Objektové metody modelování

Úvod do objektového modelování, třídy, objekty a USE CASE

Tutoriál I

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

RNDr. Zdeněk Franěk, Ph.D.

Úvod do objektového modelování a jazyka UML

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

- Sylabus viz IS.SLU.CZ
- Hodnotící metody: 40 % seminární práce, 60 % kombinovaná zkouška.
- 40 bodů + 60 bodů za kombinovanou zkoušku (30bodů písemný test a 30 bodů ústní zkouška) = 100bodů
- Stupnice pro udělení známky:
- 91-100 = A, 81-90 = B, 71-80 = C, 61-70 = D, 51-60 = E, 50 a méně = F
- DOCHÁZKA min. 70%
- Software I: MS VISIO
- Software II: Enterprise Architect firmy Sparx – trial verze
- Literat.: H. Kanisová, M. Muller: UML srozumitelně, Computer press, ISBN 80-251-0231-9 + sylabus.

Úvod do objektového modelování a jazyka UML

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

- 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)

Úvod do objektového modelování a jazyka UML

- Co je to UML? – souvislosti

  - Metodika RUP fmy Rational, nyní IBM, obecně UP
  - Metodika Select Pespective fmy Select Business Solutions
  - Souhrn metod – UML 1997 verze 1.5, nyní 2.0
  - CASE nástroje Computer Aided Software Engineering
  - Například Rational Architect Enterprise

Úvod do objektového modelování a jazyka UML

- 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

Úvod do objektového modelování a jazyka UML

Metodiky a nástroje vývoje SW

- Metodologie
  - RUP, UP, Select perspektive

- CASE nástroje
  - Software Rational Enterprise Architect
  - Enterprise Architect
  - OPEN SOURCE
  - Komponentový vývoj

Úvod do objektového modelování a jazyka UML

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

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

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

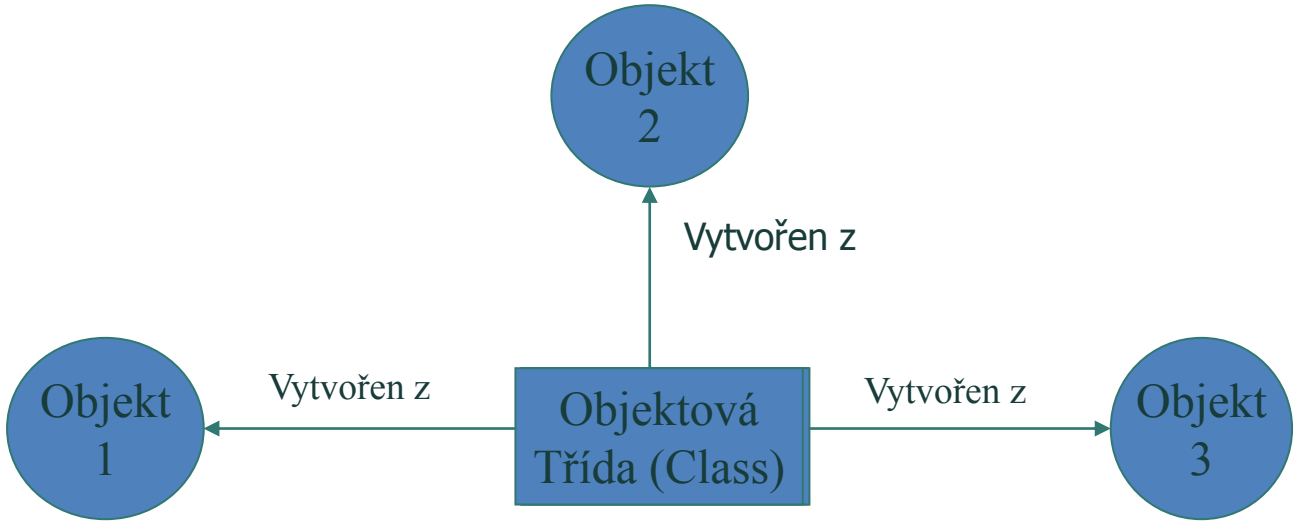
- Základní předpoklad – návrh modelu tříd (Class model), který v podstatě nezobrazuje jednotlivé objekty, ale šablonu-předpis pro vytvoření objektů = třída objektů

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.

