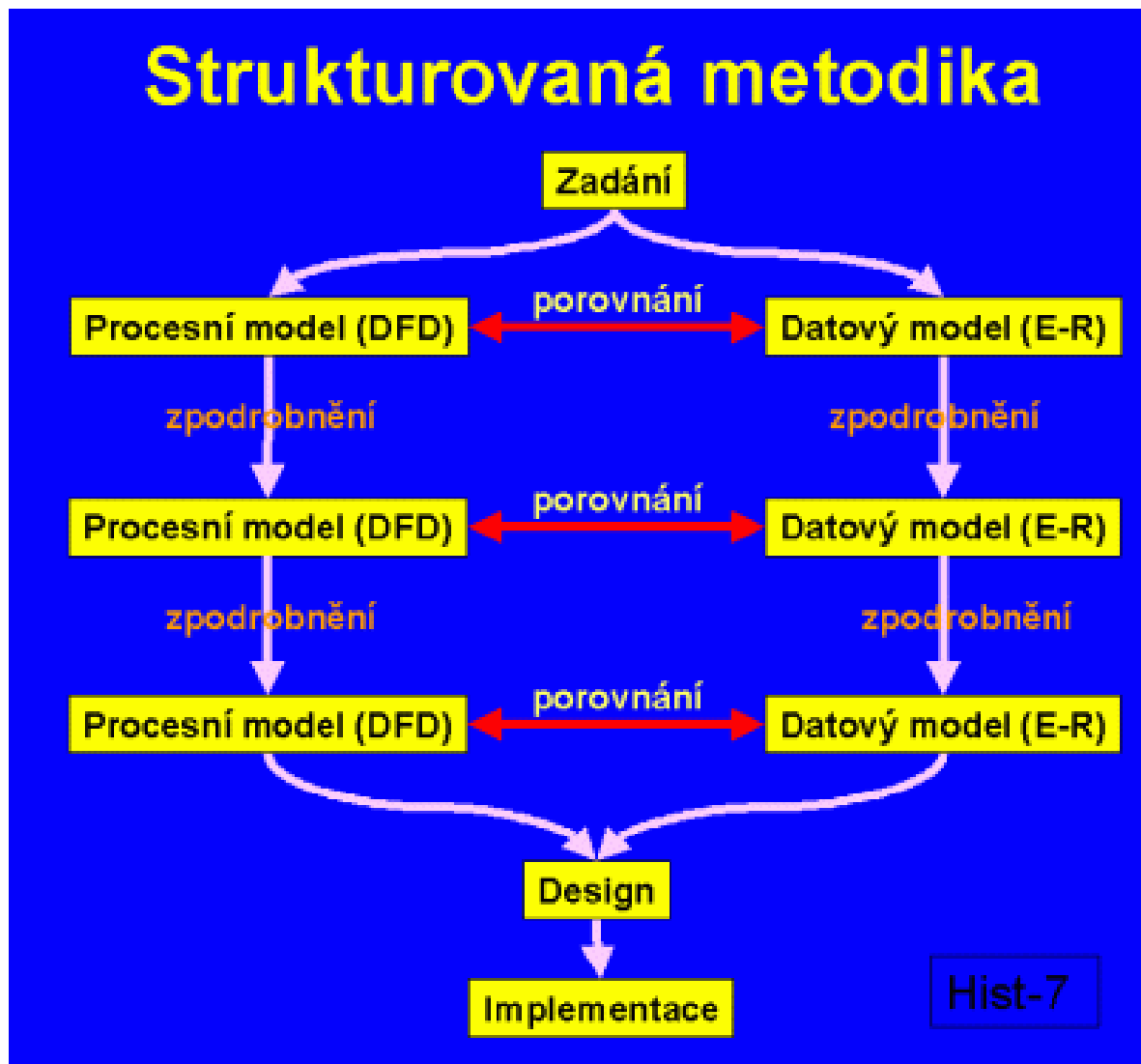
Používané techniky (2)

- Programovací jazyky – neustále nové
- Hierarchický rozklad větších úloh na menší:
 - opakované úlohy – součást technologií - zabudováno do OS (File System), programovací jazyky (SORT, random), správa paměti (zásobník)
 - specializace podpůrných nástrojů – tabulkový kalkulátor, SŘBD
 - podprogramy
 - strukturované programování, modulární programování, techniky programování, metodiky

Metodiky

- Metodika říká **proč** to dělat, programovací technika říká **jak** to udělat – rozdělení rolí – analytik + programátor.
- Velikost úloh vedla k většímu kladení váhy na zvládnutí metodik - těžko se udržovaly „slepované“ systémy.
- Rozklad při řešení složitých úloh – dva základní pohledy: data + procesy.
- Datové a funkční modely – souvztažnost.

Metodiky



[<http://objekty.vse.cz/Programovani/Historie-metodiky>]

Metodiky (2)

Nedostatečnost hierarchického rozkladu vedla k rozčlenění tvorby systému na etapy.

