
Úvod do předmětu, průběh, kritéria hodnocení. Soudobé trendy budování rozsáhlých aplikací.

Obsah

Charakteristika a cíle předmětu a průběh výuky	1
Charakteristika	2
Průběh výuky	2
Úlohy	2
Projekty	2
Zkouška	3
Složky hodnocení	3
Kritéria hodnocení předmětu	3
Obsah předmětu	3
Obsah předmětu	4
Metodiky a nástroje vývoje rozsáhlých (Java EE) aplikací	4
Datová vrstva javových aplikací	4
Webové aplikace - základy	4
Webové aplikace - pokročilá témata	5
Webové aplikační rámce	5
Aplikační logika Java EE aplikací	5
Prezentační vrstva - nové technologie	5
Enterprise JavaBeans 3.0	6
Systémy řízení zpráv	6
Architektury orientované na služby	6
Webové služby	6
Webové služby - alternativní přístupy (REST)	7
Testování Java EE aplikací	7
Bezpečnost Java EE aplikací	8
Soudobé trendy budování rozsáhlých aplikací	8
Orientace na služby	8
Orientace na pokrytí podnikových procesů	8
Orientace na integrovatelnost	8
Agilní metodiky vývoje	9
Vývojové nástroje	9
Testování jako integritní součást vývoje	10
Volba vhodných technologií	10

Charakteristika a cíle předmětu a průběh výuky

Charakteristika

Předmět PA165 Vývoj programových systémů v jazyce Java [<http://is.muni.cz/predmety/predmet.pl?fakulta=1433;obdobi=3523;kod=PA165>] je magisterským předmětem určeným pro zájemce o hlubší proniknutí do principů výstavby rozsáhlejších programových systémů na platformě Java.

Cílem je seznámit s principy:

- kvalitního návrhu rozsáhlých systémů (jak navrhnout architekturu, jakou zvolit infrastrukturu/rámec...),
- jejich tvorby (jaké zásady, nástroje,...)
- testování (při vývoji, zavádění a v provozu),
- refaktoringu (=zlepšování kódu bez změny funkcionality),
- ladění výkonu (optimalizace)

TENTO PŘEDMĚT NENÍ O JAVĚ, ALE O NÁVRHU ROZSÁHLÝCH APLIKACÍ - Java je ale vhodná platforma, kde to ukázat...

Průběh výuky

Výuka probíhá jako:

- Dvouhodinová přednáška - Po od 18.00 v A107
- Dvouhodinová cvičení v nové učebně B130 (v přízemí "za halou")
- Nedílnou součástí je řešení projektu - ve cvičení s možností konzultací nebo ve volném čase
- Na doplnění - pro závazné zvládnutí základních postupů - jsou zařazeny dvě úlohy zadávané (příp. i řešené) na cvičeních

Úlohy

Těžištěm předmětu je samostatná práce na projektech. Společně zadávaných úloh je proto minimum a znamenají jen 15 bodů.

Projekty

- Projekty jsou koncipovány jako týmové (čtyřčlenné týmy).
- Tematicky vždy pokrývají netriviální část z celkového obsahového záběru předmětu a musí tomu být

i technologicky přizpůsobeny.

- Zadáání projektu stanovuje nebo schvaluje cvičící.
- Zadáání projektů mohou souviset/navazovat s tématem bakalářské/diplomové práce nebo práce v rámci výzkumného projektu.

Zkouška

Zkouška bude ústní.

Student při ní musí prokázat dobrý přehled ve studované problematice s tím, že principy a technologie, které aplikoval při řešení projektu, musí zvládnout do větší hloubky.

Složky hodnocení

Hodnocení předmětu je stanoveno podle celkového součtu bodů z jednotlivých kritérií:

- Úlohy - 15 bodů
- Projekt - 50 bodů
- Ústní zkouška - 35 bodů

Kritéria hodnocení předmětu

K ukončení zkouškou potřebujete:

- A 94 - 100 bodů
- B 88 - 93 bodů
- C 82 - 87 bodů
- D 76 - 81 bodů
- E 70 - 75 bodů
- F 0 - 69 bodů

K ukončení zápočtem potřebujete:

- 60 bodů

Obsah předmětu

Obsah předmětu

Sylabus předmětu PA165 v IS [<https://is.muni.cz/predmety/predmet.pl?id=363919>].

