

**Západočeská univerzita**  
**FAKULTA APLIKOVANÝCH VĚD**

Z Á P A D O Č E S K Á  
U N I V E R Z I T A



Okruhy otázek ke státní závěrečné zkoušce z předmětu  
Databázové technologie (DB)

Databázové systémy 1 (DB1)

Databázové systémy 2 (DB2)

**Případové studie databázových systémů (PSDS)**

Studijní program:	3902	Inženýrská informatika
Obor:	2612T025	Informatika a výpočetní technika – Softwarové inženýrství
	3902T031	Softwarové inženýrství
Akademický rok:	2005/2006	

## Obsah

PSDS (Případové studie databázových systémů) .....	<b>Chyba! Záznam není definován.</b>
Metodika návrhu a realizace informačního systému (IS) – strukturální a objektová analýza .....	3
Systémový přístup .....	3
Modelování (modelová tvorba) .....	4
Modely .....	5
Strukturovaná analýza .....	6
Objektově orientovaná analýza .....	6
CASE systémy, klasifikace, vlastnosti, použití .....	8
Computer Aided Software (System) Engineering .....	8
Dělení CASE systémů dle podpory .....	8
Dělení CASE systémů dle rozsahu možností .....	8
Vlastnosti CASE systémů .....	9
KLADNÉ .....	9
ZÁPORNÉ .....	9
Použití CASE systémů .....	9
Příklady CASE systémů .....	9
Architektura klient - server .....	10

# 1 Metodika návrhu a realizace informačního systému (IS) – strukturální a objektová analýza

---

Předmětem této otázky je popsat základní proces návrhu a realizace informačního systému. Jedná se o rozsáhlou otázku, ve které je možné zmínit různé modely vývoje systému a jejich fáze. Strukturální analýza je prakticky celá látka předmětu KIV/SAI. Objektová analýza je prakticky celá látka předmětu KIV/ASWI.

---

**Informační systémy (IS)** jsou systémy pro sběr, udržování, zpracování a poskytování informací a dat. Příkladem informačního systému může být kartotéka, telefonní seznam, kniha došlé pošty nebo účetnictví. Systém nemusí být nutně automatizovaný pomocí počítačů a může být i v papírové formě.

Při tvorbě informačního systému se zabýváme systémovou analýzou (systémová=určitý postup) a projektováním informačního systému.

Používané metody:

- **systémový přístup**
- **modelování**
- **projektování**

Při řešení problému pomocí systémové analýzy se zabýváme následujícími dvěma kroky:

**1. analýza problému** (systémová analýza – systems analysis)

Výsledkem tohoto kroku je návrh řešení, jeho časového horizontu, požadavky na finance

**2. návrh řešení problému** (projekt systému – systems design)

V tomto bodě se zabýváme transformací zvoleného řešení do navrhovaného informačního systému. Výsledkem tohoto kroku je volba hardwaru, návrh způsobu reprezentace dat v počítači, volba (nebo vlastní návrh) softwaru a řešení komunikace uživatele se systémem (uživatelské rozhraní).

## Systémový přístup

- obecná metoda vědeckého myšlení, jejíž podstatou je analýza fungování složitých celků

v důsledku jejich struktury

### klasický newtonský (mechanistický) přístup

- poznávání celku jeho rozdělením na části a studiem jejich vlastností
- vztahy mezi částmi se neuvažují

### systémový přístup

- poznávání celku prostřednictvím vztahů mezi jeho částmi
- celek může mít vlastnosti nevyplývající přímo z vlastností jeho částí

**postup:**

**1. vymezení hlediska zkoumání, stanovení cíle**

odlišení daného systému od jiných systémů, jež lze na objektu definovat

**2. vymezení hranic systému, zahrnutí prvků a procesů**

odlišení daného systému od okolí – určení hranice systému  
rozhodnutí, které entity zahrnout (či nezahrnout) do systému – seznam prvků  
a procesů systému

### **3. proces strukturování**

rozhodnutí, jak uspořádat vlastnosti množiny zvolených entit (prvků systému)  
definování vztahů prvků a procesů  
rozlišení podstatných a nepodstatných prvků a vztahů

## **Modelování (modelová tvorba)**

- vychází z obecných zákonitostí teorie podobnosti, zejména z principu analogie.

