

# Objektově-orientované databázové systémy

Z FITwiki

Založeno na materiálech od pmikus

## Základní pojmy

### Data

reprezentace skutečnosti, schopné přenosu, uchování, interpretace, zpracování, hodnoty různých datových typů

### Informace

data, která mají sémantiku

### Znalosti

informace po zařazení do souvislosti

### Systém

množina prvků a vazeb mezi nimi, účelově definované na nějakém nosiči

### Nosič

- množina prvků systému ve vzájemných informačních a procesních vztazích
- prvky nosiče = zdroje
- výsek reálného světa (knihovna, škola, úřad, atd.)

### Zdroje

- fyzické (osoby, materiál, stroje, finance)
- konceptuální (informace)

### Struktura

je uspořádaná n-tice, prvek kartézského součinu, představuje strukturovanou hodnotu vytvořenou pevným počtem pojmenovaných dílčích hodnot obecně různých typů. Pojmenované hodnoty označujeme jako **vlastnosti**.

- žádné omezení datových typů vlastností (data, složené typy i algoritmy)
- definice pouze rozhraní - polymorfismus struktur
- komunikace zasíláním zpráv (vysílající struktura nemusí znát příjemce zprávy - polymorfismus - zpráva může být interpretována různým způsobem v závislosti na typu přijímající struktury)

### Kolekce

je uspořádaná multimnožina. Předem neomezený počet hodnot stejných datových typů

- standardní operace: add, item, count, remove
- kurzor (iterátor) - ukazovátka do kolekce, posouvání oběma směry
- uspořádání nad kolekcí (jedno nebo více podle různých klíčů)
- vlastnosti - nejčastěji agregáty: maximum, minimum, suma, průměr
- tabulka v relační databázi je vlastně kolekcí bez vlastních vlastností a s často omezenou funkcí, jejímiž prvky jsou struktury s vlastnostmi jednoduchých typů
- častý požadavek jedinečnost prvku v kolekci

### Prostá struktura

je struktura bez identifikace.

## Obsah

- 1 Základní pojmy
  - 1.1 Vztahy
- 2 Caché
- 3 Java EE
  - 3.1 Vrstvy
  - 3.2 Java EE aplikace
  - 3.3 Kontejnery
  - 3.4 Technologie

## Objekt

- struktura s identifikací - OID - jednoznačná identifikace objektu v celém systému
- identifikovatelný, odkazovatelný
- perzistentní - dynamická nebo statiská prezistence
- porovnávání objektů:
  - identické objekty/strukтуры - stejné atributy
  - totožné objekty - stejné OID

## Extent

- kolekce obsahující obor hodnot pro objekty jistého typu
- důležitý navigační prvek v OO databázích
- kolekci udržuje OODBMS
  - v případě dědění může být neefektivní (nová instance - vložení do všech extentů hierarchie dědičnosti)
  - uživatel má možnost omezit, které třídy budou tvořit extenty
- je nejcastejším vstupním bodem do objektové databáze (iterujeme přes objekty určitého typu v extentu a z nich se pomocí vztahu dostáváme k dalším informacím)

## abstraktní typ

strukturní, které slouží jen jako stavební kameny (vzory) pro vytváření následníků

Poznámka: Abstraktní typ může mít extent (budou tam vyskytnuty všech následníků toho datového typu)

## konkrétní typ

strukturní, které skutečně budou mít své výskyty a budou skutečnými strukturami

# Vztahy

- dva pohledy na strukturu:
  - definiční - definice metadat (jak má struktura daného typu vypadat z hlediska vlastností a jejich typů)
  - výskytový - vytváříme konkrétní výskyty (instance) struktury (jaké jsou hodnoty vlastností)

## Vztah

je vazba mezi výskyty objektů

- výskyt struktury, v níž se vyskytuje vlastnost typu struktura (je nezajímavé, jde-li o prostou strukturu nebo objekt), nazveme vlastníkem; hodnotu vlastnosti nazveme členem

## Typy vztahů

- Členem je prostá struktura - jedná se o hierarchické členění struktur (základní prostředek pro vytváření komplexních hierarchicky strukturovaných hodnot)
- Členem je kolekce prostých struktur - opět pouze hierarchické zanoření
- Členem je jediný objekt - vztah 1:1 realizovaný pomocí OID
- Členem je kolekce objektů - vztah 1:N, kolekce obsahuje reference na objekty

- Objekty ani kolekce objektů nemohou existovat bez vztahů a naopak.

## Inverzní vztahy

se mohou automaticky vytvořit při vzniku vztahu (atribut inverse), velmi usnadňují vývoj IS (automatické udržování integrity apod.).

## Dědičnost

umožňuje definici typu pomocí typu jiného.