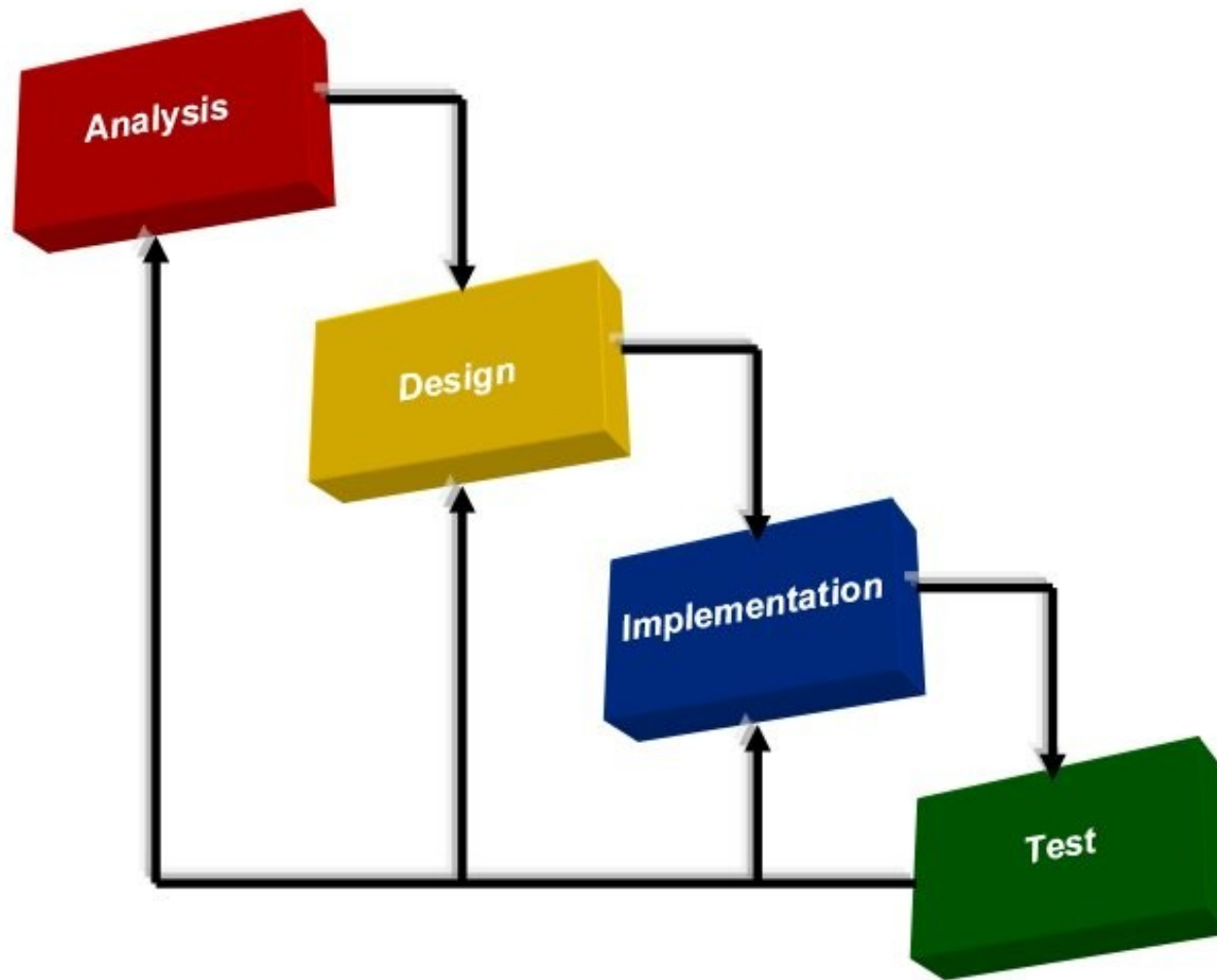
- Strukturované metodiky – DFD, E-R model
- Rozdělení do etap
 - zadání
 - analýza
 - design
 - implementace
 - testování, předání do provozu a údržba
- Možnost nápravy – návrat do předešlých etap.

Metodiky (3)