### **model**

- zjednodušené nebo zobecněné zobrazení systému, zavedeného na objektu, které se s tímto objektem shoduje v podstatných vlastnostech
- analogicky (schéma, struktura, znakový systém) určené části přírodní nebo sociální reality jakožto originálu; tento model (analog) slouží k hlubšímu poznání originálu, jeho konstrukce a organizace, ale i přeměn a jejich podmínek

### **využití (funkce) modelu:**

- nahrazuje subjektu pochopení originálu nebo manipulaci s originálem

#### **a) poznávací**

studium struktury původního objektu (např. prostřednictvím verbálního popisu jinak nedostupného objektu)

vzor / plán / návrh (budoucího objektu nebo procesu)

#### **b) manipulační, ověřovací**

myšlenkové nebo materiální experimenty

simulace

ověření vlastností skutečných objektů (např. stability konstrukce)

#### **c) komunikační, dokumentační**

### **základní typy modelů**

- statické (struktura), dynamické (chování)
- konceptuální (nezávislý na implementaci), technologický (pro konkrétní prostředí)
- textový, matematický, číselný, obrazový (2D, 3D, 4D)

### **Požadavky na model**

- úplnost
- přesnost
- minimální redundance
- jednoduchost, čitelnost (7±2)
- konzistence
- hierarchie úrovní
- poměr grafika (většina) – text (menšina)

### **Systémová metodologie a typy vytvářených modelů**

**technika:** popis operací při řešení problému (*co dělat – pomůcka, nástroj*)

**metodika:** postup, jak zvolit operace vhodné k řešení problému (*jak to dělat – návod*)

## Modely (databázové)

### 1. Konceptuální model reality

alternativní názvy: esenciální, pojmový, sémantický, věcný model (schéma, diagram)

- má vyjádřit esenci – podstatu systému
- říká, co musí systém dělat, aby zajistil uživatelské požadavky
- implementačně nezávislý
- jednotný centrální popis různých informačních obsahů, které mohou být v datových základnách
- věcně orientovaný – obsahuje sémantický model dat – věcný obsah (sémantika) databáze
- vymezuje, **co** budeme v datové základně sledovat, aniž obsahuje údaje o tom, **jak**

budeme tyto předměty v datové základně realizovat  
slouží jako společný základ pro:

- chápání světa objektů uživateli, projektanty, správce databáze apod.
- interpretaci uživatelských pohledů a návrh implementace
- zobrazení mezi uživatelskými pohledy a fyzickým uložením
- definici pravidel vývoje a manipulaci v informační bázi

**využití:** údaje potřebné pro specifikaci zadání úlohy nebo pro komunikaci s uživatelem

### 2. Logický model reality (logická struktura dat)

- určuje, jak je konceptuální struktura dat implementována v konkrétním technickoprogramovém prostředí

logické (databázové) modely: lineární, hierarchický, síťový, relační, objektově orientovaný

**využití:** údaje potřebné pro projektanta při algoritmizaci a programování

### 3. Fyzická struktura dat

- model fyzického uspořádání dat (vyjadřuje jejich uložení na konkrétním médiu)

