

Objektové metody modelování

Základní pojmy objektového modelování,
Use Case, třídy, objekty

Přednáška 2



**SLEZSKÁ
UNIVERZITA**

OBCHODNĚ PODNIKATELSKÁ
FAKULTA V KARVINĚ

RNDr. Zdeněk Franěk, Ph.D.
Ing. Tomáš Barčák, Ph.D.

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:

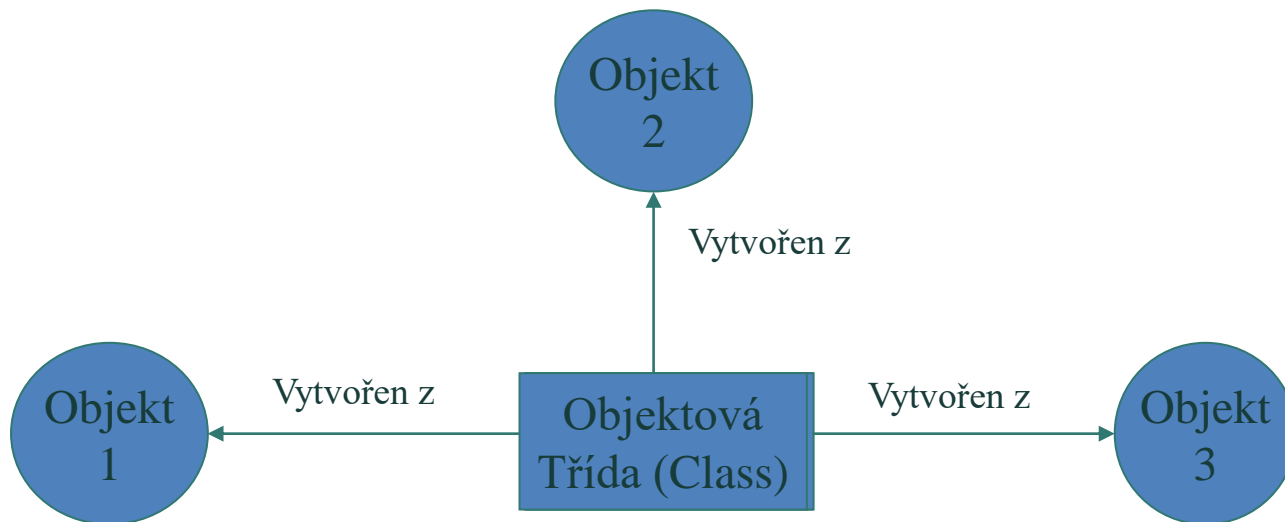
- Třída je to co navrhujeme a programujeme.
- Objekty jsou to, co vytváříme (ze třídy) při běhu aplikace.
- Každý objekt má jiný identifikátor a jiný stav v čase, což znamená jiné hodnoty v jeho proměnných.

Příklad: Třída student v IS/SU:

UČO, Příjmení, Kód stud.programu, ...

Objekt (při běhu aplikace, např. vyhledávání) NOV0002, Novák, INMPOOM, ...

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)

Pozn.: Pojmy string, public, souvisí s programovacími jazyky jako je JAVA, C#, Viz předmět: Úvod do programování, Dr. Perzina KINM.

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, aktualizuj hodnoty, přidej nový záznam, vypočítej stipendium ...)

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 (příslušností), 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** (Pozn. nejčastější vztah)
 - 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**

Požadavky na funkcionalitu systému (**nejsou** předmětem UML)

- 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 mají určitou představu a zkušenost
- proces získávání požadavků od budoucích uživatelů
- různá úroveň uživatelů
- konfrontace s představou tvůrců systému