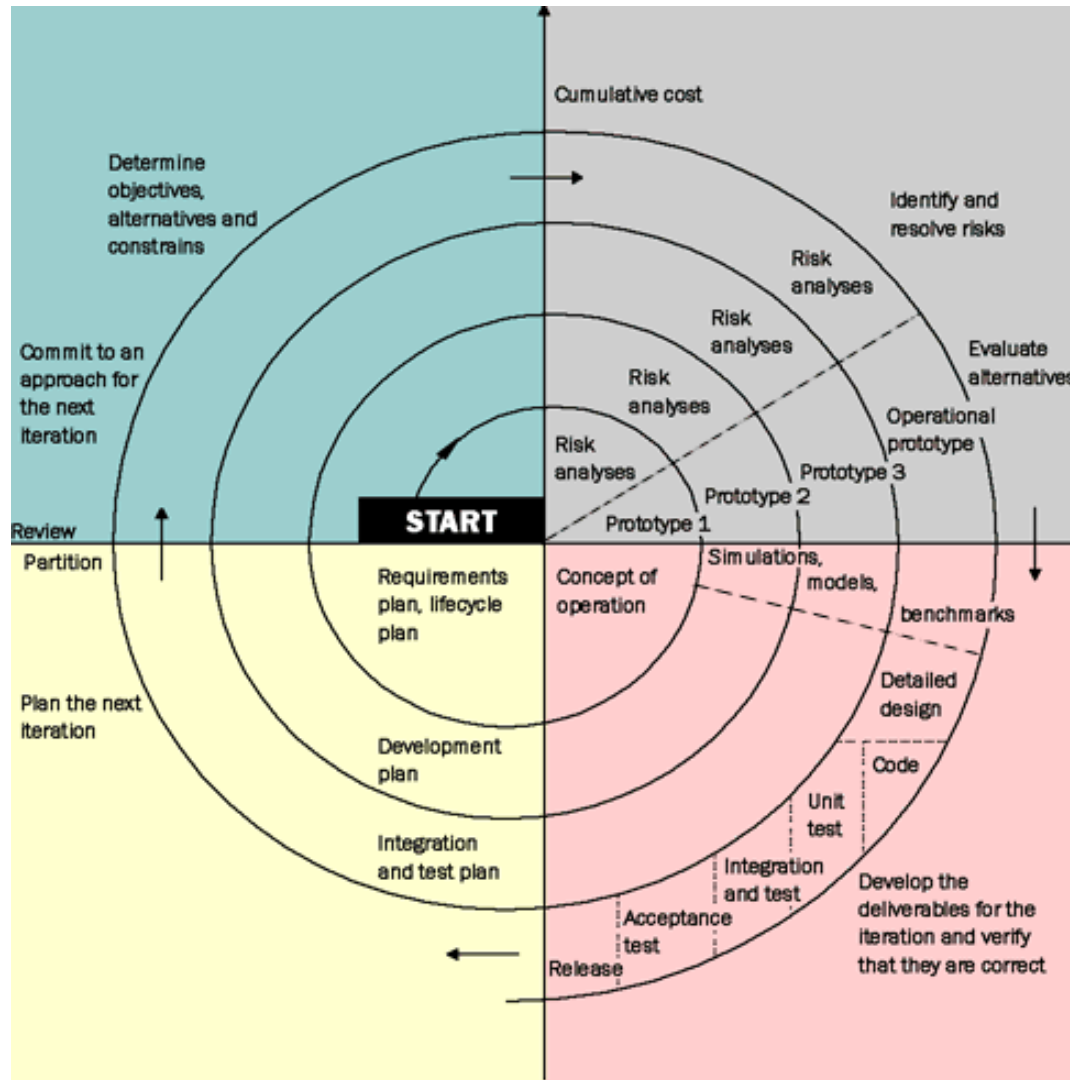
Podpůrné prostředky

- CASE nástroje - „malovátka“ modelů, generování kódu.
- Tvorba prototypů.
- Strategické postupy – vodopádový model, iterační model.

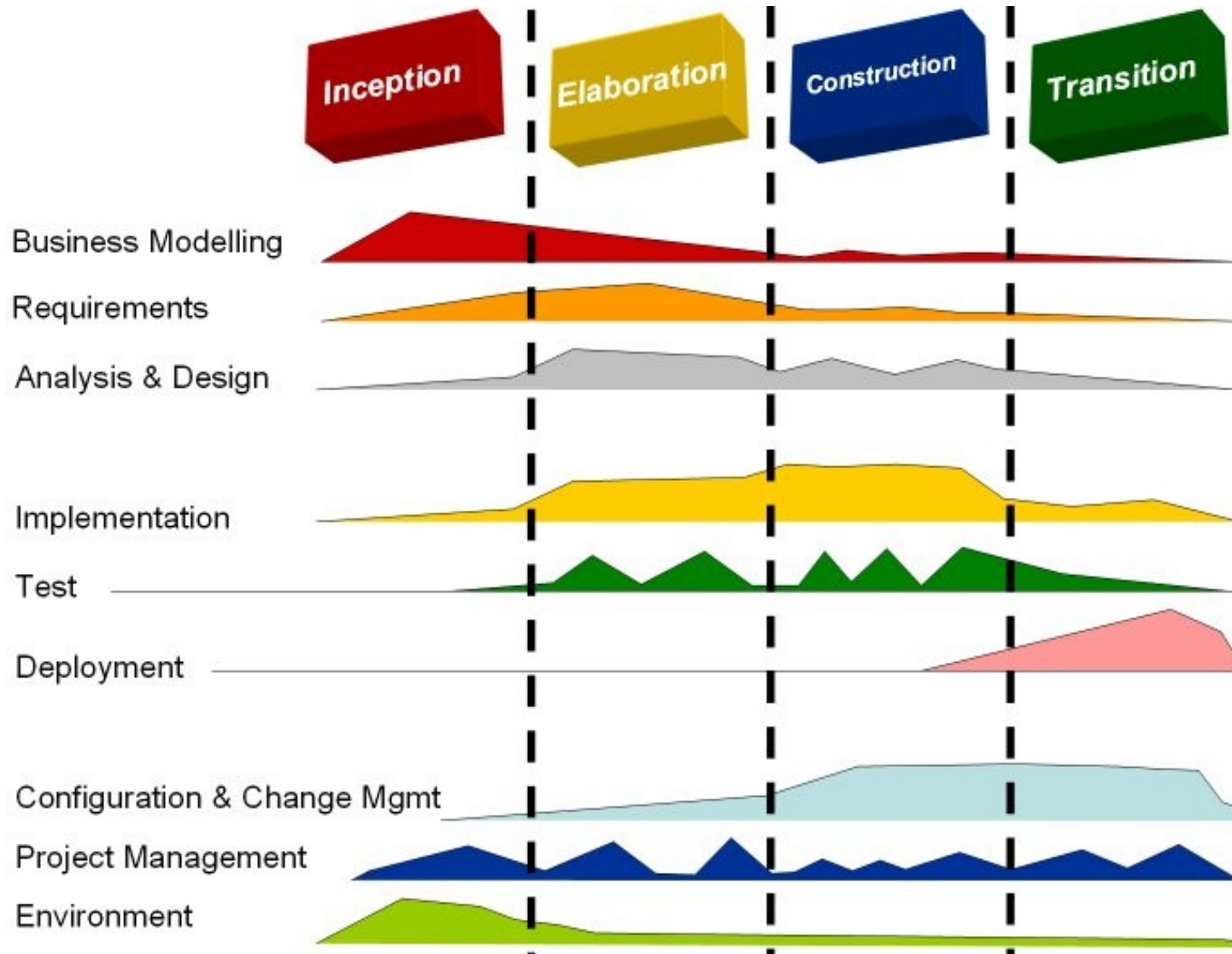
Vodopádový model



Spirálový model



RUP model



Objektově orientovaný přístup

- Simulace reálných systémů - první vlastnosti objektů a princip řízení událostmi.
- SIMULA 67 (**S**imple **u**niversal **l**anguage) byl oficiálně uveden Ole Johanem Dahlem a Kristenem Nygaardem poblíž Osla.
- 1973 – používán na 250 místech.
- Vývoj byl pozastaven, ale na základě definovaných vlastností navázal vývoj jazyka Smalltalk, později také C++ a další.

Data x objekty

- Data součást programů.
- Data uložena v souborech.
- Data v databázových systémech.

- Objekty umožňují tvořit adaptivní systémy.