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

Vztah mezi třídou a objekty – grafické znázornění:



Struktura tříd

- Struktury tříd jsou založeny na dvou principech:
  - Zodpovědnost třídy
  - Zapouzdření třídy

- 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
  - aktualizační 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

Úvod do objektového modelování a jazyka UML

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

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

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

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

Vysvětlení pojmů na případové studii

Modelová situace

Sw fma získala zakázku na analýzu, návrh a vývoj IS, který by funkčně pokrýval potřeby sběrny oprav elektrospotřebičů. Sběrna oprav je „zákazník" a 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

Vysvětlení pojmů na případové studii II

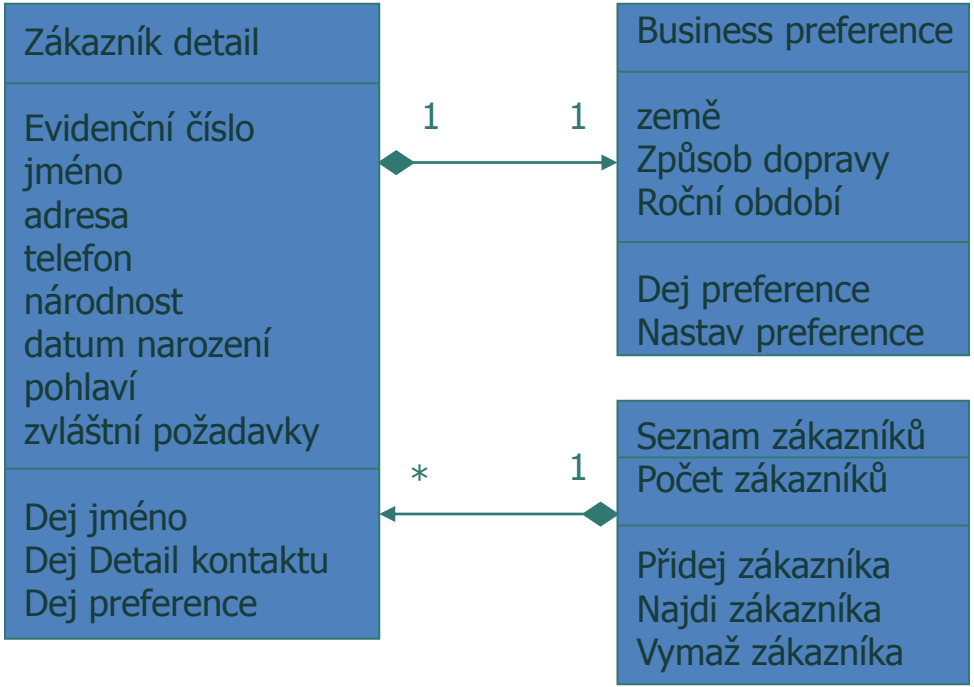
Požadavky zákazníka (requirements) na modul IS oprava elektrospotřebičů:

01. Příjem zakázky na opravu elektrospotřebiče
02. Výdej zakázky majiteli
03. Správa číselníků
04. Monitoring oprav
05. Evidence zákazníků
06. Vyhodnocení oprav
07. Vyřízení reklamace opravy
08. Oprava spotřebiče v servisu

Ke každému požadavku zpracován detailní popis

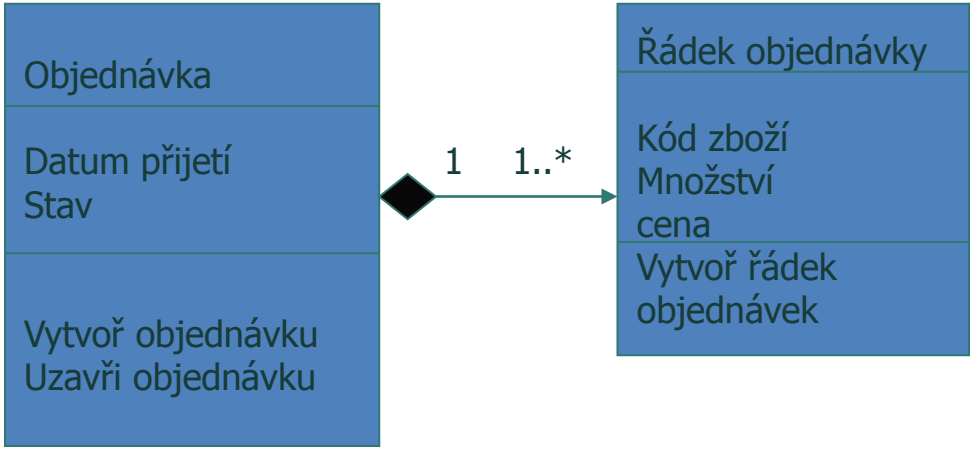
Příklad 1 Vazba typu agregace

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ



Příklad 2 Vazba typu kompozice

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ



Příklad 3 Vazba typu asociace

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

Typy vazeb:

