

Objektové metody modelování

Úvod do objektového modelování

Prezentace 1



**SLEZSKÁ
UNIVERZITA**

OBCHODNĚ PODNIKATELSKÁ
FAKULTA V KARVINĚ

RNDr. Zdeněk Franěk, Ph.D.

Ing. Tomáš Barčák, Ph.D.

Úvod do objektového modelování a jazyka UML



- Syllabus viz IS/SU
- 12 přednášek a seminářů 1+2
- Veřejné obhájení seminární práce: 40 bodů
- Písemná zkouška: 30 bodů
- Ústní zkouška: 30 bodů
- **DOCHÁZKA min. 70 %**
- **MINIMUM PRO SLOŽENÍ ZKOUŠKY > 50**
 - A: 91 – 100, B: 81 – 90, C: 71 – 80,**
 - D: 61 – 70, E: 51 - 60**

Úvod do objektového modelování a jazyka UML



- Software I: MS VISIO
- Software II: ENTERPRISE ARCHITECT ® firmy SPARX ®
- Software III: IBM® Rational® Software Architect
- Literatura základní:
 - H. Kanisová, M. Muller: UML srozumitelně, Computer Press, ISBN 80-251-0231-9 + sylabus.
 - BUCHALCEVOVÁ, A., STANOVSKÁ I. Příklady modelů analýzy a návrhu aplikace v UML, Vysoká škola ekonomická v Praze, Nakladatelství Oeconomica, 2013, 198 s., ISBN 978-80-245-1922-7
- Literatura rozšiřující: ARLOW, J., NEUSTADT, I. UML2 a unifikovaný proces vývoje aplikací. Brno: Computer Press, 2011. ISBN 978-80-251-1503-9

Úvod do objektového modelování a jazyka UML



SLEZSKÁ
UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ
FAKULTA V KARVINĚ

Internetové zdroje:

<http://mpavus.wz.cz>

<http://uml.czweb.org/index.html>

<http://www.rational.com>

<http://www.ibm.com>

<http://www.sparx.com>

<http://ecom.ef.jcu.cz/web2/download/podklady/zakladni-pojmy-ea.pdf>

<http://dspace.upce.cz/bitstream/handle/10195/64679/E1.pdf?sequence=1&isAllowed=y>

<https://www.root.cz/clanky/nastroje-pro-tvorbu-uml-diagramu/>



➤ Co je to UML?

- Modelovací jazyk UML (Unified Modeling Language) umožňuje popsat objektovou analýzu a návrh SW (IS).
- UML je souhrnem především grafických notací k vyjádření analytických a návrhových modelů.
- UML je také jazyk pro vizualizaci, specifikaci, stavbu a dokumentaci SW systémů.
- UML je otevřený standard (OMG: Object Management Group – standardizační komise počítačového průmyslu).

➤ Co je to UML? – souvislosti

- Metodika vytváření informačních systémů **RUP** (**R**ational **U**nified **P**rocess - racionální jednotný proces), firma Rational, od r. 2003 součástí IBM.
- Metodika Select Perspective fy Select Business Solutions.
- Souhrn metod – UML 1997, verze 1.5, nyní 2.0.
- CASE nástroje „Computer Aided Software Engineering“, tzn. počítačem podporované softwarové inženýrství

- Příklady:

Architect Enterprise Architect fy *SPARX* - dostupný na SU OPF

(virtuální počítač <https://view.opf.slu.cz/>)

- **IBM® Rational® Software Architect**

SW není přímo dostupný, ale je dostupná dokumentace a metodika:

<https://public.dhe.ibm.com/software/websphere/JavaDevTools/Presentations/SDP/Handouts/RSA-RationalSoftwareArchitect-V6-DataSheet.pdf>



- Postup analytických prací – modelovací techniky
 - Uživatelské požadavky
 - Procesní modelování
 - Případy užití
 - Modelování tříd a objektů
 - Diagramy objektové spolupráce
 - Stavové diagramy
 - Diagramy aktivit
 - Datové modelování a mapování tříd objektů do tabulek relačních databází
 - Případové studie – příklady