Objekt v OOT

- **Objekt** – entita s jasně definovanou hranicí a identitou zahrnující její chování a stavy [Vondrák 2004].
- **Objektově orientovaná architektura systému**
- struktura propojených objektů, které prostřednictvím vzájemné komunikace (interakce) definují výsledné chování celého systému [Vondrák 2004].

Definice objektu

- Objekt je identifikovatelná samostatná entita, která je daná:
 - **identitou** – jedinečnost, která ji umožňuje odlišit od ostatních objektů
 - **stavem**
 - **chováním** – službami poskytovanými v interakci s ostatními objekty

Definice objektu

U objektu definujeme

- **atributy:** datové hodnoty popisující objekt, mohou se měnit v čase
- **dobu existence:** časový interval daný okamžikem vzniku a zániku objektu,
- **stavy:** odrážejí různé fáze doby existence objektu, charakterizujeme je hodnotami atributů objektu.

Identita objektu

- Každý objekt má svou vlastní identitu, která ho umožňuje odlišit od ostatních objektů.
- Většinou je reprezentována nějakou jedinečnou informací (ISBN, rodné číslo, sériové číslo, ...).
- Objekty, které nejsou popsány žádnou takovou jedinečnou informací mají přesto svou jedinečnou identitu.



Struktura objektu

- Objekt by měl být schopen popsat sám sebe. (kniha má název, autora, ISBN, cenu, počet stran atd.).
- Rozumíme tím popis objektu – de facto seznam sledovaných vlastností.
- Vnější pohled na objekt – množina zodpovědností (služeb), které objekt poskytuje. Formálně popsány pomocí protokolu zpráv, které můžeme objektu zaslat.
- Vnitřní pohled na objekt – umožňuje nahlédnout na všechny atributy, které jsou před vnějším objektem ukryty a na způsob, jak jsou jednotlivé služby implementovány.

Stav objektu

Objekt **Faktura** má atributy DatumVytvoreni, DatumZaplaceni, Castka, Poznámka. Hodnoty atributů jednoho objektu faktura v určitém čase:

DatumVytvoreni = 10.6.2010

DatumZaplaceni = 30.6.2010

Castka = 2000 Kč

Poznámka = větší částce se nebráníme

Pro jednoduchost budeme nyní *stavem objektu* rozumět momentální kombinaci hodnot všech jeho atributů.

Stav objektu

:Faktura

DatumVytvoreni = 10.6.2005

DatumZaplaceni = 30.6.2005

Castka = 2000 Kč

Poznamka = Větší částce se nebráníme

stav můžeme nazvat „ZAPLACENA“

Chování objektu

- Objekty jsou vytvořeny tak, aby mohly provádět požadované akce.
- Objekt Faktura může být například stornován (třída má operaci-metodu Storno) a může vyvolávat událost informující o změně jeho stavu – například událost informující o zaplacení faktury.

Příklady

Fyzické objekty

- Kniha
 - Vlastnosti: ISBN, název, autor, ...
 - Činnosti: lze ji koupit, prodat, přečíst, roztrhat.

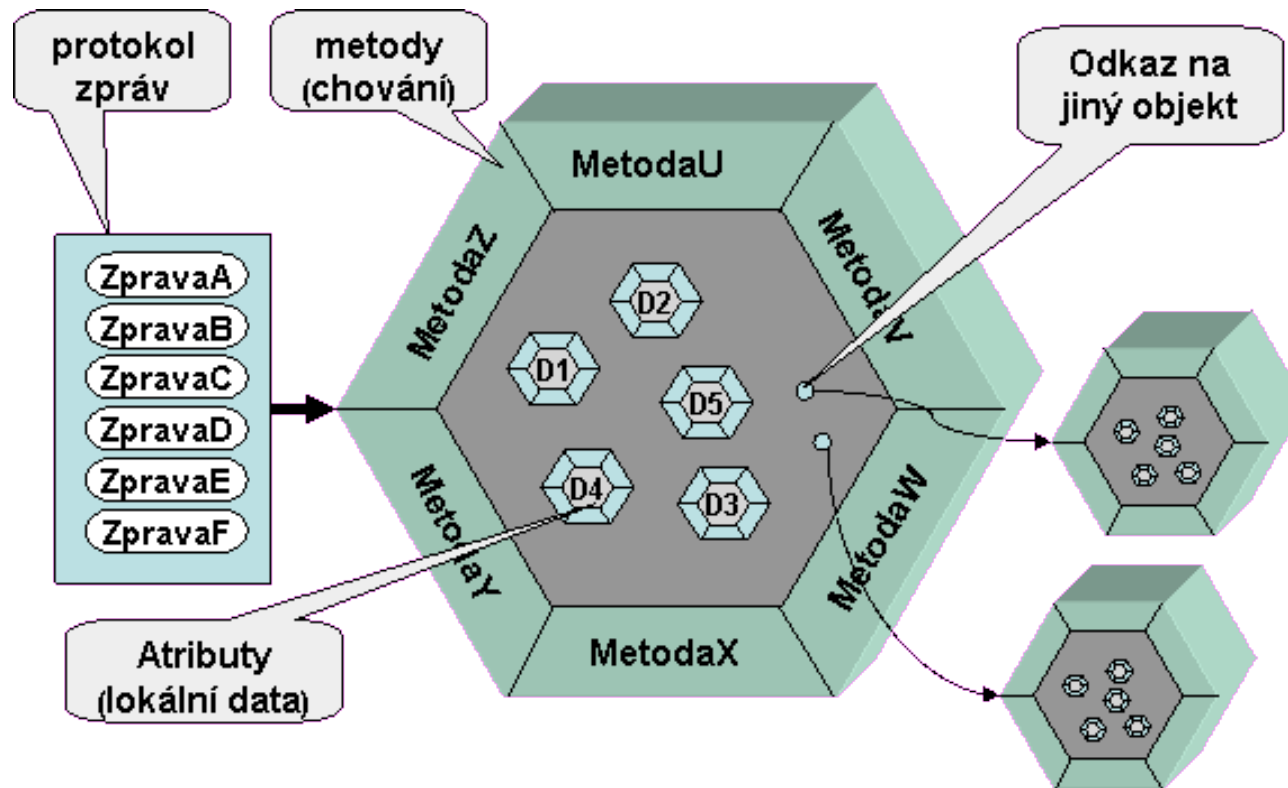
Abstraktní objekty

- Útvar
 - Vlastnosti: počet vrcholů, barva, tloušťka čáry
 - Činnosti: lze jej nakreslit, vybarvit, vypočítat obsah.