Zdroje požadavků II

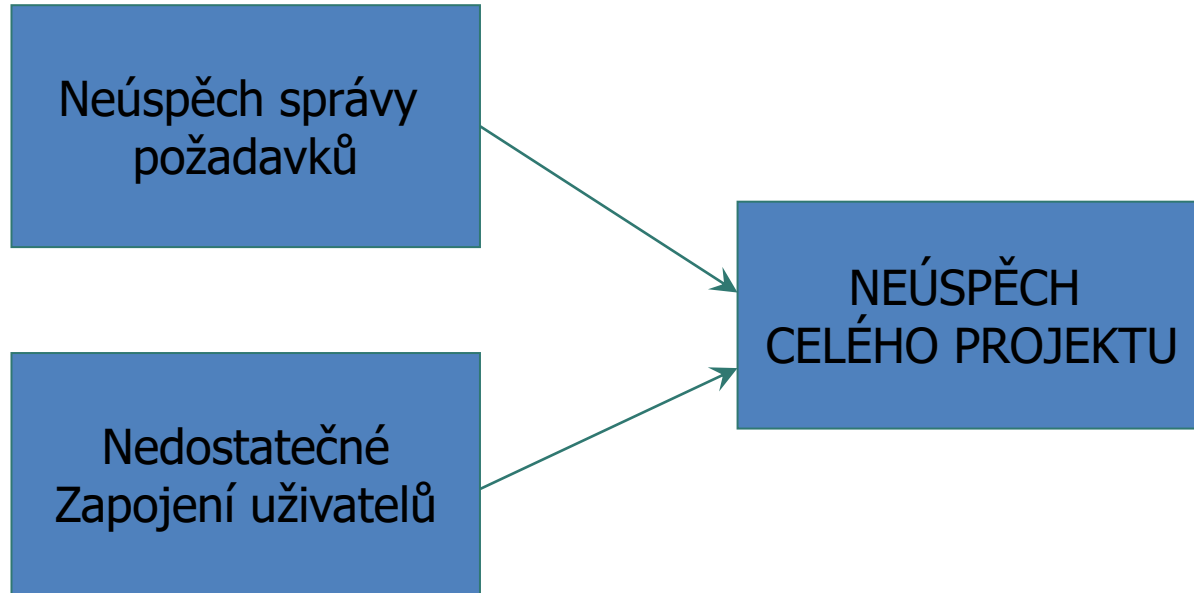
- legislativa
- požadavky zákazníků
- existující systémy uživatelů
- pracovní procesy uživatelů
- vlastní know-how
- prostředí zákazníka
- hardware / software vybavení



Nefunkční požadavky

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

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

Vysvětlení pojmů na případové studii

Modelová situace:

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

Pro naše potřeby modelujeme jeden Modul IS pro zprostředkování oprav.

Vysvětlení pojmů na případové studii

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 je zpracován detailní popis.

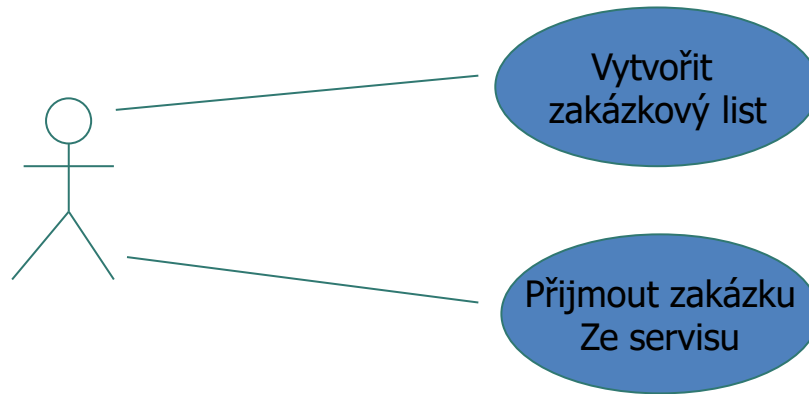
Případy užití - úvod

- 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, alternativní scénář.
- Případ užití je sada scénářů, které spojuje dohromady cíl.