Metodiky a nástroje vývoje SW

➤ Metodologie

- RUP, UP, Select Perspective

➤ CASE nástroje

- Enterprise Architect
- Software Rational Enterprise Architect
- OPEN SOURCE např. IO DRAW
- Komponentový vývoj

Modelování

- **Definice: Modelování je proces, ve kterém se zkoumanému systému označovanému jako dílo přiřazuje podle určitých pravidel jiný systém nazývaný model.**
- Fyzikální modelování, matematické modelování je založeno na podobnosti a je popsáno rovnicemi.
- V informačních systémech se realita vyjadřuje a popisuje pomocí SW a dat. Pro modelování se využívá zejména diagramů.
- **!!! Rozdíl je zejména v roli uživatelů a ve způsobu využívání systému !!!**



Základní pojmy – objektově orientovaný přístup (problémy neobjektového přístupu)

- Dříve vývoj sw byl pojat strukturovaně, rozdělení aplikace na funkční a datovou část.
- Data uložena v souborech, relačních databázích.
- Psaní kódu přístupem shora dolů s využitím procedur.
- Řada nevýhod:
 - složitost, soudržnost datové a funkční vrstvy, manipulace s daty s více míst programu
 - analytický návrh jen v datové části
 - s příchodem architektury „klient – server“ další problémy

Základní pojmy – objekty

- Co je to objekt?
 - Definice: Objekt je seskupení dat a funkcionality, které jsou spolu spojeny za účelem plnění soudržné množiny zodpovědností

- Objekt má:
 - identitu, vlastnosti (atributy), chování (je realizováno metodami) a jedinečnou zodpovědnost (dovednost)

Základní pojmy – objekty

- Objekt poskytuje služby pomocí operací
 - Rozhraní objektu je množinou operací, které nabízí pro jiné objekty (nebo externí agenty)
- Objekt je černá skříňka, která nabízí služby svým klientům
- Objekty spolu komunikují předáváním zpráv:
 - Eliminace datových duplicit
 - Zprávy mohou být vykonány formou vykonání funkcí, znalost identity
 - Komunikace objektů pomocí operací jen definovaných v rozhraní !

Co je to třída?

- Základní předpoklad – návrh modelu tříd (Class model), který v podstatě nezobrazuje jednotlivé objekty, ale šablonu-předpis pro vytvoření objektů = třída objektů.

Vztah mezi třídou a objekty:

- Třída je to co navrhujeme a programujeme.
- Objekty jsou to, co vytváříme (ze třídy) při běhu aplikace.
- Každý objekt má jiný identifikátor a jiný stav v čase, což znamená jiné hodnoty v jeho proměnných.

Vysvětlení pojmů na případové studii

Modelová situace:

SW firma získala zakázku na analýzu, návrh a vývoj IS, který by funkčně pokrýval potřeby sběrný oprav elektrospotřebičů. Sběrna oprav je „zákazník“, předmětem podnikání je zprostředkování oprav ve značkových i neznačkových servisech podle druhu el. spotřebičů.

Pro naše potřeby modelujeme jeden Modul IS pro zprostředkování oprav.

Požadavky zákazníka („requirements“) na modul IS oprava elektrospotřebičů:

01. Příjem zakázky na opravu elektrospotřebiče
02. Výdej zakázky majiteli

Požadavky zákazníka

- Požadavek = popis (specifikace) jisté funkce nebo vlastnosti, která by měla být ve vyvíjeném systému implementována.
- Požadavek = vyjádření přání uživatele.
- Dva základní typy požadavků:
 - Funkční (specifikují požadavky na funkčnost systému).
 - Nefunkční (specifikují jisté vlastnosti systému, případně podmínky omezující funkčnost systému).
- Požadavky by měly říkat, **co** bude systém nabízet a **ne, jak** to zařídí.