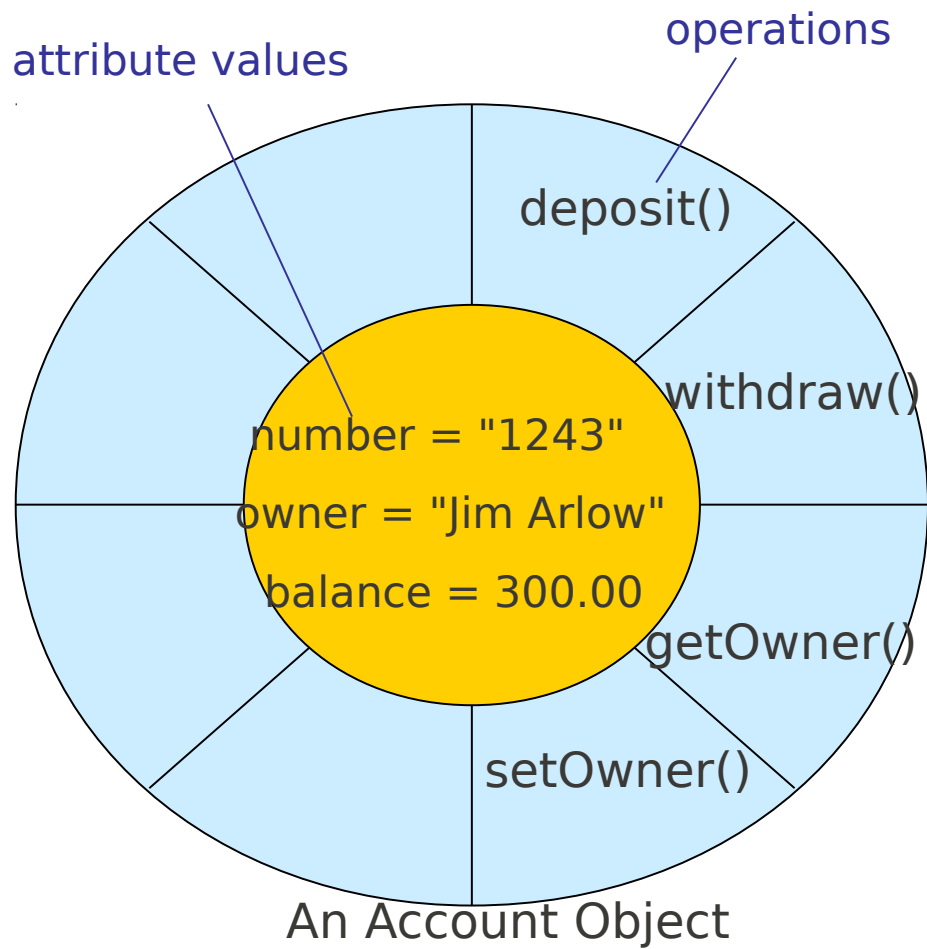
Základní vlastnosti

- Zapouzdření (Encapsulation)
- Dědičnost (Inheritance)
- Polymorfismus

Objekt



Zapouzdření



Objekt se navenek projevuje pouze svými operacemi (realizovanými metodami) a komunikačním protokolem.

Zapouzdření

- Ochrana dat – lze měnit pouze přes metodu.
- Ochrana uživatele – dostane požadovaný údaj – nemusí znát realizaci.

Příklad:

Pošlu dotaz na věk osoby.

- je uložen přímo?
- propočítá se až v okamžiku požadavku z data narození?

Zprávy

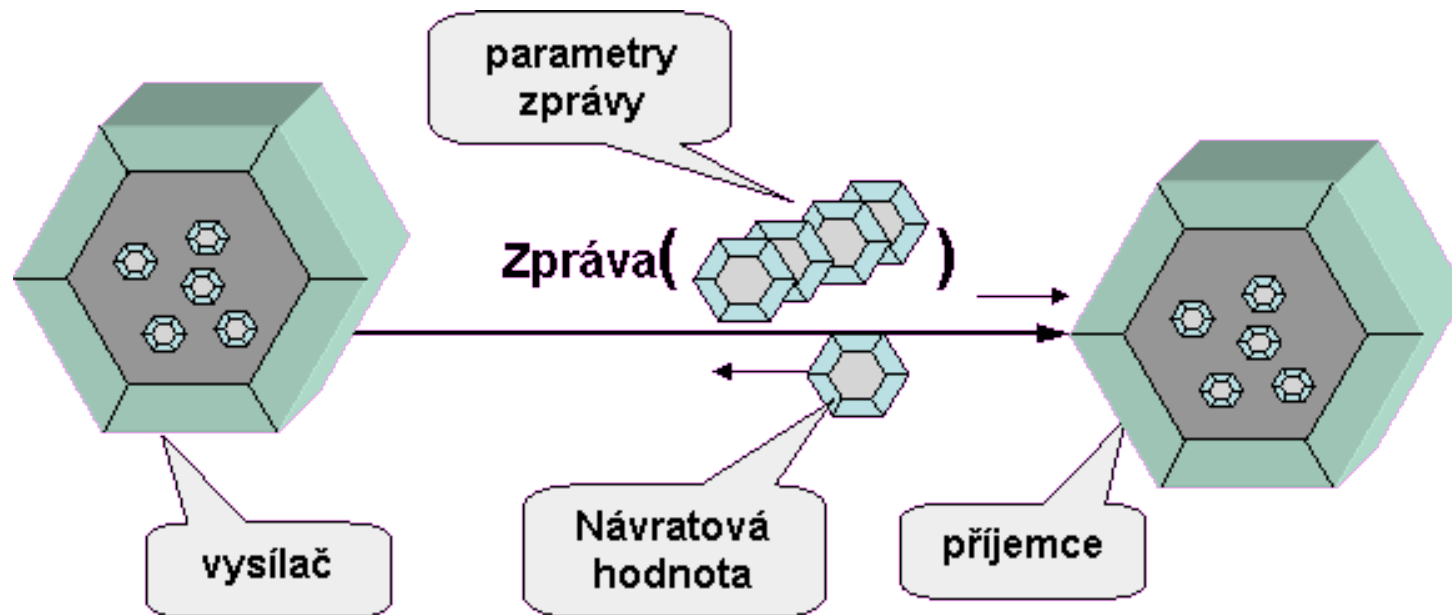
- Komunikace mezi objekty.
- Žádost o provedení operace (metody). Jak bude operace provedena, záleží na příjemci zprávy.
- Reakce na zprávu – ovlivnění odesílatelem.

