PA165: Úvod do Java EE

Petr Adámek

Obsah přednášky

- Organizace předmětu
  - Formy výuky
  - Hodnocení
  - Osnova
- Java EE aplikace
- Architektury Java EE aplikací
- Technologie Java EE
- Základní koncepty

ORGANIZACE PŘEDMĚTU

Formy výuky

- Přednáška
  - Doporučená účast
- Cvičení
  - Povinná účast
  - Příklady k probírané látce
  - Konzultace k projektům
- Projekt
  - Řešený ve čtyřčlenných týmech
  - Kontrolní body
  - Průběžná práce během semestru

Hodnocení

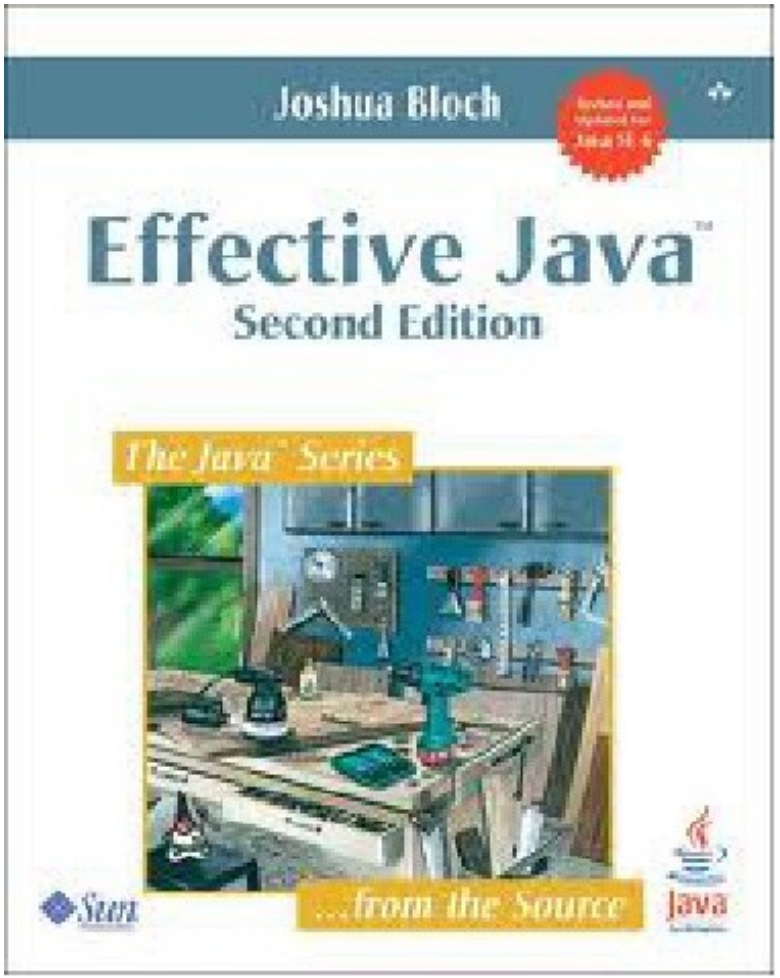
- Projekt: 70 bodů
  - Kontrolní body: 4x10 bodů
  - Obhajoba: 30 bodů
- Závěrečná zkouška: 30 bodů

- Ukončení:
  - Zápočet: min. 60 bodů
  - Zkouška: min. 70 bodů

Osnova předmětu

- Úvod do Java EE (architektura, technologie, koncepty)
- Perzistence dat (ORM, JPA, Spring JDBC, iBatis, Testování)
- Aplikační logika (IoC, AOP, Transakce, Bezpečnost, Testování)
- Prezentační vrstva (Webové frameworky, Stripes, Spring MVC, Wicket, JSF, Bezpečnost)
- Integrační technologie (Webové služby SOAP, REST, JMS, RMI, IIOP, ESB)
- Testování (jednotkové, integrační, funkční, akceptační, uživatelské přívětivosti, výkonnosti, bezpečnosti)

Doporučené zdroje



- **Effective Java (2nd Edition)**
- Joshua Bloch
- http://amazon.com/dp/0321356683/
- Ostatní viz osnova v IS

PLATFORMA JAVA EE

Co je to platforma Java EE

- Platforma pro vývoj moderních informačních systémů
  - Poskytuje potřebnou infrastrukturu
  - Průmyslový standard (JCP)
  - Aktuální verze je Java EE 6 (JSR 316)

- Podpora pro vývoj
  - Webových aplikací
  - Webových služeb
  - Vícevrstvých aplikací

Charakteristika moderních IS

- Složité, rozsáhlé a komplexní systémy
- Nutnost integrace s ostatními systémy v rámci organizace i mimo ni
- Přizpůsobitelnost požadavkům různých zákazníků
- Provoz na různých platformách
- Podpora pro velké množství klientů (zejména u webových aplikací)
- Bezpečnost