1 k 1

1 k 1..*

1 k *

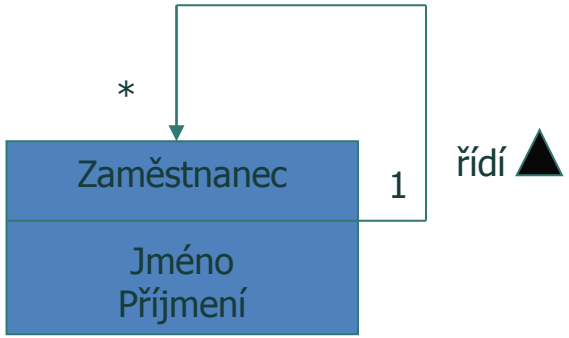
1 k 1..5

1 k 0..1

Příklad 4 Reflexivní asociace

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

Objekt zaměstnanec může mít nula nebo více podřízených.
Řada zaměstnanců nebude mít podřízené, ale každý zaměstnanec bude mít svého nadřízeného



Příklad 5 Generalizace - dědičnost

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

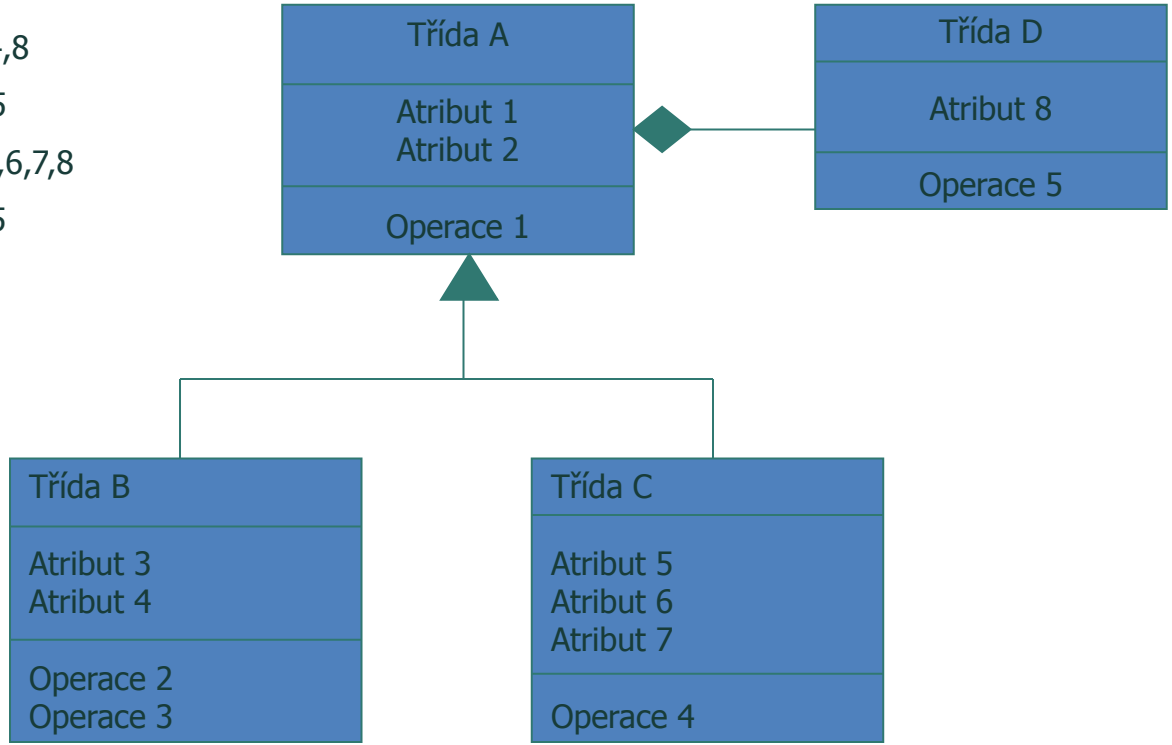
Objektová třída

B atributy 1,2,3,4,8

operace 1,2,3,5

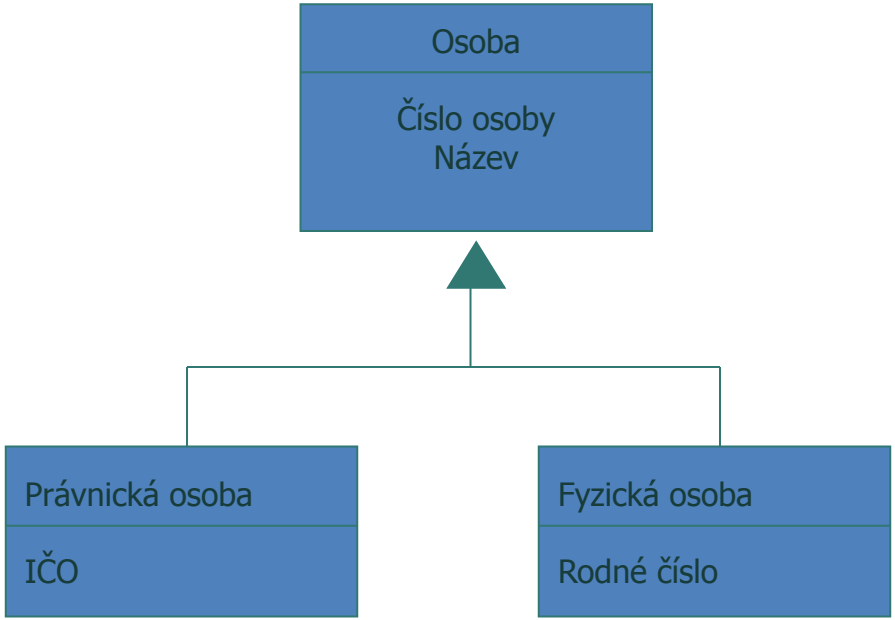
C atributy 1,2,5,6,7,8

operace 1,2,4,5



Příklad 6 Abstraktní třída

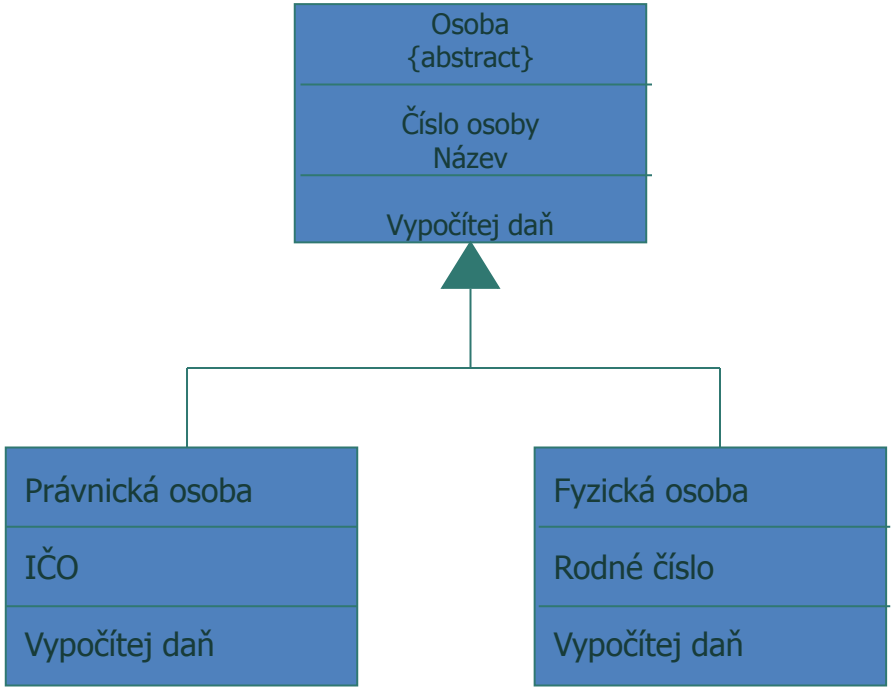
Abstraktní třída je zvláštní třída – ve výjovém prostředí nebude nikdy vytvářena její konkrétní instance



SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

Příklad 7 Polymorfismus objektů

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

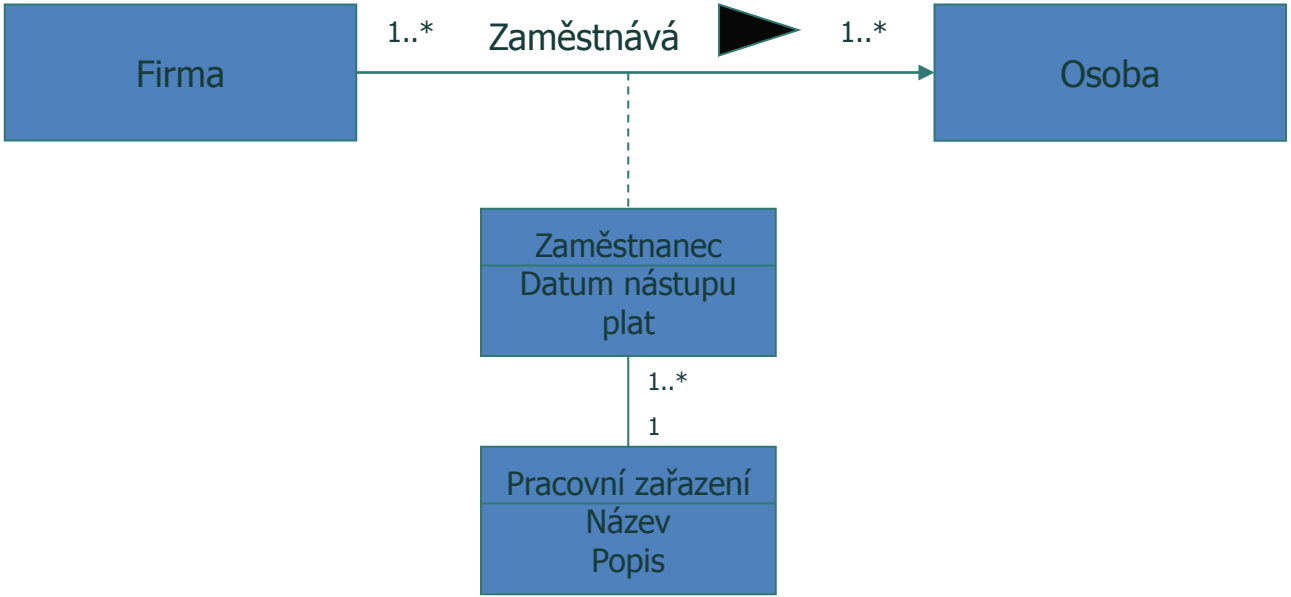


Některé objekty mají totožná rozhraní realizovaná pomocí operací, ale metody, které se skrývají za těmito operacemi jsou rozdílné = polymorfismus.

Příklad 8 Asociační třída

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

– Asociační třídy dovolují přiřadit atributy, operace a další rysy k asociační vazbě, která řeší vztah mezi třídami mnoha ku mnoha.