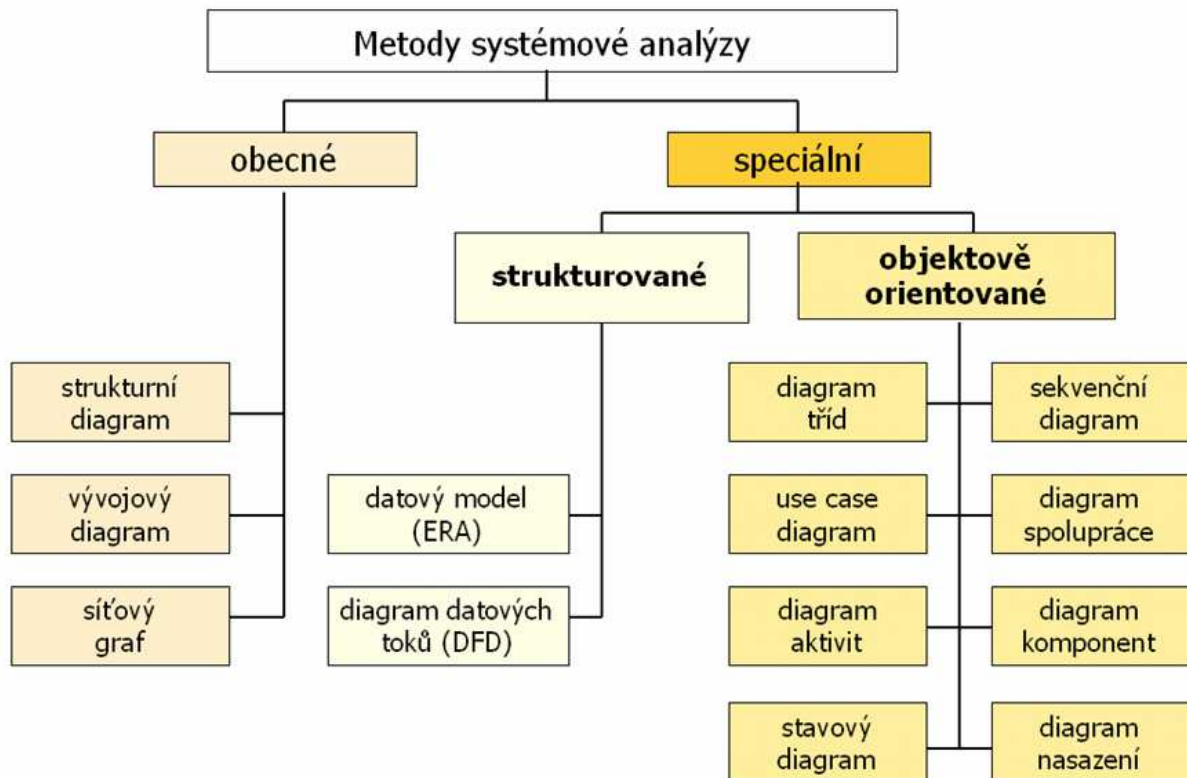
#### Cíle návrhu informačního systému:

- mít v systému všechna potřebná data
- nemít v systému žádná nepotřebná data
- vyjádřit vztahy mezi daty
- popsat transformaci dat v systému

#### Obecné metody systémové analýzy

Mezi tyto metody patří nejrůznější typy grafů jako například síťové grafy. Dále je to statický popis systému. Dále jde o funkční popis systému, např. vývojové diagramy (flowchart – jednoduché diagramy z PPA1), byznys modelování (model podnikových

procesů – grafy typu těch, které jsou v Microsoft BizTalk), workflow management (počítačová podpora podnikových procesů), Ganttovy diagramy, síťové diagramy CPM(Critical Path Method-Metoda kritické cesty)/PERT(The Project Evaluation and Review Technique – metoda vyhodnocování a sledování projektu)



## Strukturovaná analýza

**Strukturovanou analýzou se zabývá předmět KIV/SAI.**

Patří sem hlavně ERA diagramy (Entity-Relationship-Attribute diagram - datové zobrazení systému) a DFD (Data Flow Diagram – funkční zobrazení systému).

