

СОФТУЕРНА АРХИТЕКТУРА MVC И JAVASERVER FACES (JSF)

СОФТУЕРНА АРХИТЕКТУРА MVC И JAVASERVER FACES (JSF)

Цел

*Запознаване с архитектурата на
разработката на Уеб приложения
с JSF2+ и отделните слоеве*

Съдържание

- Софтуерни рамки - определение
- Шаблони в софтуерните архитектури
- Въведение в JSF
- MVC
- Слоеви на архитектурата – цел и задачи
- Примери за уеб приложение – сравнение между JSP и JSF.

Съдържание на модулите (1/2)

Част 1:

1. Анотации и мета-програмиране
2. JSF въведение
3. JSF правила за навигация
4. JSF управляеми компоненти
5. JSF компоненти на интерфейс

Съдържание на модулите (2/2)

Част 2:

- 6. JSF & CSS стилове
- 7. Интернационализация в JSF
- 8. JSF и съобщения
- 9. JSF и валидация
- 10. JSF и конвертори

Съдържание

Част 1:

☐ JSF въведение

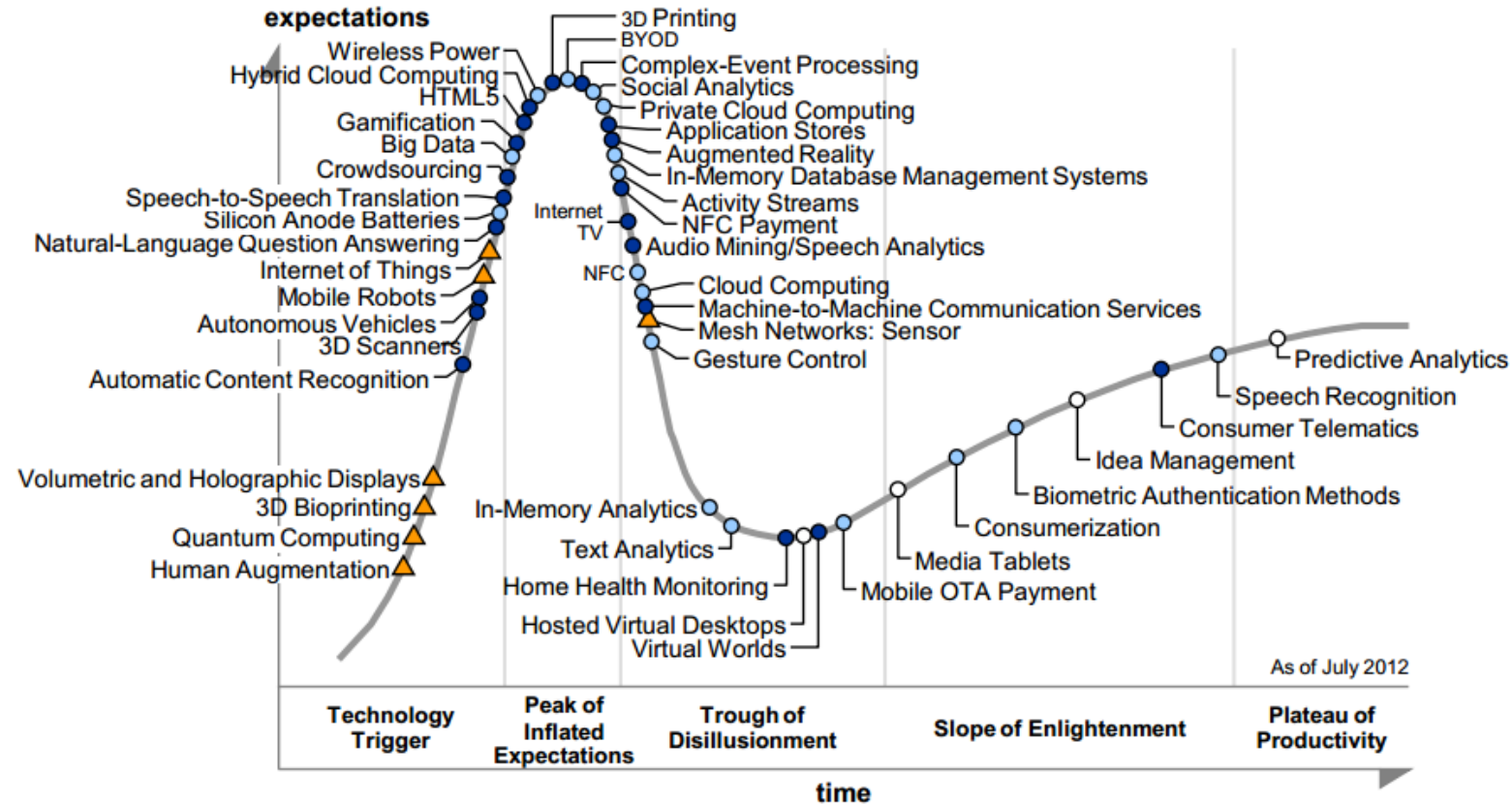
- ☐ JSF правила за навигация
- ☐ JSF управляеми компоненти
- ☐ JSF компоненти на потребителски интерфейс