Požadavky na vývoj IS

- Rychlý vývoj
- Snadná údržba
- Snadná rozšiřitelnost a přizpůsobení
- Snadná integrovatelnost s jinými systémy
- Podpora pro agilní metodiky
- Podpora pro týmový a multi-týmový vývoj
- Přenositelnost
  - Různé SW i HW platformy, různé nástroje i aplikační servery
- Škálovatelnost
- Bezpečnost

ZÁKLADNÍ KONCEPTY

Základní koncepty

- Infrastruktura
- Modularita
- Nezávislost a nízká invazivnost
- Deklarativní přístup
- Dodržování obecných zásad pro vývoj udržovatelného kódu

Infrastruktura

- Vývojář by se měl zaměřit na vlastní problémovou doménu a neměl by být nucen se zabývat obecnými problémy, které je nutné řešit v každé aplikaci.
  - Architektura aplikace, bezpečnost, řízení transakcí, perzistence dat, komunikace a integrace, vzdálený přístup, infrastruktura prezentační vrstvy, lokalizace, atd.
- Platforma Java EE a na ní postavené aplikační rámce (frameworky) proto nabízejí potřebnou infrastrukturu.

- Nikdy neimplementujte svůj vlastní framework!

Modularita

- Aplikace je vyvíjena jako množina spolupracujících komponent
- Komponenty by měly
  - Být volně propojené (*loosely coupled*), tj. mělo by mezi nimi být co nejméně závislostí
  - Být znovupoužitelné (ať už pouze v rámci projektu, nebo i mimo něj)
  - Mít dobře navržené rozhraní
  - Být dobře otestované
- Pokud máme množinu dobře navržených komponent, je snadné měnit a přizpůsobovat chování aplikace
  - Výměnou komponenty
  - Změnou konfigurace komponenty
  - Změnou propojení mezi komponentami

Nezávislost a nízká invazivnost

- Komponenty by měly být nezávislé nejenom mezi sebou, ale také na konkrétních technologiích a aplikačních rámcích
- To zjednodušuje údržbu a zvyšuje znovupoužitelnost
- Koncept POJO komponent

Deklarativní přístup

- Určité aspekty chování programu nejsou definovány tradičním imperativním kódem (tj. posloupností příkazů *jak se to má udělat*), ale specifikací cíle (tj. *co se má udělat*).
- To vede ke zjednodušení a zpřehlednění kódu.
- Vhodné např. pro řízení transakcí, řízení bezpečnosti a přístupových práv, automatické konverze, různá automatická mapování, apod.
- Vlastní deklarace požadovaného chování může být umístěna
  - V popisovači nasazení (*deployment descriptor*)
  - Přímo v kódu prostřednictvím anotace (modernější a preferovaný přístup)

Imperativní řízení transakcí

```
public void someMethod() {

    UserTransaction transaction = context.getUserTransaction();

    try {
        transaction.begin();
        [redacted]
        transaction.commit();
    } catch (Exception ex){
        try {
            transaction.rollback();
        } catch (SystemException syex){
            throw new EJBException
                ("Rollback failed: " + syex.getMessage());
        }
        throw new EJBException
            ("Transaction failed: " + ex.getMessage());
    }
}
```