USE CASE I – opakování z 1. semináře



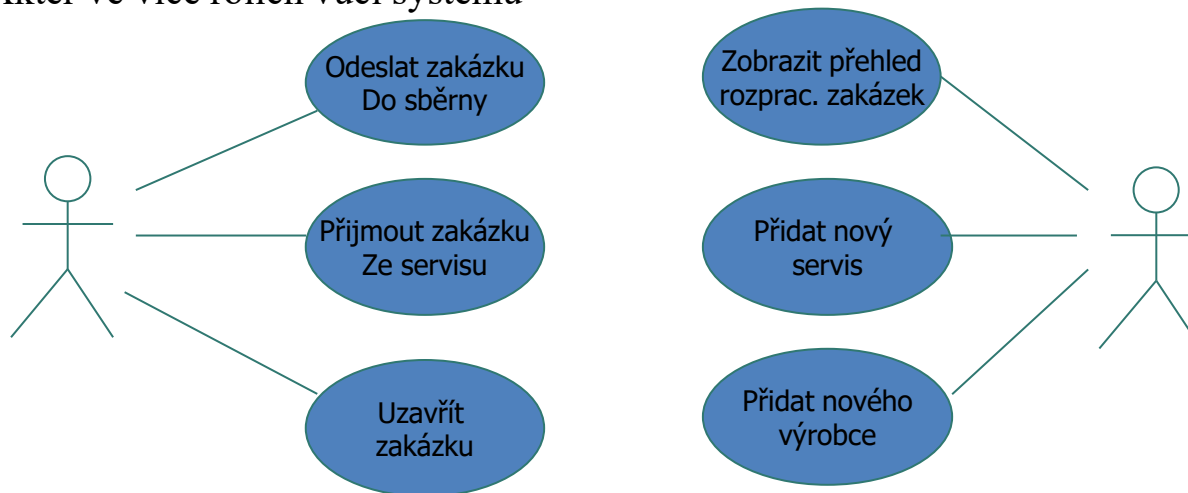
- Aktér = role, ve které vystupuje uživatel v rámci jeho komunikace se systémem
- Př. Aktér= uživatelská role vůči systému



USE CASE 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



Opakování z 1. semináře

Scénář - Případy užití: Vytvořit zakázkový list - Příjem spotřebiče do opravy

KROK	Role	AKCE
• 1	Uživatel	spustí volbu Založit zakázku
• 2	System	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	System	zobrazí formulář seznamu zákazníků v abecedním pořadí
• 6	Uživatel	vybere zákazníka ze zobrazeného seznamu a přiřadí ho k zakázce
• 7	System	zavře formulář seznamu zákazníků a vrátí se do editace zakázky
• 8	Uživatel	aktivuje zobrazení seznamu spotřebičů
• 9	System	zobrazí formulář seznamu spotřebičů v třídění dle názvu spotřeb.
• 10	Uživatel	vybere spotřebič ze seznamu
• 11	System	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	System	vytiskne zakázkový list a uzavře formulář zakázky

Případ užití je sada scénářů, které spojuje dohromady cíl.

USE CASE IV Alternativní scénář



Scénář – Založení nového zákazníka

K bodu 4 předchozího scénáře

4a. Zákazník neexistuje v seznamu zákazníků sběrný, bude založen nový

Krok	Role	Akce
4a1	Uživatel	zvolí nabídku <i>Nový zákazník</i>
4a2	System	zobrazí prázdný formulář zákazníka se zpřístupněnými údaji zákazníka
4a3	Uživatel	pořídí údaje zákazníka: příjmení, jméno, titul, adresa, telefon
4a4	System	založí nového zákazníka a převezme jeho identifikaci do zakázkového listu

- Základní scénář typu „všechno jde hladce“
- Alternativní scénář mimořádné stavy a z nich odvozené větve ze základního scénáře.

USE CASE V - Větvení činnosti ve scénáři



Scénář – Příjem spotřebiče do opravy

Případ užití začíná, když zákazník přináší spotřebič do opravy.

Užití doplňkových konstruktů pro větvení činnosti ve scénáři.