Průběžné informace k obsahu přednášek a cvičení naleznete:

- v osnově předmětu - IS [<https://is.muni.cz/auth/el/1433/podzim2006/PA165/>]
- na wiki předmětu - <http://kore.fi.muni.cz:5080/wiki/index.php/PA165>

Nyní podrobněji k tématům...

Metodiky a nástroje vývoje rozsáhlých (Java EE) aplikací

Trendem dneška jsou agilní metodiky vývoje s důrazem na práci v týmu, kolektivní vlastnictví kódu, důsledné testování, kvalitu kódu.

- Metodiky - Agile Programming (vč. Extreme Programming), Test-driven Development, Continuous Integration
- Nástroje - sdílení kódu a práce v týmu, pokročilé řízení sestavování, dokumentace, testování

Datová vrstva javových aplikací

- Trendem nejsou jen "těžké" technologie disponující "neomezenou" škálovatelností, distribuovaností, robustností,... (EJB).
- Při řešení jednodušších problémů mají své místo "lehké" postupy (ORM - Java Persistence API, Hibernate...).
- Z praktického hlediska poznáte oboje: Hibernate (Object-relational Mapping system) a EJB 3.0

Webové aplikace - základy

Mezi platformami podnikových i jiných - nejen Java EE - aplikací zastávají význačné místo *webové aplikace*:

- jsou většinou klient/server, zde řešíme hlavně serverovou část, ale stále více se pozornost obrací i na klientskou část (využívající asynchronní komunikaci - AJAX)
- v úvodu poznáte hlavní serverové webové technologie platformy Java EE (servlety, JSP), jejich tvorbu a nasazení
- a rovněž prostředí pro jejich běh (javové webové kontejnery - Tomcat, Jetty)

Webové aplikace - pokročilá témata

Kromě základních servletů a JSP se používají:

- JSF - moderní prostředek návrhu webové (prezentační) vrstvy s podporou fy Sun v NetBeans
- knihovny (JSP) značek - umožní zapoužít aplikační logiku a dostat ji z JSP stránek. Existuje standardní knihovna JSTL.
- Internacionalizace/lokalizace webových aplikací

Webové aplikační rámce

Vývoj se kromě triviálních aplikací dnes děje za pomoci *webových rámců* (web frameworks).

- Rámce řeší "kde co": od podpory prezentační vrstvy (šablony, validace vstupů, lokalizace), aplikační logiky (řízení toku výpočtu, Aspect-oriented Programming, testování), až po datovou vrstvu (např. perzistenci javových objektů) a bezpečnost aplikací (JAAS, Acegi)
- Proto v jedné přednášce poznáte rámec Struts (starý, dobrý, známý...).
- Existují i velmi moderní "lehké" rámce (Spring).

Aplikační logika Java EE aplikací

Trendem je umožnit vývojáři soustředit se na samotnou podstatu, tj. *aplikační logiku*, a ostatní (perzistenci, transakce, lokalizaci, prezentaci...) ponechat na vývojovém a běhovém prostředí.

- používají se technologie a rámce umožňující používat "POJO" (Plain-Old Java Objects)
- využívá se postupů "obrácení řízení" výpočtu (Inversion of Control) a aspektově-orientovaného programování (Aspect oriented Programming)
- + nové možnosti Java 5 - anotace (příklad: EJB 3.0 v Java EE 5)

Prezentační vrstva - nové technologie

Trendem je nabídnout uživatelům webových aplikací *stejný komfort obsluhy* jako u aplikace desktopové:

- používá se asynchronní komunikace prohlížeč (tj. klientská část) - server
- (= uživatel nemusí klikat "Submit" a čekat, aplikace komunikuje na pozadí, "tahá si" potřebná data, sama aktualizuje formuláře...)

