

Objektové modelování a programování - základní pojmy, podstata, využití. Softwarový proces. UML. Událostmi řízené programování. Architektura MVC.

základní pojmy

- **Program** – posloupnost příkazů popisujících určitou činnost
- **Proces** – prováděný (běžící) program
 - **Sekvenční** – instrukce zpracovávány postupně
 - **Paralelní** - instrukce zpracovávány souběžně
- **Procesor** – zařízení, které dokáže vykonávat příkazy programu
- **Data** – objekty (údaje), s nimiž pracují procesy
- **Zdrojový kód** – kód programu zapsaný v nějakém programovacím jazyce
- **Cílový kód** – binární kód, (vytvořený po překladu zdr. kódu) je spustitelný
- **Bytekód (java)** – spustitelný mezikód, vzniká jako cílový kód, ale je spouštěn a prováděn běhovým prostředím (Runtime Environment, Virtuální stroj)

Objektové programování – používá programovací jazyk k napodobení objektů skutečného světa definováním tříd.

Zásady objektového programování (zdroj: Rudolf Pecinovský)

- Programovat proti rozhraní a ne proti implementaci
- Dbát na důsledné zapouzdření a skrývání implementace
- Zapouzdřit a odpoutat části kódu, které by se mohly měnit
- Maximalizovat soudržnost (cohesion) entit (balíčků, tříd a metod). Každá entita by měla řešit jen jeden konkrétní úkol
- Koncentrovat zodpovědnost za řešení úkolu na jednu entitu – návrh řízený zodpovědností (responsibility driven design)
- Minimalizovat vzájemnou provázanost (coupling) entit
- Vyhýbat se duplicitám kódu

Podstata, využití

základní objektové koncepty

- **Abstrakce** – separování důležitých rysů od nedůležitých v závislosti na kontextu
- **Zapouzdření (encapsulation)** – data a operace objektu tvoří nedělitelný celek
- **Dědičnost (inheritance)** – schopnost objektů dědit vlastnosti a chování předka
- **Polymorfismus** – jev, kdy operace stejného jména je používána pro více objektů odlišných tříd. **Vlastnost, která umožňuje:**
 - jednomu objektu volat jednu metodu s různými parametry (parametrický polymorfismus)
 - objektům odvozených z různých tříd volat tutéž metodu se stejným významem v kontextu jejich třídy, často pomocí rozhraní
 - přetěžování operátorů znamená provedení operace v závislosti na typu operandů (operátorový polymorfismus)
- **Komunikace** (zasílání zpráv = volání metod) – objekty mezi sebou komunikují zasíláním zpráv. Výsledkem přijetí zprávy příjemcem je vykonání nějaké operace

Objektové modelování – snaha reálně popsat existující (nebo vznikající) systém v zjednodušené (abstraktnější) podobě

Význam modelování

- snadné změny s nízkými náklady oproti reálnému systému
- usnadnění komunikace v týmu a se zákazníkem

- přehled o aktuálním stavu projektu
- vytváření dokumentace

Princip tří architektur

- konceptuální úroveň – model reality, nejvyšší abstrakce, popisuje obsah IS, ne formu (CO)
- technologická úroveň – popis technologie s ohledem na prostředí implementace (JAK)
- fyzická úroveň – popis detailů implementace v konkr. prostředí (ČÍM)

Softwarový proces

Životní cyklus informačního systému (IS)

- Analýza firemního prostředí – jak funguje firma
- Analýza IS – co má IS nabízet
- Návrh IS – jak realizovat požadovaný systém
- Implementace IS – realizace – programování
- Nasazení IS – nasazení u zákazníka

Unified Modeling Langue (UML) - grafický jazyk pro specifikaci, vizuální popis, tvorbu a dokumentaci jednotlivých součástí softwarového systému. Jazyk pro OO modelování.

Modelování typových úloh – nástrojem je **Diagram typových úloh** (Use Case Diagram), vyjadřuje vztahy Aktér-úloha a úloha-úloha. Důležité jsou scénáře.

Modelování tříd – nástrojem je **Diagram tříd** (Class Diagram). Jeto základní strukturální diagram UML.

- **Třída** – abstraktní definice množiny objektů (atributy, metody)
- **Rozhraní tříd** – definuje kontrakt, ke kterému se třídy přihlašují
- **Asociace** – definují vztahy mezi objekty

Modelování dynamiky systému – dva základní (vzájemně izomorfní) diagramy **OSD** a **OCD**

- **sekvenční diagram** (Object Sequence Diagram - OSD) – zobrazuje interakci objektů s důrazem na časovou posloupnost, mapován k jedné typové úloze
- **diagram objektové spolupráce** (Object Collaboration Diagram - OCD) – pohled na strukturu spolupráce – vztahy mezi objekty
- **diagram aktivit** – lze použít pro modelování chování, ale není izomorfní s OSD a OCD, modeluje typovou úlohu jako posloupnost aktivit – ne jako interakci uživatele a systému

Událostmi řízené programování (Event-Driven Programming) – událost vzniká buď jako výsledek interakce mezi uživatelem a GUI nebo jako důsledek změny vnitřního stavu aplikace či OS.

Obsluha události – úsek kódu, který je při vzniku události automaticky vyvolán a provádí činnost k události připojenou (Event Handler)

Typy událostí

- klik/dvojklik
- změna stavu komponenty
- stisk/uvolnění klávesy/tl. myši
- překreslení
- událost systému nebo zpráva časovače

Architektura MVC - Model-view-controller (MVC) - je softwarová architektura, která rozděluje **datový model** aplikace, **uživatelské rozhraní** a **řídící logiku** do tří nezávislých komponent tak, že modifikace některé z nich má minimální vliv na ostatní.

- **Model (model)** - doménově specifická reprezentace informací, s nimiž aplikace pracuje

- **View (pohled)** - převádí data reprezentovaná modelem do podoby vhodné k interaktivní prezentaci uživateli
- **Controller (řadič)** - reaguje na události (typicky pocházející od uživatele) a zajišťuje změny v modelu nebo v pohledu

Čisté OOP versus hybridní prog. Jazyky

- **čisté OOP** jazyky jsou jazyky, které nepřipouštějí jiné programovací modely. Funkci nemůžeme napsat samostatně, pokud není součástí třídy. Nemůžeme deklarovat globální proměnnou. Příklady čistých jazyků jsou Smalltalk a Eiffel.
- **S hybridními** jazyky můžeme dělat cokoli chceme včetně úplného vypuštění OOP principů. Příklady hybridních jazyků jsou všechny ty, které jsou kompatibilní s již existujícími, jako například C++ nebo Object Pascal.

Zdroje:

Přednášky PRO1, OMO1, PRO2

<http://cs.wikipedia.org/wiki/Model-view-controller>

[http://cs.wikipedia.org/wiki/Polymorfismus_\(programování\)](http://cs.wikipedia.org/wiki/Polymorfismus_(programování))

<http://www.fi.muni.cz/usr/jkucera/pv109/2002/xkriz1.htm>