- Pojmy **předek** a **následník**, **přímý předek**, **přímý následník** -> dědičnost, přímá dědičnost
- Vlastnosti se mohou přidávat, modifikovat i ubírat.
- Binární relace dědičnosti je tranzitivní uzávěr relace přímé dědičnosti. Obě tyto binární relace jsou uspořádáním.
- **Generalizace** je postup vytváření hierarchie dědičnosti hledáním společných vlastností a vytvářením předků.
- **Specializace** je opačný postup ke generalizaci, kdy se rozrůžňují typy a vznikají tak následníci.
- **Abstraktní typy** existují pouze kvůli hierarchii dědičnosti, ale nelze vytvářet jejich instance.
- jednoduchá dědičnost - každý následník smí mít pouze jediného předka (zakreslený graf je potom stromem)
- vícenásobné dědičnosti - počet předků není omezen (zakreslený graf je obecný acyklický graf)
- v grafu dědičnosti se nesmí vyskytovat cyklus

## Caché

- Vytvořeno společností InterSystems
- Postrelační databáze (je schopna pracovat jak z relačního tak objektového pohledu)
- Aplikační i datový server + integrované vývojové prostředí
- Přístup k objektům přes ID, pojmenovaný dotaz, Ad-hoc dotaz

### Architektura databáze

- základní jednotkou modelování jsou objekty organizované do tříd
- Caché Class Dictionary - úložiště tříd a dat
- každá třída jsou dvě synchronní sady kódu - pro relační i objektový přístup
- Vlastnosti jsou do relačního modelu mapovány
  - jednohodnotové vlastnosti - sloupce
  - vícehodnotové - zvlášť tabulka s vazbou na tabulku objektů třídy
- třídy lze tvořit pomocí
  - Caché studia (ručně)
  - Pomocí DDL SQL
  - Programově pomocí objektů
  - Pomocí XML
  - pomocí UML (import UML specifikací)

### Typy tříd

- Abstraktní třídy - nelze instanciovat
- Registrované třídy - lze instanciovat, nejsou perzistentní
- Persistentní třídy - lze instanciovat jsou perzistentní
- Vnořené třídy - nejsou perzistentní samy o sobě, ale jen jako součást perzistentní třídy
- Odvozené třídy - vznikají dědičností, polymorfismem
- Datové typy

### Scriptování

- Caché Object Script
  - značně odlišné od klasických programovacích jazyků
  - významné mezery
  - 2 typy proměnných - globální (perzistentní) a dočasné
  - 2 typy funkcí - vnitřní (systémové) a vnější (uživatelské)
- Caché Basic
  - Vychází z VisualBasic
- obojí je možné kombinovat

### CSP - "Caché server pages"

- Dynamicky generovaná stránka, session automaticky
- Podobné JSP

# Java EE

- Platforma pro vývoj a provoz aplikací (virtuální stroj, podpůrné nástroje)
- Databázová vrstva, Business vrstva (chování), Webová vrstva, Klientska vrstva

## Vrstvy

1. Databazova
2. **Business**
  - Implementace chovani aplikace
  - Potencionalne distribuovana
3. **Webova**
4. Klientska
  - Aplikace nebo prohlizec

*Tucne vrstvy spadaji do kompetence Java EE serveru*

- datová vrstva
  1. Java Beans
  2. Java Persistence API (JPA)
- webová aplikace
  1. Java Server Pages (JSP)
  2. JSP Standard Tag Library (JSTL)
  3. Java Server Faces (JSF)

## Java EE aplikace

EJB (Enterprise Java Beans) moduly

- definují veřejná rozhraní
- implementují chování
- lze je odděleně nasadit na servery
- EJB vrstva zajišťuje vzdálené volání funkcí rozhraní

Webove moduly

- uzivatelske rozhrani

## Kontejnery

- Prostředí pro běh aplikace na serveru (webový, EJB)
- Webovy kontejner - beh webu
- EJB kontejner - beh EJB modulu, volani funkci
- Java EE kontejner - Webovy + EJB kontejer

Dostupne kontejnery

- Java EE kontejnery
  - GlassFish (Sun)
  - JBoss (Red Hat)
  - Geronimo (Apache)
  - WebSphere (IBM)
- Pouze webove servery

- Tomcat (Apache)
- Jetty (Mort Bay Consulting)

## Technologie

- **Java Beans** - pomocí anotací vytvoření entity persistence z třídy
- **Java Persistence API** - O/R framework, relační úroveň (JDBC spojen s databází), objektově orientované API, mapování tříd na tabulky (detaily anotacemi)
- **Java Servlet** - implementuje chování serveru
- **Java Server Pages** - dynamické webové stránky (XHTML kód, kód v Javě (scriptlet), definované značky, výrazy)
- **JSP Standard Tag Library** - množina základních značek
- **Facelets** - nástupce JSP v Java EE 6, založeno na XML, šablonování komponent a stránek, překlad na interní objektovou reprezentaci
- **Java Server Faces** - MVC framework nad Facelets, značky pro generování základních prvků uživatelského rozhraní

### Zpracování JSP stránky

- stránka se překládá na servlet (třídu)
- překlad zajišťuje kontejner

Citováno z „[http://wiki.fituska.eu/index.php?title=Objektov%C4%9B-orientovan%C3%A9\\_datab%C3%A1zov%C3%A9\\_syst%C3%A9my&oldid=12994](http://wiki.fituska.eu/index.php?title=Objektov%C4%9B-orientovan%C3%A9_datab%C3%A1zov%C3%A9_syst%C3%A9my&oldid=12994)“

Kategorie: Státnice 2011 | Pokročilé informační systémy

---

- Stránka byla naposledy editována 22. 6. 2015 v 09:15.