Zdroje požadavků I

- Požadavky jsou na samém počátku projektu.
- Sami jako tvůrci systému máme určitou představu.
- Proces získávání požadavků od budoucích uživatelů.
- Různá úroveň uživatelů.
- Konfrontace s představou tvůrců systému.



Zdroje požadavků II

- legislativa.
- požadavky zákazníků
- existující systémy uživatelů
- pracovní procesy uživatelů
- vlastní know-how
- prostředí zákazníka
- hardware, software vybavení



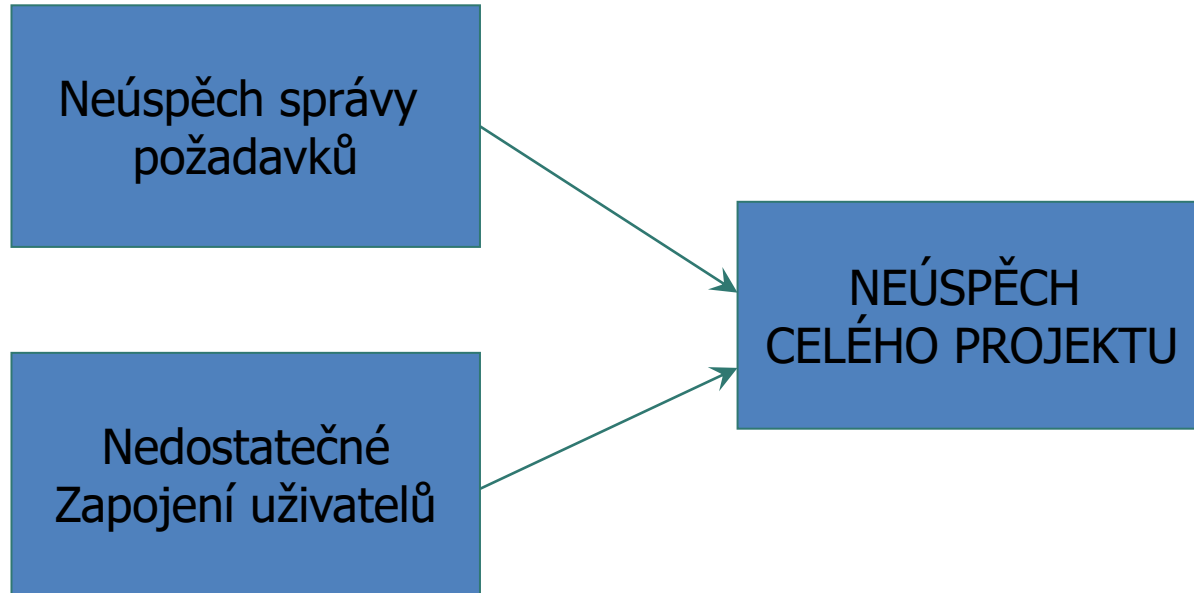
Nefunkční požadavky

- dodržení určitých standardů
- využití určených komponent
- rychlost odezev systému na určité operace
- nároky na výkonnost systému
- bezpečnost systému
- použitá architektura

Úvod do objektového modelování a jazyka UML



Neúspěch správy požadavků



Požadavky – Postup prací:

- Identifikace funkčních požadavků
- Identifikace nefunkčních požadavků
- Identifikace případů užití a jejich navázání k funkčním požadavkům
- Promítnutí nefunkčních požadavků do technické architektury systému
- **! Provázání funkčních požadavků s případy užití má kontrolní funkci !**
- Příklad užití = 1.technika pokrytá jazykem UML
- Případy užití jsou logickým pokračováním analytických prací vycházejících z uživatelských požadavků
- PŘÍPADOVÁ STUDIE = PŘÍKLAD

Případy užití – úvod

- Případy užití, typové úlohy, užité případy = USE CASE.
- Případy užití zachycují přesně funkčnost, která bude IS pokryta a vymezují tak jednoznačně rozsah prací.
- Je součástí UML.
- Každý případ užití popisuje jeden ze způsobu užití systému, popisuje tedy jednu jeho požadovanou funkčnost.
- Scénář, základní scénář.
- Případ užití je sada scénářů, které spojuje dohromady cíl.