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

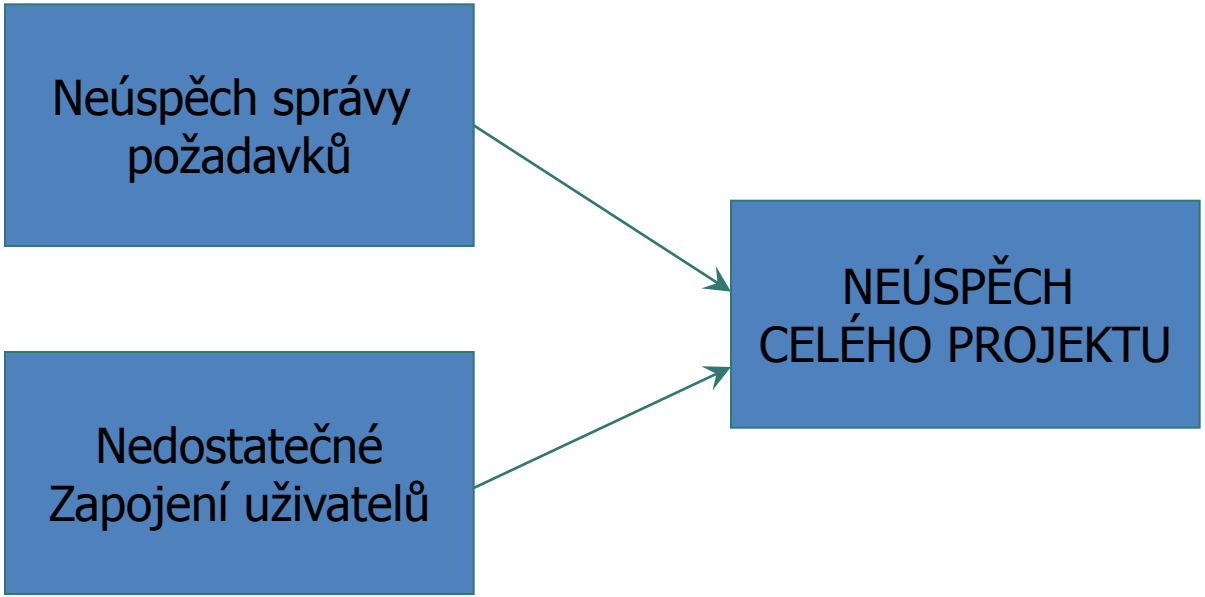
Zdroje požadavků II

- 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
- Atd.

Neúspěch správy požadavků



Úvod do objektového modelování a jazyka UML

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 !
- Případ užití = 1. technika pokrytá jazykem UML.
- Případy užití jsou logickým pokračováním analytických prací vycházejících z uživatelských požadavků
- PŘÍPADOVÁ STUDIE = PŘÍKLAD

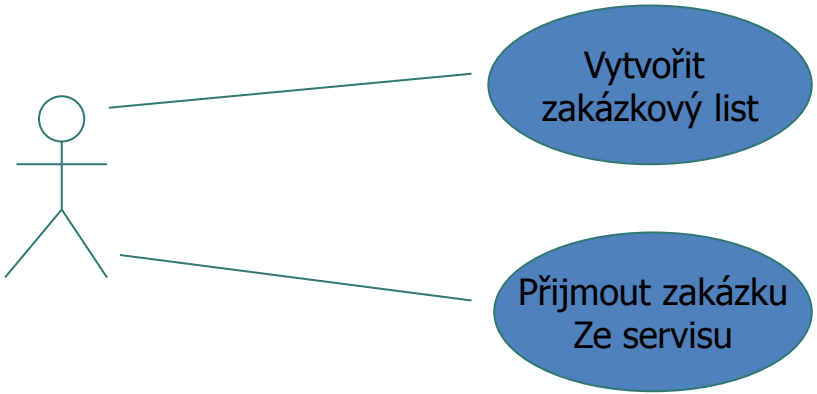
Případy užití - úvod

- Případy užití, typové úlohy, užitné 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

Aktéři

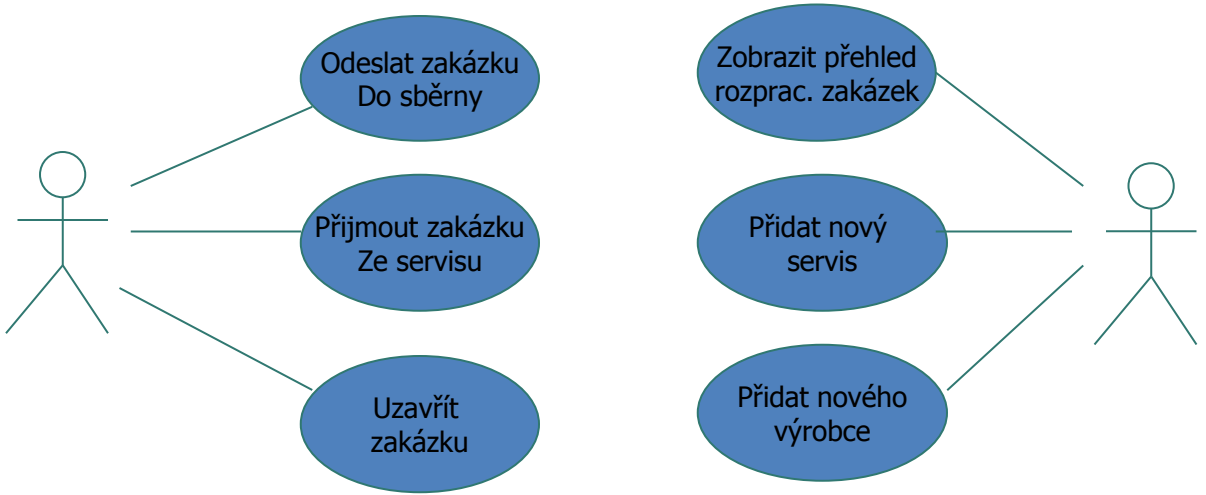
- Aktér = role, ve které vystupuje uživatel v rámci jeho komunikace se systémem