Struktura zprávy

Aby dokázal **objekt1** odeslat smysluplnou zprávu objektu **objektu2**, musí **objekt1** vědět tři věci:

- **Identifikátor objektu2** (objekt1 musí vědět komu zprávu posílá). Obvykle je identifikátor objektu2 uložen jako jeden z atributů objektu1.
- **Název operace objektu2**, kterou si objekt1 přeje vykonat.
- **Argumenty** (parametry) - Další dodatečné informace, které bude objekt2 potřebovat k provedení požadované operace.

Posílání zpráv



Struktura zprávy

Adresát.metoda(parametr1,parametr2,...)

Příklad:

- Hynku, půjč mi na pivo.
- Viléme, řekni, kolik máš let.
- Jarmilo, podívej se na oblaka.

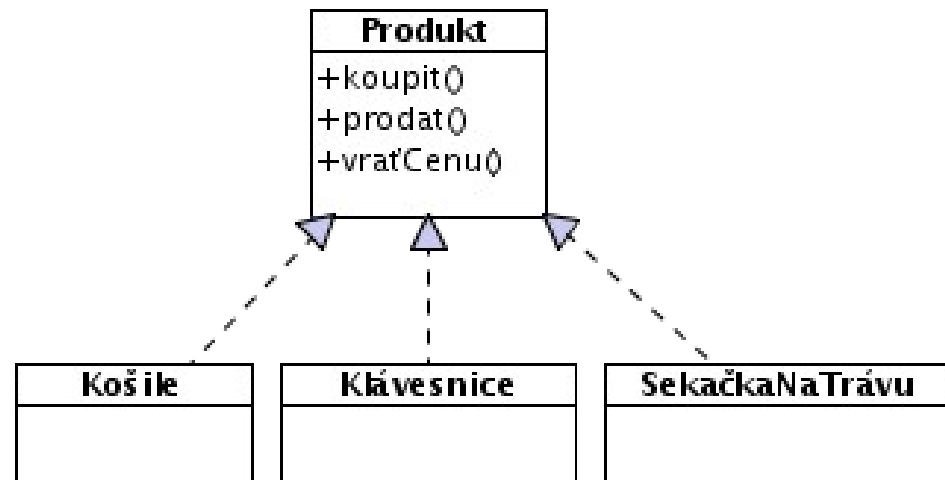
Typy zpráv

- **Informační zpráva** - slouží k poskytování informací cílovému objektu.
- **Dotazovací zpráva** – slouží k získání nějakých informací od cílového objektu.
- **Příkazovací zpráva** – vyžaduje po cílovém objektu provedení nějaké akce na sobě samém nebo na svém okolí.

Polymorfismus

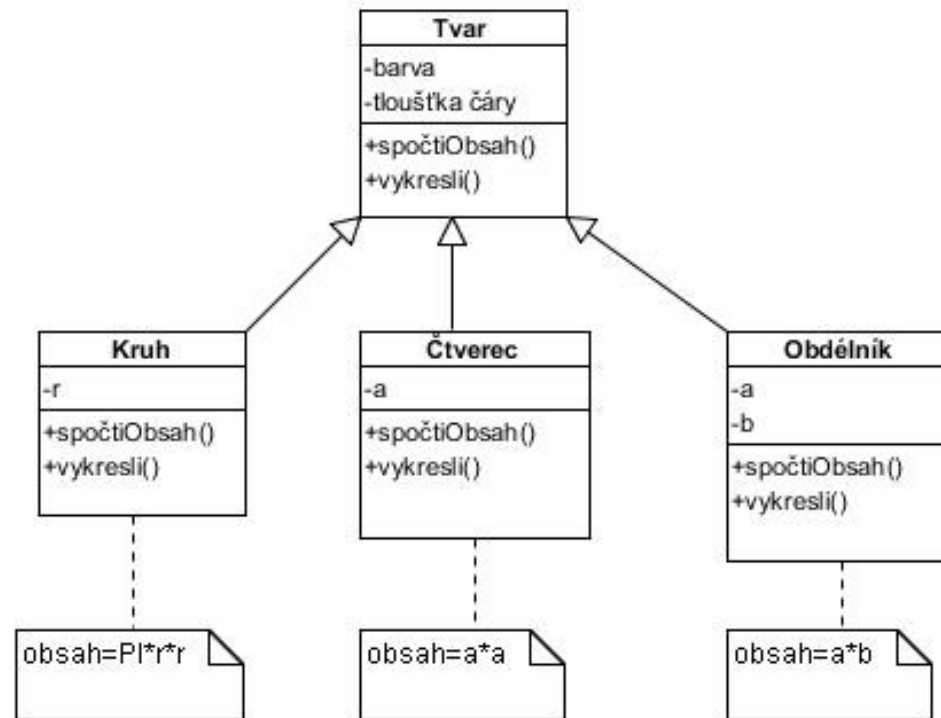
- Dva různé objekty mají ve svém protokolu stejnou zprávu, ale každý z nich na ni reaguje jinou metodou.
- Jedním způsobem, jak použít polymorfismus je **dědičnost**.
- Druhým způsobem je definice **rozhraní**, které definuje názvy operací. Implementační detaily jsou věcí objektu, který dané rozhraní implementuje.