Бъдещите технологии (2013-10-19)

- *„In 2013, mobile devices will pass PCs to be most common Web access tools. By 2015, over 80% of handsets in mature markets will be smart phones.“*

източник Forbes.com

Emerging Technologies Hype Cycle 2012



Plateau will be reached in:

○ less than 2 years ● 2 to 5 years ● 5 to 10 years ▲ more than 10 years ⊗ obsolete before plateau

Gartner

2016, august

Gartner Hype Cycle for Emerging Technologies, 2016



gartner.com/SmarterWithGartner

Source: Gartner
© 2016 Gartner, Inc. and/or its affiliates. All rights reserved.

Gartner.

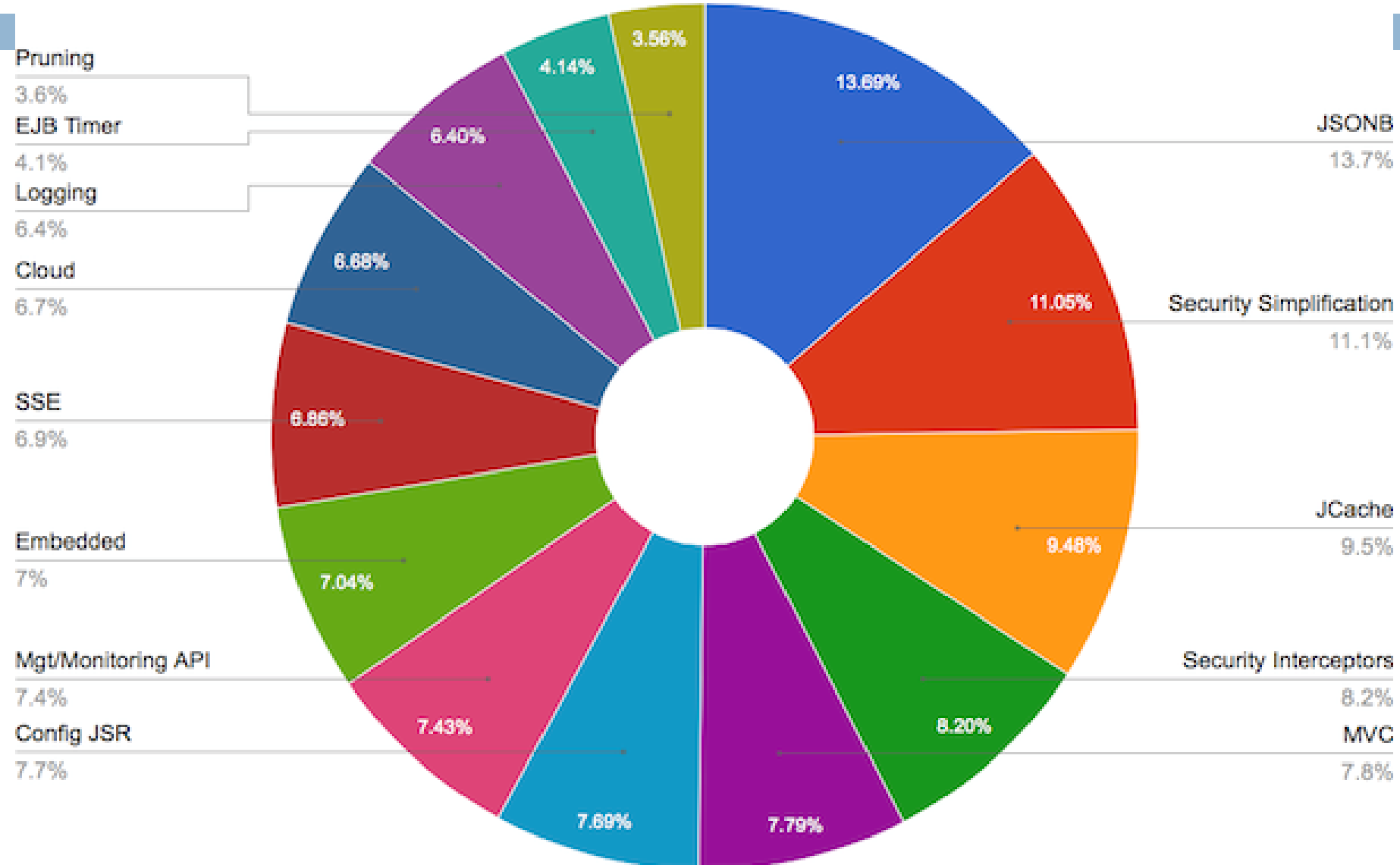
Определение на JavaEE (Oracle Corp, 2015)