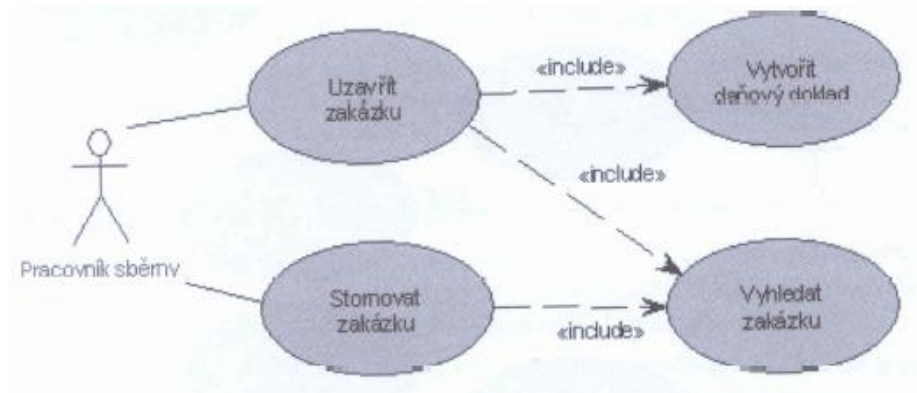
Krok	Role	Akce
1	Obsluha	dá pokyn k založení zakázkového listu
2	System	zobrazí formulář zakázkového listu se zpřístupněnými údaji
3	Obsluha	aktivuje výběr zákazníka ze seznamu existujících
4	System	zobrazí seznam existujících zákazníků sběrný
5	Uživatel	POKUD zákazník dosud neexistuje v evidenci sběrný, pořídí údaje zákazníka... JINAK vybere zákazníka z evidence KONEC-POKUD
6	System	předá údaje zákazníka do zakázkového listu
7

USE CASE VI – Vztahy (relace) mezi případy užití - INCLUDE



INCLUDE = ZAHRNUTÍ

- Příklad relace include mezi případy užití



Případ užití: Uzavřít zakázku

Případ užití začíná, když zákazník přichází vyzvednout opravený spotřebič

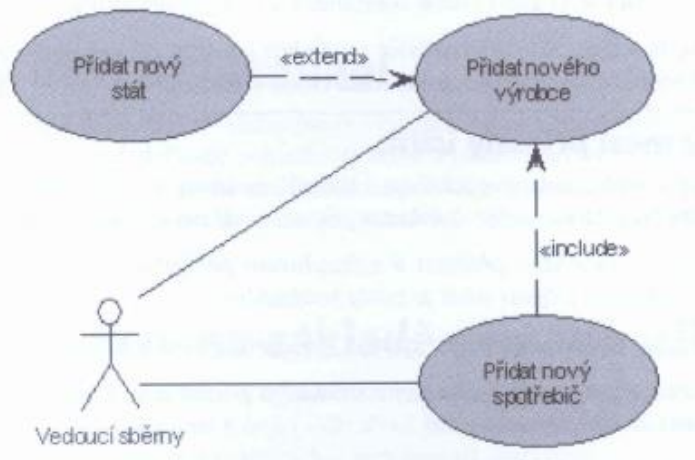
Krok	Role	Akce
1	Obsluha	vyhledá zakázku – viz případ užití Vyhledat zakázku
2	Systém	zobrazí detailní okno zadané zakázky
3	Obsluha	seznámí zákazníka s výsledkem opravy a dá pokyn k sestavení daňového dokladu – viz případ užití Vytvořit daňový doklad
4

USE CASE VII – Vztahy (relace) mezi případy užití - EXTEND



EXTEND = ROZŠÍŘENÍ

- Příklad relace extend mezi případy užití



Případ užití: Přidat nového výrobce

Pracovník sběrný ověří, zda pro spotřebič existuje v dané zemi výrobce, pokud ne založí nový záznam výrobce

Krok	Role	Akce
1	Obsluha	spustí volbu Přidat výrobce
2	System	zobrazí detailní formulář výrobce a zpřístupní údaje pro editaci
3	Obsluha	pořídí údaje spotřebiče – název výrobce a stát (výběrem z číselníku států)
Bod rozšíření: Přidat nový stát		
4	Obsluha	dá pokyn k uložení informací
5

USE CASE VIII Shrnutí a doporučení



Případ užití je sadou scénářů (sekvencí dialogů uživatele se systémem) – základního a event. alternativních, které sledují společný cíl.

Případ užití je vždy iniciován aktérem.

Případ užití vyjadřuje co (ale nikoliv jak) budoucí systém nabídne uživateli.

Případy užití reprezentují vnější pohled na systém, nečekejte proto odhalení vzájemných vztahů mezi případy užití a třídami uvnitř systému.

Pokud možno, provádějte konceptuální modelování přímo s budoucími uživateli systému, pomůže vám to odhalit případy užití.



Děkuji za pozornost

Otázky?