Příklad případové studie – IS Sběrny oprav spotřebičů



Studie vychází z modelové situace, kdy SW firma řeší zakázku na analýzu, návrh a vývoj IS, který funkčně pokrývá potřeby sběrný oprav spotřebičů (SbOpSp).

- Předmětem podnikání SbOpSp je zprostředkování oprav el. spotřebičů ve značkových i neznačkových smluvních servisech podle druhu spotřebičů.
- Pro jednoduchost je v dalším výkladu uvažováno pouze o jednom modulu IS – modul pro zprostředkování oprav. (NE další moduly jako je účetnictví, fakturace)
- Zpracování požadavků SW firmou ve spolupráci SbOpSp:

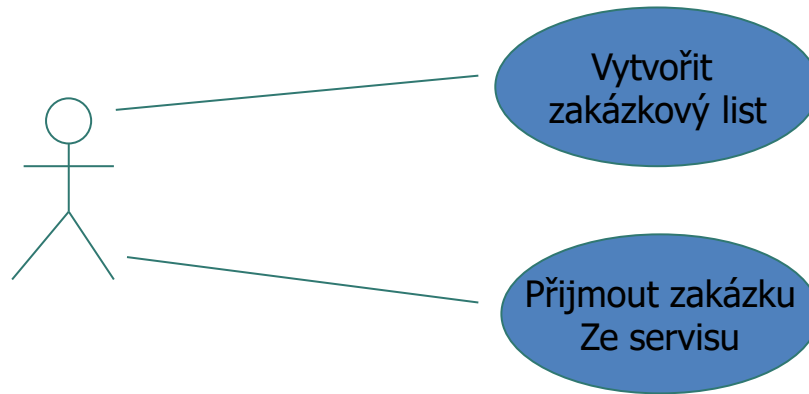
1Příjem spotřebiče do opravy, 2Výdej zakázek majiteli, 3Monitoring oprav, 4Evidence zákazníků, 5Vyhodnocení oprav, 6Vyřízení reklamace opravy, 7Oprava spotřebiče v servisu

Po zpracování požadavků (a popisu firemních procesů, viz předmět Procesní modelování) může SW firma přistoupit k analýze tvorby IS.

USE CASE - Aktéři I



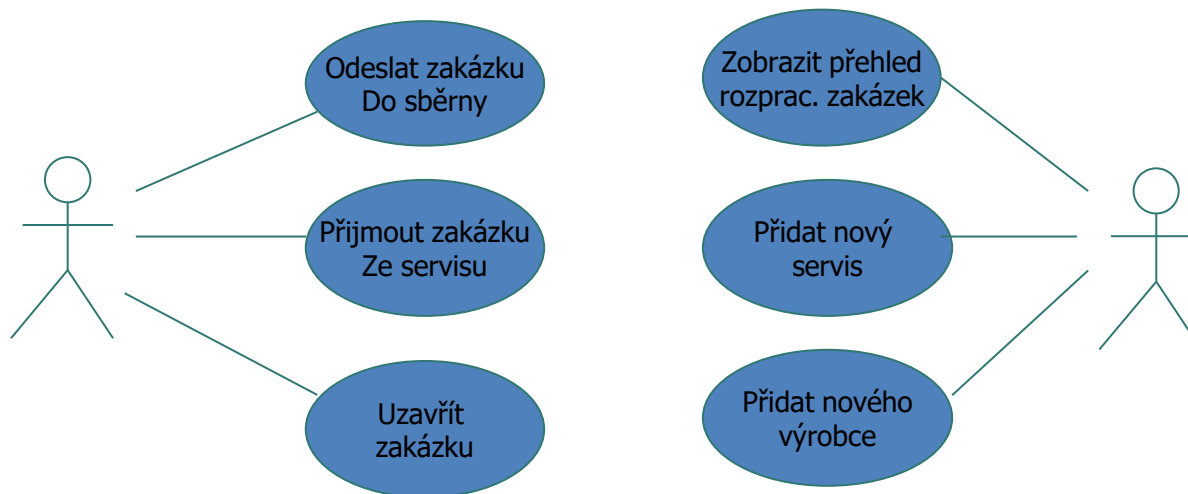
- **Aktér** = role, ve které vystupuje uživatel v rámci jeho komunikace se systémem
- **Př. Aktér** = uživatelská role vůči systému