- vyžaduje tedy (přenositelné) skriptování na straně klienta
- zastřešující označení je Asynchronous JavaScript + XML = AJAX (i když se v reálu XML úplně vždy nevyužívá...)
- existuje řada knihoven, které to již umí

Enterprise JavaBeans 3.0

- Enterprise JavaBeans jsou odjakživa "vlajkovou lodí" platformy Java EE.
- Představují ucelené řešení aplikační, perzistenční i části prezentační vrstvy rozsáhlých Java EE aplikací.
- Do verze 3.0 byly technologií mocnou, robustní, škálovatelnou.
- Nyní (od 3.0) jsou i (relativně) snadno použitelnou - vč. podpory v IDE jako je NetBeans.

Systémy řízení zpráv

Pro integraci nezávislých systémů (tj. vytvoření "konfederace" IS) je vhodnou platformou systém řízení zpráv (Messaging System).

- systémy existují nadále nezávisle
- mohou si asynchronně vyměňovat zprávy (možnost zajištění doručení, perzistence zpráv...) prostřednictvím speciálního middleware - systému řízení zpráv

Java poskytuje rozhraní JMS (Java Messaging System) nabízející řadu implementací - i open-source/zdarma.

Architektury orientované na služby

Důraz je dnes kladen na integrovatelnost a znovupoužitelnost - nikoli objektů či komponent, ale rovnou jejich funkcionality - reprezentované službami.

- rozsáhlé podnikové systémy se budují na architekturách orientovaných na služby
- problémem je interoperabilita (technické aspekty), podpora middleware (Enterprise Service Bus), ale stále více i vazba na podnikové procesy (Business Processes)
- = máme služby, úkolem je je "rozehrát, dirigovat" - service orchestration

Webové služby

Základní technologií servisně orientovaných architektur jsou webové služby (Web Services), většinou na bázi SOAP (Simple Object Access Protocol).

- Java EE dnes poskytuje plnohodnotné prostředí pro tvorbu a běh WS
- vč. zabezpečení a
- integrace

Webové služby - alternativní přístupy (REST)

Ne vždy je nutné/vhodné používat "těžké" WS technologie jako je SOAP.

Pro jednoduchá řešení se nabízejí (i v Javě realizovatelné) služby na bázi architekturních principů REST (REpresentational State Transfer):

- vše je zdroj (resource), vč. zdrojů malých (fine-grained)
- zdroj je adresovatelný URI (fakticky URL)
- zdroje mají své reprezentace (digitální podoby, např. XML, HTML, obrazová data)
- aplikace komunikují výměnou těchto reprezentací
- aplikace mohou být zároveň servery (poskytují zdroje) i klienty (potřebují jiné zdroje)
- využívá se jednoduché komunikace - základní HTTP (metody GET, POST, PUT, DELETE znamenají čti, vytvoř, změň a smaž zdroj)
- výhoda je v kešovatelnosti, škálovatelnosti... HTTP technologie je dobře a komplexně zvládnutá (to se o SOAP říci nedá)
- rovněž režie REST je oproti SOAP malá

Testování Java EE aplikací

Dnes lze testovat "cokoli" nad Java EE:

- datová vrstva (DBUnit)
- aplikační vrstva (JUnit a nástavby, Artima SuiteRunner...)
- prezentační vrstva (JWebUnit)
- a mnoho dalších nástrojů

Specifickým testovacím postupem je použití tzv. Mock-objects (zástupných objektů) umožňujících tes-

tování jednotek (např. servletu nebo JDBC klienta) i mimo reálné aplikační prostředí - je to mnohem pohodlnější, dynamičtější, rychlejší...

Bezpečnost Java EE aplikací

Bezpečnost Java EE aplikací má řadu rovin a úzce souvisí se systémovým zajištěním bezpečnosti běhového prostředí Java EE aplikace.