Polymorfismus – příklad



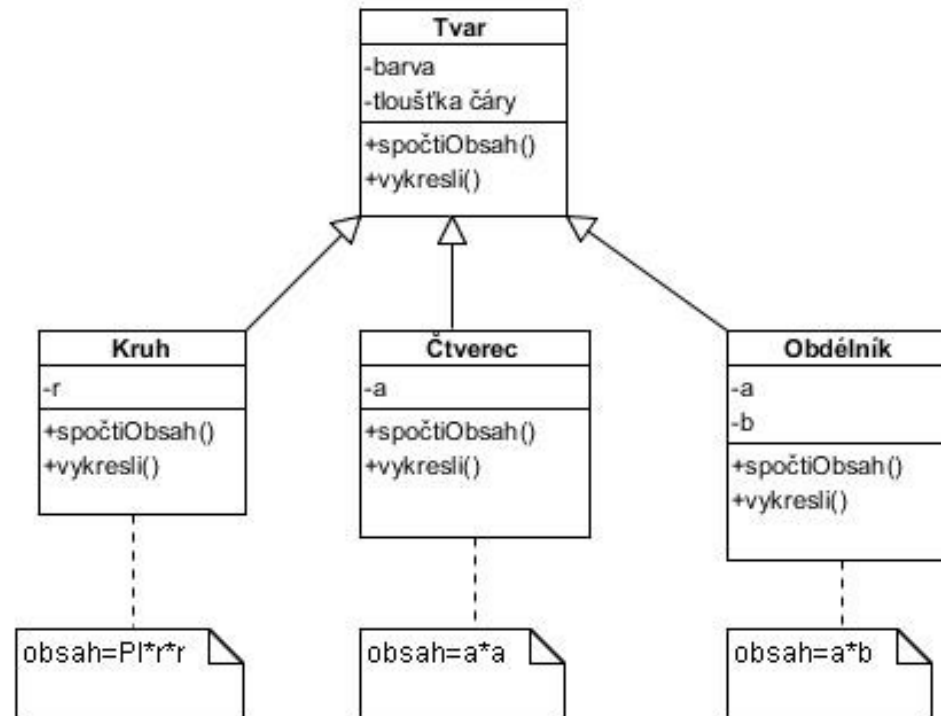
Polymorfismus – příklad 2

Visual Paradigm for UML Standard Edition (VSB-Technical University of Ostrava)



Polymorfismus - příklad 2

Visual Paradigm for UML Standard Edition (VSB-Technical University of Ostrava)



Dědičnost

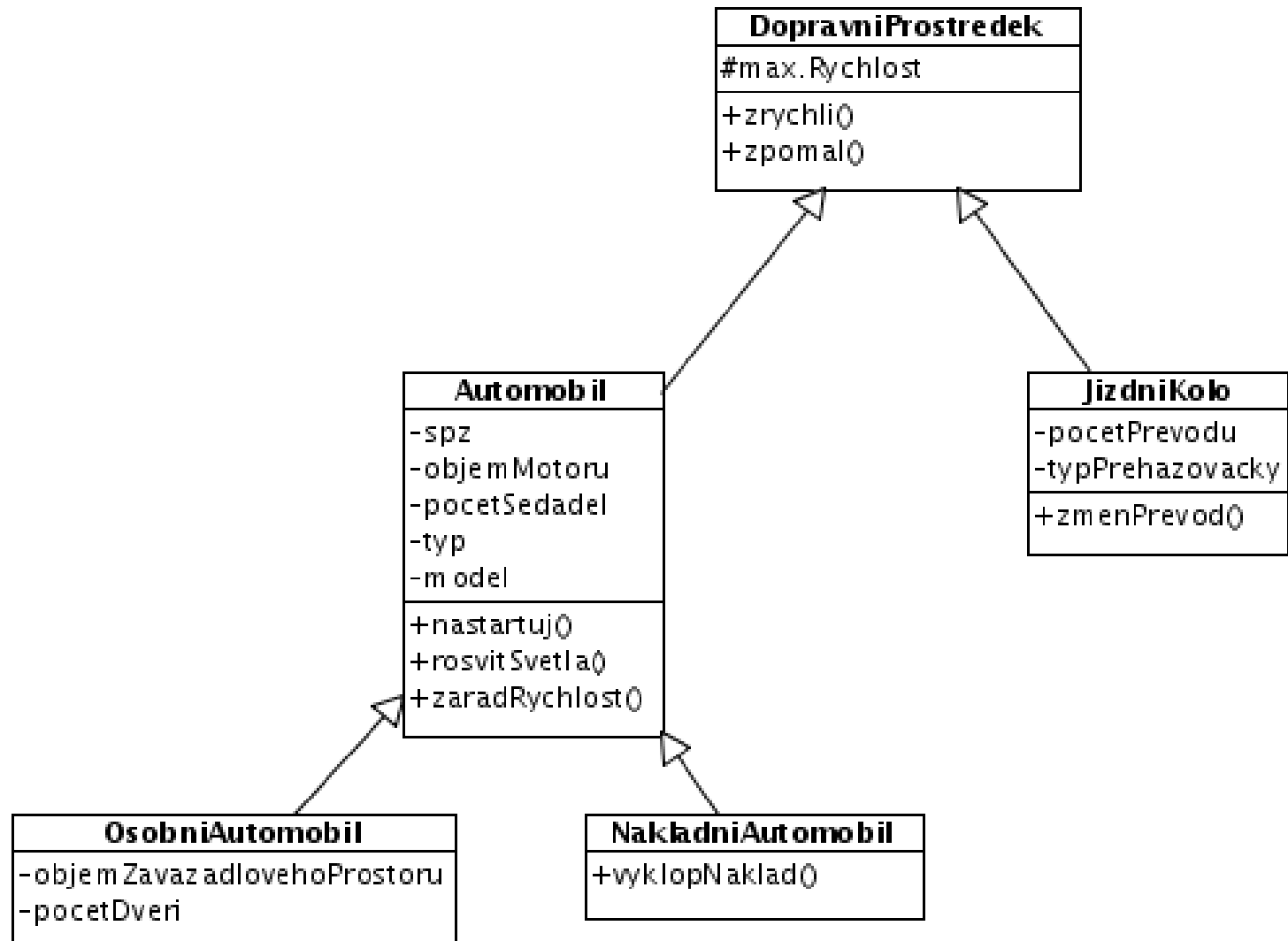
Vztah mezi dvěma třídami (**A** je potomkem **B**).