USE CASE Aktéři II



- Aktér = role, ve které vystupuje uživatel v rámci jeho komunikace se systémem
- Aktérem nemusí být nutně člověk, může to být např. externí systém
- Př. Aktér ve více rolích vůči systému



Úvod do objektového modelování a jazyka UML



USE CASE (Případ užití) - Scénář : Příjem zakázky do opravy

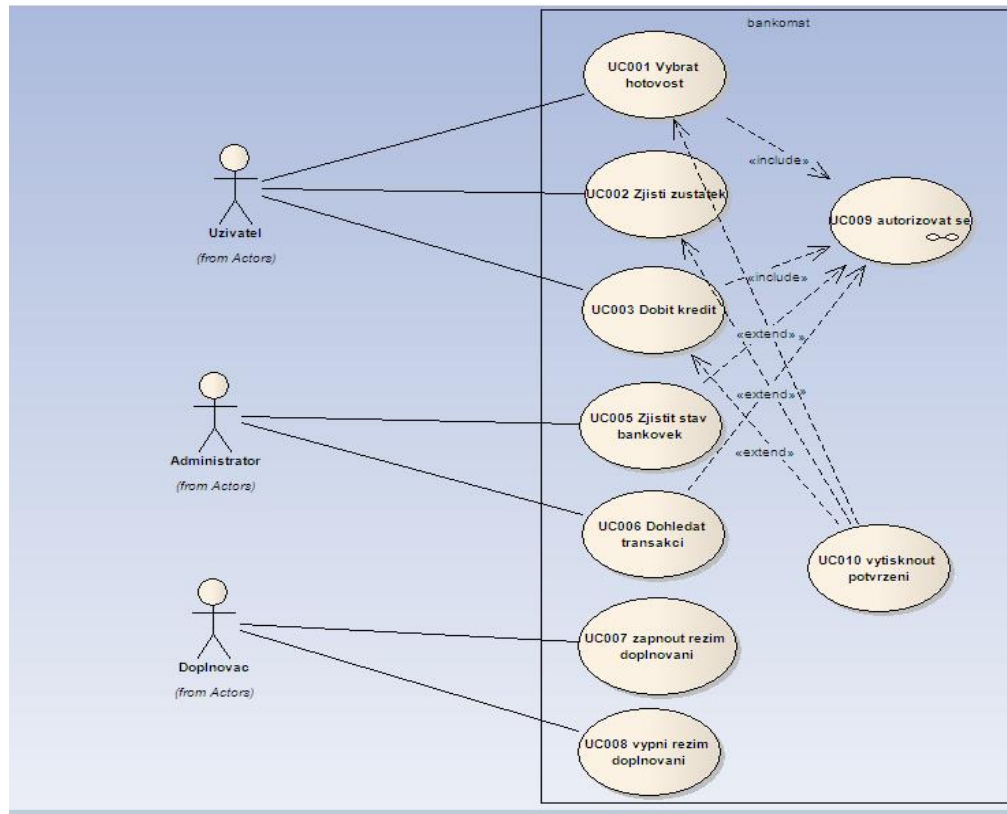
KROK	Role	AKCE
• 1	Uživatel	spustí volbu Založit zakázku
• 2	Systém	zobrazit formulář detailu zakázky a zpřístupní údaje pro pořízení
• 3	Uživatel	pořídí vstupní informace zakázky, jedná se o tyto údaje
• 4	Uživatel	aktivuje výběr zákazníka z evidence sběrný
• 5	Systém	zobrazí formulář seznamu zákazníků v abecedním pořadí
• 6	Uživatel	vybere zákazníka ze zobrazeného seznamu a přiřadí ho k zakázce
• 7	Systém	zavře formulář seznamu zákazníků a vrátí se do editace zakázky
• 8	Uživatel	aktivuje zobrazení seznamu spotřebičů
• 9	Systém	zobrazí formulář seznamu spotřebičů v třídění dle názvu spotřeb.
• 10	Uživatel	vybere spotřebič ze seznamu
• 11	Systém	deaktivuje formulář seznamu <u>spotř.</u> a návrat k editaci <u>zak.</u> listu
• 12	Uživatel	zapiše údaj o poruše spotřebiče a dá pokyn k tisku <u>zak.</u> listu
• 13	Systém	vytiskne zakázkový list a uzavře formulář zakázky