**Nejznámější metodologie strukturované analýzy:**

YSM – Yourdonova strukturovaná metodika (Yourdon Structured Method)

SSADM structured systems analysis and design methodology

## Objektově orientovaná analýza

**Objektově orientovanou analýzou se zabývá předmět KIV/ASWI.**

Patří sem hlavně UML a metodiky s ním spojené (RUP, Rational Rose). Předmětem zkoumání je objekt. Tento termín byl poprvé použit v 60. letech minulého století v jazyce Simula. Objekt je cokoli reálného či abstraktního (věc, entita) s jasně vymezenou rolí v daném kontextu, o čemž uchováváma údaje a metody, které s těmito daty manipulují. Lze zmínit obecné principy OOP (dědičnost, polymorfismus, ...).

## **Nejznámější metodologie objektově orientované analýzy:**

OOATool Object-Oriented Analysis Tool (P. Coad, Edward Yourdon)

RUP Rational Unified Process

## **UML (Unified Modeling Language)**

Standardizovaný jazyk pro tvorbu diagramů v objektově orientovaných modelech (analýze a návrhu) informačních systémů. Není vázán na konkrétní metodiku.

Základním účelem UML je umožnit a usnadnit komunikaci (v týmu projektantů, mezi projektantem a zadavatelem).

Model lze zobrazit více diagramy, což zajišťuje více pohledů na tentýž systém.

### **Diagram tříd**

model statické struktury systému

### **Use Case diagram**

model funkcionality systému

### **Sekvenční diagram**

model časové dynamiky uvnitř Use Case

### **Diagram aktivit**

model dynamiky jednotlivých Use Case a operací v třídách

### **Diagram spolupráce**

model interakční dynamiky uvnitř Use Case

### **Stavový diagram**

model životního cyklu objektu

### **Diagram komponent**

model komponent a jejich spolupráce

### **Diagram nasazení**

model rozložení komponent při běhu systému

## 2 CASE systémy, klasifikace, vlastnosti, použití

---

### Computer Aided Software (System) Engineering (CASE)

Modely softwarových systémů jsou příliš složité

- nutná podpora různých úrovní abstrakce, různých pohledů
- nutnost rozdělení mezi jednotlivé vývojáře

CASE nástroje

- slouží pro standardizaci používaných postupů
- nástroj na podporu práce analytiků a programátorů při vývoji informačních systémů, zejména ve fázi analýzy a návrhu – tvorba modelů
- mezistupeň mezi analýzou problému a programováním
- označení pro integrovanou tvorbu programových aplikací pomocí programových prostředků s minimální potřebou manuálního psaní zdrojového kódu programu

**obsah:** databáze (repository, systémová encyklopedie) navrhovaných prvků informačního systému

**funkce:** podpora realizace projektu informačního systému

**základ:** metodika = návod na vytváření modelů a určení závislostí mezi nimi

### Dělení CASE systémů

**Nižší CASE – podpora tvorby software**

- návrhy obrazovek, formulářů, menu
- jazyková podpora

**Vyšší CASE – podpora analýzy a návrhu**

- tvorba diagramů a modelů
- kontrola konzistence modelu

*Příklad:* AxiomSys: strukturovaná analýza

**Integrované CASE – podpora živ. cyklu**

- od analýzy po generování kódu
- round-trip engineering
- podpora tvorby dokumentace

*Příklad:* Oracle Designer

**Komponentové CASE – otevřenost**

- repository s rozhraním (SCM, testování)
- integrace nástrojů od různých výrobců

*Příklad:* Rational Suite Enterprise

### Dělení CASE systémů dle rozsahu možností

**Jedna fáze**

- podpora jedné fáze ŽC (analýza, prog.)