- Př. Aktér= uživatelská role vůči systému



Aktéři

- 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



SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

Scénář - Případy užití: Příjem zakázky do opravy

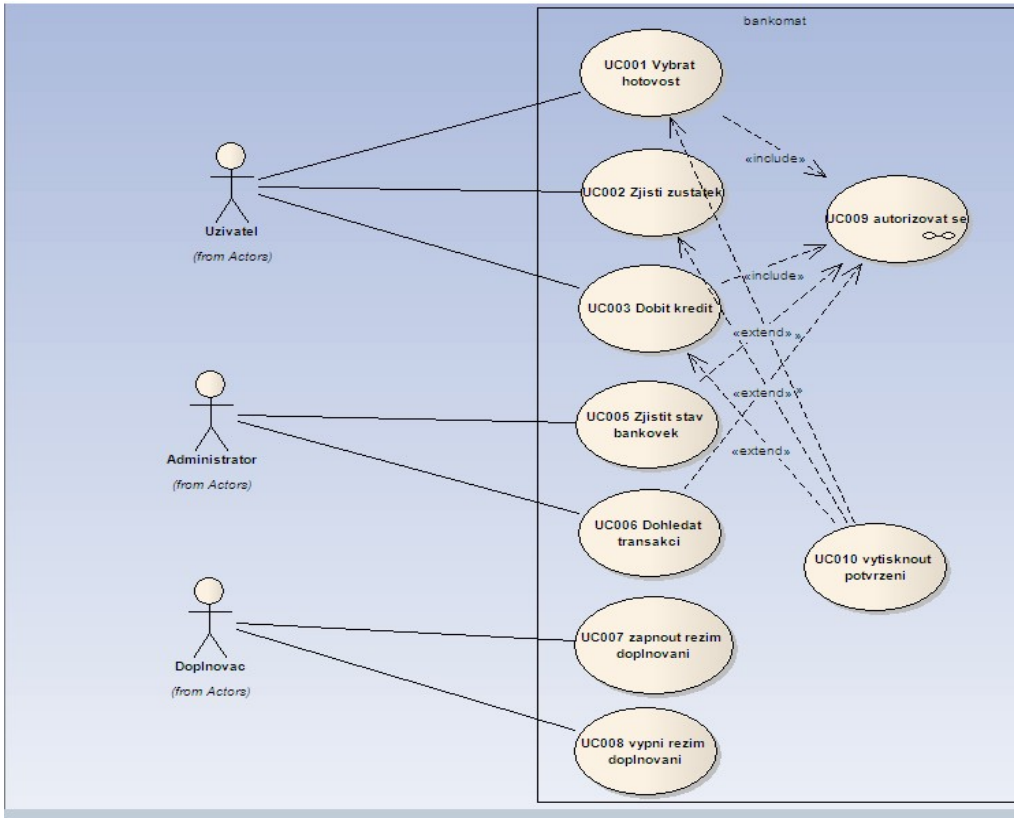
KROK	Role	AKCE
1	Uživatel	spustí volbu Založit zakázku
2	Systém	zobrazit formulář detailu zakázky a zpřístupní údaje pro pořízení
3	Uživatel	pořídí vstupní informace zakázky, jedná se o tyto údaje
4	Uživatel	aktivuje výběr zákazníka z evidence sběrny
5	Systém	zobrazí formulář seznamu zákazníků v abecedním 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 spotř. a návrat k editaci zak. listu
12	Uživatel	zapíše údaj o poruše spotřebiče a dá pokyn k tisku zak. listu
13	Systém	vytiskne zakázkový list a uzavře formulář zakázky

Případ užití je sada scénářů, které spojuje dohromady cíl

Úvod do objektového modelování a jazyka UML

SLEZSKÁ UNIVERZITA
OBCHODNĚ PODNIKATELSKÁ FAKULTA V KARVINÉ

Případy užití: Bankomat



Děkuji za pozornost

Otázky?