Případ užití je sada scénářů, které spojuje dohromady cíl.

Další příklad USE CASE – pro názornost



Případy užití: Bankomat



Základní pojmy – objekty

- Co je to objekt?
 - Definice: Objekt je seskupení dat a funkcionality, které jsou spolu spojeny za účelem plnění soudržné množiny zodpovědností.
- Objekt má:
 - identitu, vlastnosti (atributy), chování (je realizováno metodami) a jedinečnou zodpovědnost (dovednost).

Základní pojmy – objekty

- Objekt poskytuje služby pomocí operací.
 - Rozhraní objektu je množinou operací, které nabízí pro jiné objekty (nebo externí agenty).
- Objekt je černá skříňka, která nabízí služby svým klientům.
- Objekty spolu komunikují předáváním zpráv:
 - Eliminace datových duplicit.
 - Zprávy mohou být vykonány formou vykonání funkcí, znalost identity.
 - Komunikace objektů pomocí operací jen definovaných v rozhraní !

Co je to třída?

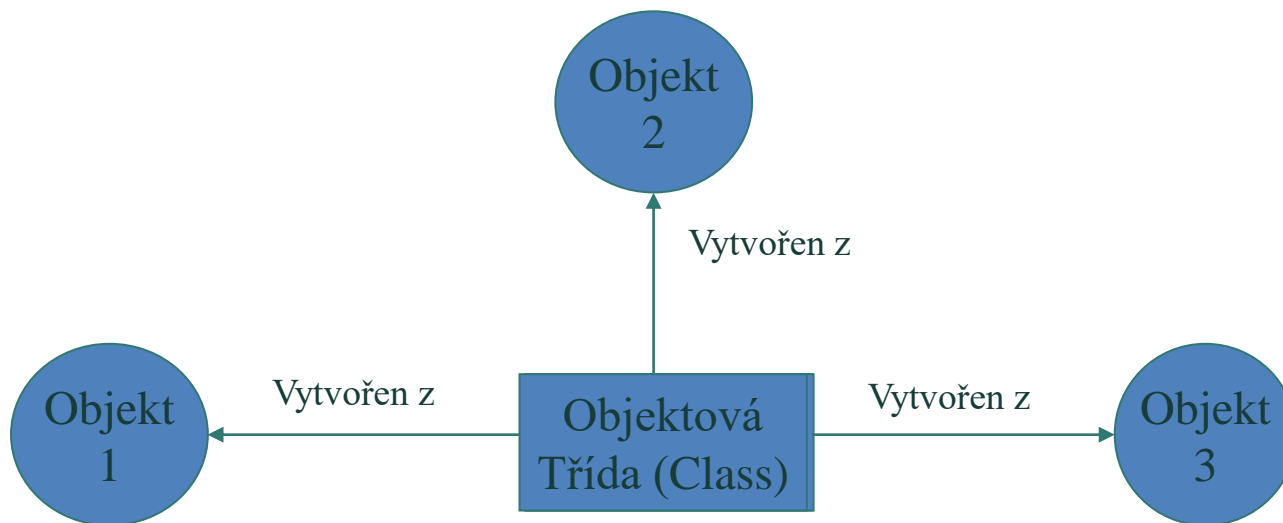
- Základní předpoklad – návrh modelu tříd (Class model), který v podstatě nezobrazuje jednotlivé objekty, ale šablonu-předpis pro vytvoření objektů = třída objektů.