**Jedna metodika**

- podpora dané metodiky přes ŽC

**Více fází, více metodik**

- transformace modelů, vlastní metodiky

**Vývoj + management**

- včetně podpůrných funkcí pro řízení



## Vlastnosti CASE systémů

### KLADNÉ

#### Zvýšení produktivity

- automatizace prací - čas na podstatné věci
- lepší přehled o modelu a implementaci
- podpora rozdělení práce
- snazší údržba dokumentace i systému

#### Zvýšení kvality

- podpora analýzy - lepší znalost požadavků
- kontroly konzistence a úplnosti
- synchronizace reality a dokumentace
- podpora používání standardů

### ZÁPORNÉ

#### Cena

- CASE jsou drahé (řádově 100000 Kč)
- vybírat s rozvahou (reklamní slogany vs. skutečné možnosti vs. skutečné potřeby)

#### Doba návratnosti investice

- na počátku potřebné zaškolení a zaučení
- přínosy obvykle až od 2-3 projektu

#### Změna stylu práce

- nástroj bez používání metodiky k ničemu
- nutnost podřídit se CASE (techniky, metoda)
- podpora managementu nutná

## Použití CASE systémů

- automatizovaná evidence vytvořených objektů a specifikací, dokumentace vývoje systému
- grafická podpora modelování (notace)
- kontrola správnosti modelů podle zvolené metodiky, zajištění integrity a konzistence návrhu (předem definovaná integritní omezení a jejich kontrola, automatické uplatnění změn vytvořených v jedné části ve všech souvisejících částech návrhu)
- podpora týmové práce (identifikace osob a týmů zodpovědných za jednotlivé modely, tvorba více modelů současně, současná práce více osob na jednom modelu)
- správa verzí
- automatický převod definovaných modelů do konkrétního logického a fyzického návrhu (generování programu, popisu databáze, příp. prototypu)
- reverse engineering – zpětné generování konceptuálního modelu z existující aplikace

## Příklady CASE systémů

Powerdesigner (Sybase), Together (Borland), Oracle designer (Oracle), SELECT (LBMS), Rational Rose (Rational Software Corporation)

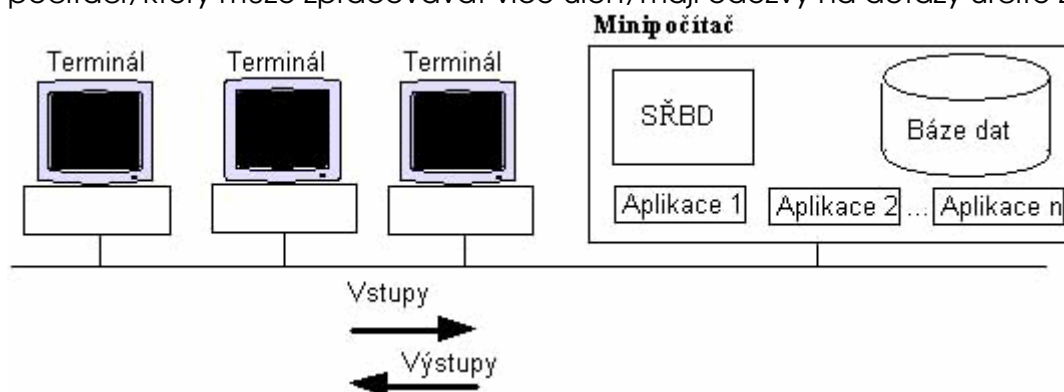
### 3 Architektura klient – server

otázka je naprosto stejná jako v DB2

## Architektury databázových systémů

### Centrální architektura

V této architektuře jsou data i SŘBD v centrálním počítači. Tato architektura je typická pro terminálovou síť, kdy se po síti přenáší vstupní údaje z terminálu na centrální počítač do příslušné aplikace, výstupy z této aplikace se přenáší na terminál. Protože aplikační program i vlastní zpracování probíhá na centrálním počítači, který může zpracovávat více úloh, mají odezvy na dotazy určité zpoždění.