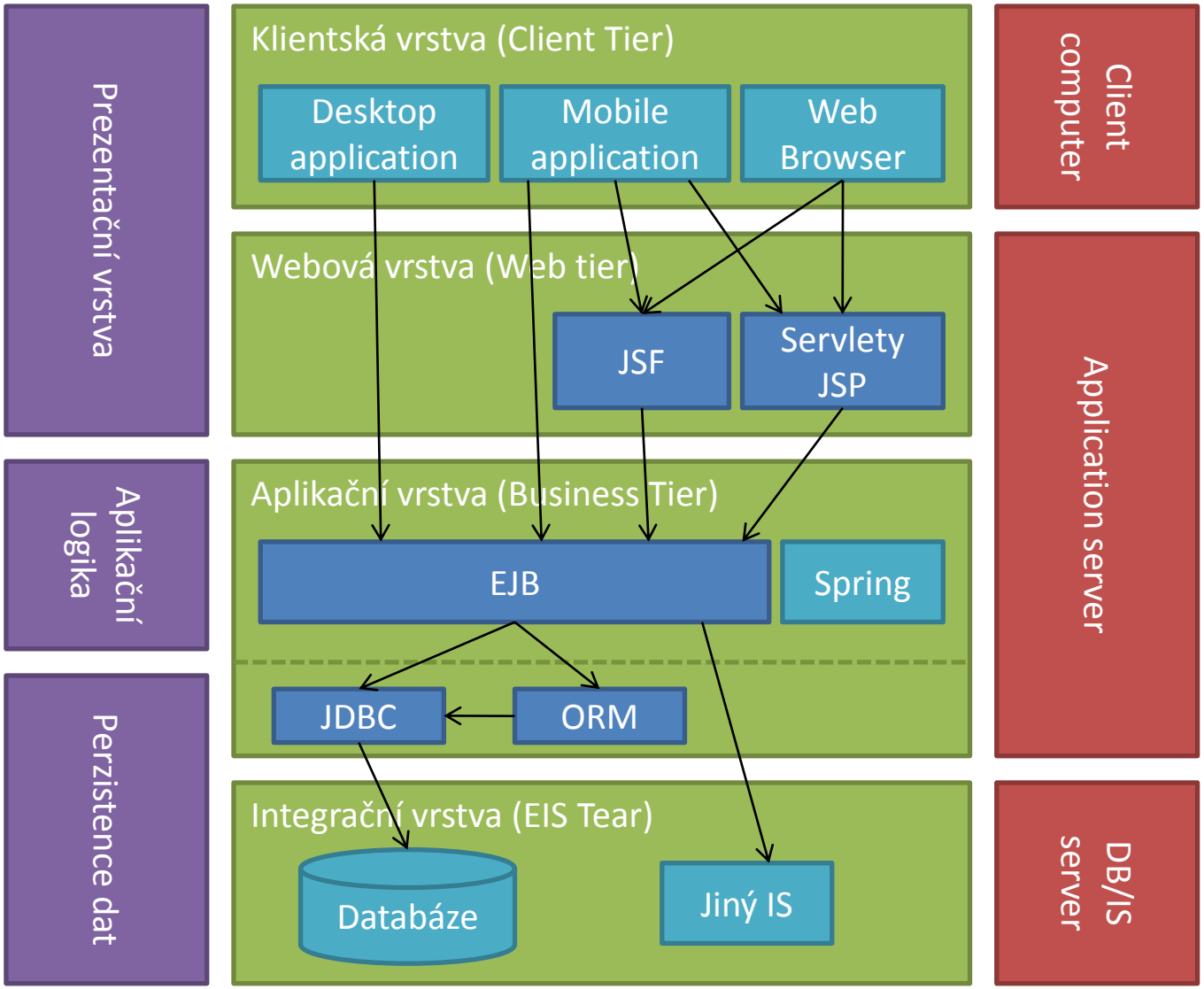
Deklarativní řízení transakcí

```
@TransactionAttribute(TransactionAttributeType.RequiresNew)
public void someMethod() {
    [redacted]
}
```

Jak to bylo v dřívějších verzích

- První verze platformy Java EE byly zaměřeny zejména na infrastrukturu a technologie.
- Snadnost vývoje byla podceňována
  - Složité technologie se složitým použitím
  - strmá učící křivka
  - Nutnost používání složitých nástrojů
- To vedlo k frustraci vývojářů a ke vzniku alternativních přístupů a technologií (Hibernate, Spring)
- Změna přišla s Java EE 5
  - Silná inspirace nástroji Spring, Hibernate, apod.
  - Anotance
  - POJO komponenty

ARCHITEKTURA & TECHNOLOGIE


Prezentační vrstva
Aplikační logika
Perzistence dat
Klientská vrstva (Client Tier)
Desktop application
Mobile application
Web Browser
Webová vrstva (Web tier)
JSF
Servlety JSP
Aplikační vrstva (Business Tier)
EJB
Spring
JDBC
ORM
Integrační vrstva (EIS Tear)
Databáze
Jiný IS
Client computer
Application server
DB/IS server

Prezentační vrstva

Desktopové aplikace
- Swing
- AWT
- SWT
- Java Web Start

Mobilní aplikace
- Java ME
- Android/iOS/BlackBerry OS/Windows Phone

Webové aplikace
- Servlety, JSP, JSTL
- MVC frameworky
  - Request based (Struts, Stripes, Spring MVC)
  - Component based (JSF, Tapestery, Wicket)
- Portlety
- Aplety

Aplikační logika

Obyčejná knihovna tříd

- Pro větší aplikace nevhodné řešení

EJB

- Vyžaduje aplikační server s podporou *EJB* nebo *EJB lite*.

Spring framework

- Není standardní součástí Java EE
- Přesto je velmi oblíbený
- Není invazivní

Persistence dat

JDBC

- Univerzální API pro přístup k DB
- Těžkopádné na přímé používání
  - Template Method
  - Spring JDBC
  - Commons DB
  - RowSet

ORM

- Standard JPA (aktuálně JPA 2.0)
- Hibernate, TopLink, Eclipse Link

Zastaralé technologie

- EJB 2.x
- JDO

Aplikační servery

Open Source – plnohodnotné

- JBoss
- Glassfish

Open Source – pouze servlet kontejner

- Tomcat
- Jetty

Komerční

- WebSphere (IBM)
- WebLogic (Oracle, dříve BEA)

Závěr

?