Vztah mezi třídou a objekty:

- Třída je to co navrhujeme a programujeme.
- Objekty jsou to, co vytváříme (ze třídy) při běhu aplikace.
- Každý objekt má jiný identifikátor a jiný stav v čase, což znamená jiné hodnoty v jeho proměnných.

Úvod do objektového modelování a jazyka UML

Vztah mezi třídou a objekty – grafické znázornění:



Struktura tříd

- Struktury tříd jsou založeny na dvou principech:
 - Zodpovědnost třídy (námi definovaný objekt nese zodpovědnost za danou problematiku)
 - Zapouzdření třídy (koncept, který dává ve třídě dohromady to, co spolu souvisí a dává nějaký smysl)

- Atribut tříd je nositel informací o objektu:
 - Název atributu (např. jméno)
 - Formát atributu (např. string)
 - Viditelnost (Public, Private, Protected)

Struktura tříd

- Operace tříd
 - chování objektu je definováno operacemi
 - aktualizací operace vykonávají operace s daty
 - operace typu interface poskytují rozhraní k jiným objektům
 - charakteristika operací je dána názvem, seznamem parametrů a návratovými hodnotami, tzv. signaturou
 - signatura musí být jednoznačná a unikátní
 - z analytického pohledu vystihuje, co daná operace vykonává (např. najdi jméno)

Základní pojmy – objekty a třídy

- Atributy objektu vyjadřují statické datové vlastnosti
 - atributy jsou zapouzdřeny uvnitř objektu, jsou skryty jiným objektům
 - Přístup k atributům je možný jen zasláním zprávy, která vyvolá operaci
 - Jinak vyjádřeno: s atributy mohou manipulovat jenom metody daného objektu

- Objekty jsou organizovány ve **třídách** sdružující jejich vlastnosti.

Základní pojmy – objekty a třídy

- **Třída** představuje šablonu (stupeň řízení) pro skupinu instancí (příslušnost), které nazýváme objekty.
- Šablona popisuje vnitřní strukturu objektu.
- Objekty stejné třídy mají stejné operace, atributy a metody.
- Třídy jsou využívány pro vytváření objektů.
- Model tříd dává základ pro funkci jednotlivých objektů.
- Modelování tříd je klíčovým prvkem objektově orientovaného vývoje.

Vztahy mezi třídami:

- **Agregace**
 - jedna třída je částí druhé
- **Kompozice**
 - agregace, kdy podřízený objekt nemůže existovat samostatně
- **Asociace**
 - znázorňuje vztahy mezi jednou či více třídami (1 ku 1, 1 k mnoha, ...)
- **Generalizace (dědění)**
 - vztah mezi obecnou třídou (super class resp. parent) a jejími potomky (subclass resp. child)
 - dědí se všechny vlastnosti, tj. atributy, relace, operace a omezení

Vztahy mezi třídami:

- **Abstraktní třída**
 - zvláštní třída bez konkrétní instance, zobecnění
- **Polymorfismus**
 - některé objekty mají totožná rozhraní realizovaná pomocí operací, ale metody, které se skrývají za těmito operacemi, jsou rozdílné
- **Asociační třídy**
 - typ vazby mnoha ku mnoha

Diagram tříd zobrazuje strukturu a vztahy mezi objektovými třídami navrhovaného IS.



Děkuji za pozornost

Otázky?