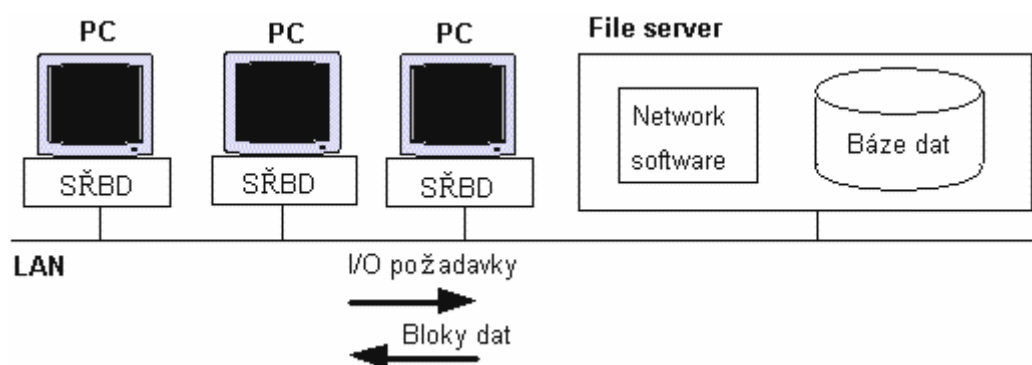


### Architektura file-server

Tato metoda souvisí zejména s rozšířením osobních počítačů a sítí LAN. SŘBD a příslušné databázové aplikace jsou provozovány na jednotlivých počítačích, data jsou umístěna na file-serveru a mohou být sdílena. Aby nedocházelo ke kolizím při přístupu více uživatelů k jedné datům, musí SŘBD používat vhodný systém zamykání (položek nebo celých tabulek).

Komunikace uživatele se systémem probíhá následujícím způsobem:

- ✓ uživatel zadá dotaz
- ✓ SŘBD přijme dotaz, zasílá požadavky na data file-serveru
- ✓ file-server posílá bloky dat na lokální počítač, kde jsou data zpracovávána podle zadaného dotazu (vyhledávání, seřazení atd.)
- ✓ výsledek dotazu se zobrazí se na obrazovce osobního počítače.

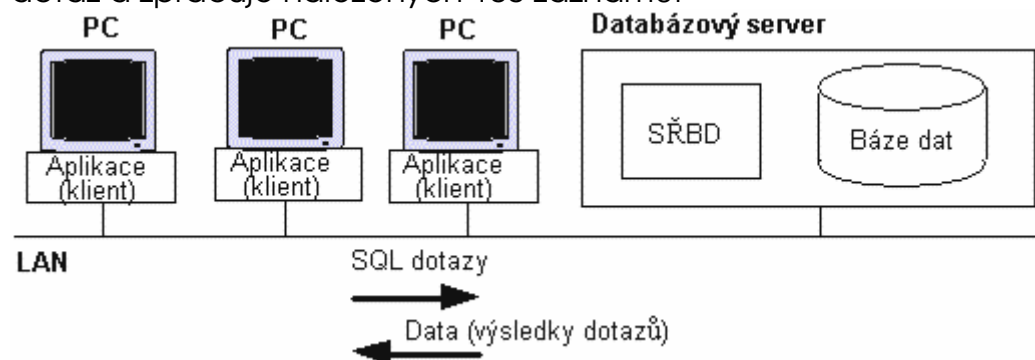


>>Architektura Klient-Server <<

V podstatě je založena na lokální síti (LAN), personálních počítačích a databázovém serveru. Na personálních počítačích běží program podporující např. vstup dat, formulaci dotazu atd. Dotaz se dále předává pomocí jazyka SQL (Structured Query Language) na databázový server, který jej vykoná a vrátí výsledky zpět na personální počítač. Databázový server je tedy nejvíce zatíženým prvkem systému a musí být tvořen dostatečně výkonným počítačem. Celá komunikace probíhá tímto způsobem:

- ✓ uživatel zadává dotaz (buď přímo v SQL nebo musí být do tohoto jazyka přeložen),
- ✓ dotaz je odeslán na databázový server,
- ✓ databázový server vykoná dotaz,
- ✓ výsledek dotazu je poslán zpět na vysílací počítač, kde je zobrazen.