A přejímá

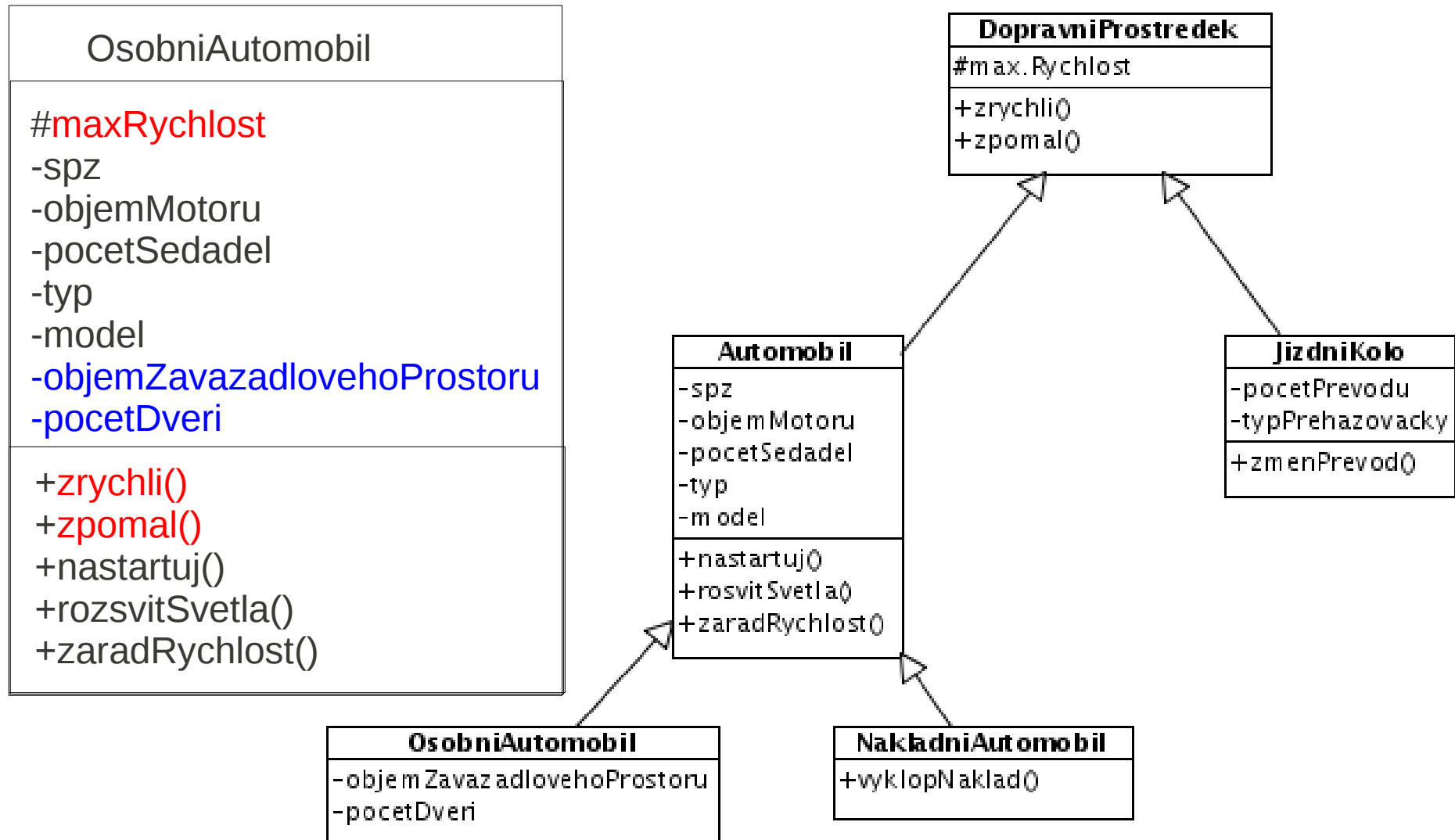
- atributy
- operace (metody)

z rodičovské třídy **B**.

Dědičnost - příklad



Dědičnost - příklad



Význam dědičnosti

- Zjednodušení a srozumitelnost modelu
- Znovupoužitelnost
- Úspora kódu

Zdroje

- <http://mpavus.wz.cz/oo/>
- <http://objekty.vse.cz/Objekty>
- <http://www.wittmannclan.de/ptr/cs/slcycles.html>
- http://newton.cs.concordia.ca/~paquet/wiki/images/e/ea/Spiral_model.gif
- Přednáškové materiály Pavla Děrgela