Zde si budeme všimnout zejména aspektů úžeji vázaných na samotnou Javu EE:

- bezpečnostní infrastruktura Javy EE: JAAS, Security Manager, vazba na PKI
- bezpečnost webových aplikací
- bezpečnost webových služeb (WS Security)
- XML zabezpečení (XML Signatures)

Soudobé trendy budování rozsáhlých aplikací

Orientace na služby

Aplikace se nebudují jako monolity, důraz je kladen na znovupoužitelnost v nových kontextech a na interoperabilitu mezi platformami

- SOA (Service-Oriented Architecture)
- ESB (Enterprise Service Bus)
- JBI (Java Business Integration)
- SCA (Service Component Architecture)

Orientace na pokrytí podnikových procesů

Procesy jsou chápány jako klíčové východisko budování IS. Architektury softwaru se jim podřizují.

- Business Process Modeling
- BPEL (Business Process Expression Language)
- XPDL (XML Process Definition Language)

Orientace na integrovatelnost

Nejen webové služby (klasické, SOAP-based) jsou nástrojem k přímé a snadné integrovatelnosti:

- Portlets
- REST webové služby
- systémy řízení zpráv - JMS
- vznikají i rozsáhlé middlewarové infrastruktury: OSGi™ - The Dynamic Module System for Java™

Zásadou je "Continuous Integration" - metodická zásada nutící vývojáře produkovat kontinuálně rostoucí, přitom stále bezchybný kód i rozsáhlých systémů.

Cílem je produkovat snadno znovupoužitelné a integrovatelné služby řešící dílčí, dobře definované úlohy.

Agilní metodiky vývoje

Trendem není rigidní vývoj typu "vodopád", ale aktivní participace vývojářů a uživatelů během celého vývojového cyklu.

- ideální jsou tedy agilní metodiky
- s důkladným testováním na všech úrovních (testy jednotek, integrační testy, testy použitelnosti)
- důraz je na týmovou práci s odpovídajícími nástroji (sdílení kódu, testů)
- využívání "sdílené zkušenosti" - vzory (nejen klasické objektové návrhové vzory - GoF, ale i vzory analytické a implementační - např. v EJB) a osvědčené praktiky
- důraz je na kontinuálnost - kvality kódu - Refactoring, integrovatelnosti - Continuous Integration

Vývojové nástroje

Pod vývojovým nástrojem si nelze představit jen integrované prostředí, kde se aplikace "nakliká" (NetBeans, Eclipse, IDEA...).

Patří sem také dílčí (řádkové) nástroje pro jednotlivé úlohy vývoje:

- prostředí pro správu zdrojů a týmový vývoj - Subversion či klasické CVS nebo profesionální ClearCase
- prostředí pro sestavování, testování, dokumentaci, distribuci aplikací - Maven či klasický Ant
- prostředí pro skriptování - JavaScript, Groovy, JRuby...
- prostředí pro tzv. Domain-driven Development - Trails, Grails, Ruby on Rails

Testování jako integritní součást vývoje

Testování se stává integritní součástí vývoje. V komerčních, podnikových, IS stejně jako při vývoji open-source. Patří k "dobrému tónu".

- Hovoří se o vývoji řízeném testy - TDD (Test Driven Development)
- Testování dnes dokáže pokrýt všechny fáze i všechny vrstvy Java EE aplikace (datovou, aplikační i prezentační).

Volba vhodných technologií

Opět není cílem použít jednu předchystanou sadu technologií a nástrojů, ale umět si z velkého množství stávajícího "materiálu" vybrat.

Role systémových architektů je extrémně důležitá.

- Realizují se řešení na bázi zcela "lehkých" technologií ("lehké" aplikační rámce, ORM místo EJB, jednoduché REST služby místo SOAP...).
- Naproti tomu existují komplexní vývojová a aplikační prostředí pomáhající realizovat rozsáhlé systémy "z jedné vody"/jednoho prostředí (NetBeans, jboss).
- ...
- Někdy se "skriptuje", navrhuje stylem RAD (Rapid Application Development), jindy se používají "těžké" technologie a mocné nástroje.
- Nerozpakujeme se navrhnout architekturu nad kombinací prostředí (Java + skriptovací jazyk, Java + .NET služby...)

Cílem tohoto předmětu je dát základní orientaci v této džungli...