Architektura klient-server redukuje přenos dat po síti, protože dotazy jsou prováděny přímo na databázovém serveru a na personální počítač jsou posílány pouze výsledky. Např. pokud je mezi 10 000 záznamy pouze 100 záznamů, které splňují podmínku dotazu, pak na personální počítač putuje pouze těchto 100 záznamů. V případě architektury file-server je však nutné poslat všech 10 000 záznamů na personální počítač, tam se teprve provede dotaz a zpracuje nalezených 100 záznamů.



### Další charakteristiky

Rozdělení aplikace na klientskou a serverovou část

GUI – může být rozdílné na různých klientech

Jednoduchý přenos dat mezi aplikacemi

Metodika vývoje produktu – často přírůstková – postup: rozdělení na části klient a server, implementace části server, implementace části klient, testování a instalace přírůstků

### Architektura klient – server

prvky architektury: hardwarová platforma, operační systémy, databázový systém, vývojové prostředí, standardy architektury

architektura klient-server znamená dekompozici funkcionality a tedy ideální model např. pro online zpracování transakcí (OLTP) v distribuovaném prostředí

síla architektury: pružnější rozdělení práce (klientům lze zpřístupnit i více serverů), aplikace běží na levnějších zařízeních, klientem může být oblíbený prezentační software (PowerBuilder, Excel), standardizovaný přístup pomocí SQL, centralizace dat – lepší a účinnější ochrana dat

### Databázový server

dedikovaný=slouží pouze jistému účelu

databázový server = středně velký 4 procesory, velký 8 procesorů

databázově - aplikační server = super server – 16 procesorů  
aplikační server = několik hodně dobrých procesorů

### Mapování tří vrstev

1. serverová data – DBMS schéma – sdílené entity >serverové funkce (triggery, uložené procedury)
2. uživatelské objekty >přemístitelné funkce
3. okna >klientské funkce > (klientská reprezentace)

Mapování datového modelu je spojeno se třemi aspekty – rozdělení dat, duplicita dat a realizace integritních omezení

**Problém** – distribuce dat mezi několik serverů

**Východisko** – výhodnější je udržování duplicitních dat – je nutno procesně zajistit konzistenci duplicitních údajů

**Problém** – integritní omezení (zajištění referenční integrity, kardinality vazeb, doménové integrity, atd.)

**Východiska** –

mohou být mapována na server jako součást jazyka DDL

na server jako součást databázového systému ve formě uložených procedur

na klientovi jako součást zpracování (tento způsob implementace je obtížně udržovatelný)

### Model uživatelských objektů

Funkčnost systémů – akce s objekty

Většina systémů klient-server zatím nepoužívá objektovou technologii

Uživatelské objekty mohou být mapovány na server, na klienta nebo jako přemístitelný model

Server: objekty implementovány pomocí uložených procedur

Klient: prostřednictvím popisu v jazyce 4GL (SAL) nebo DLL knihoven

Pravidlo: čím větší část aplikačního zpracování je obecná a realizovatelná na serveru, tím menší změny je nutno provádět ve zpracování klienta při modifikaci uživatelských úkolů

---

Zpracoval: Jan Kupka ([kupka.jan@centrum.cz](mailto:kupka.jan@centrum.cz) – ICQ:136594891)

Verze: 1

Zpracováno dle materiálů z internetu a několika knih

Upozornění:

K tomuto předmětu nejsou dostupné žádné materiály, proto nikdo neručí za to, že zpracované otázky pokrývají to, co měl jejich autor na mysli.

Tento dokument byl napsán v Microsoft Office 2007, čímž se omlouvám za nové formátování. Pokud to někdo chce předělat na starý formát, má moje svolení, já se s tím dělat nebudu ☺.

---