- *“Java Platform, Enterprise Edition (Java EE) is the standard in community-driven enterprise software. “*
- *“Java EE is developed using the Java Community Process, with contributions from industry experts, commercial and open source organizations, Java User Groups, and countless individuals.”*

JavaEE 7 (последна версия, 2017-03)

- • JSR 236: Concurrency Utilities for Java EE 1.0
- • JSR 338: Java Persistence API 2.1
- • JSR 339: Java API for RESTful Web Services 2.0
- • JSR 340: Java Servlet 3.1
- • JSR 341: Expression Language 3.0
- • JSR 342: Java Platform, Enterprise Edition 7
- • JSR 343: Java Message Service 2.0
- • JSR 344: JavaServer Faces 2.2
- • JSR 345: Enterprise JavaBeans 3.2
- • JSR 346: Contexts and Dependency Injection for Java EE 1.1
- • JSR 349: Bean Validation 1.1
- • JSR 352: Batch Applications for the Java Platform 1.0
- • JSR 353: Java API for JSON Processing 1.0
- • JSR 356: Java API for WebSocket 1.0

Survey JavaEE8-results



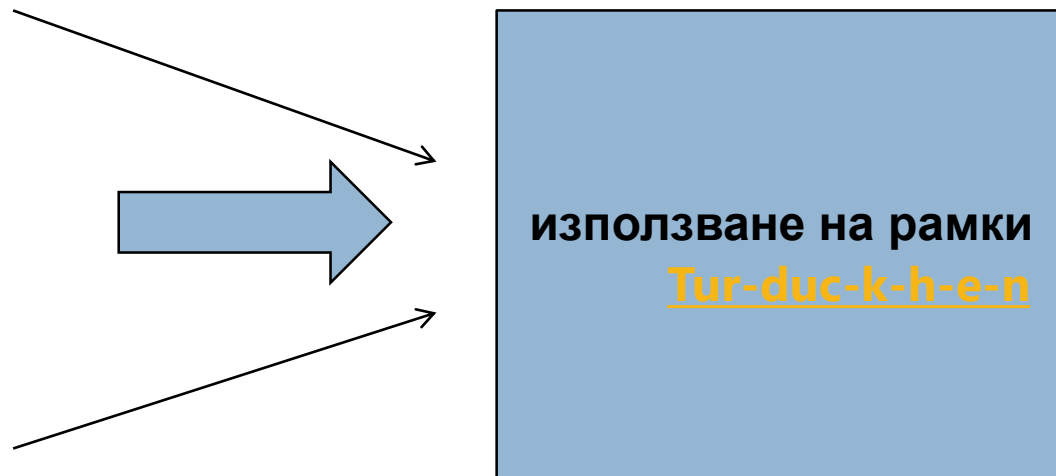
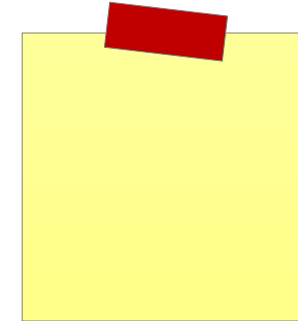
JavaEE 7(3/3a) – Towards JavaEE8

What's New

- [JSR 366](#) - Java EE 8
- [JSR 367](#) - The Java API for JSON Binding
- [JSR 368](#) - Java Message Service 2.1
- [JSR 369](#) - Java Servlet 4.0
- [JSR 370](#) - Java API for RESTful Web Services 2.1
- [JSR 371](#) - Model-View-Controller 1.0
- [JSR 372](#) - Java Server Faces 2.3
- [JSR 373](#) - Java EE Management API 1.0
- [JSR 374](#) - Java API for JSON Processing 1.1
- [JSR 375](#) - Java EE Security API 1.0

Софтуерна рамка (фреймуърк)

- предимства
 - решава даден кръг проблеми
- недостатъци
 - налага различни ограничения



Въведение в JSF

- софтуерна рамка (framework) за създаване на Уеб приложения
- работи от страната на сървъра (server-side)
- Улеснява:
 - ▣ навигацията между отделните страници
 - ▣ извеждането на динамично съдържание
 - ▣ валидацията и преобразуването на въведените данни
 - ▣ обработката на събития
 - ▣ работа с бизнес компоненти (JavaBeans)

Нов начин за разработка на уеб приложения с Java

http://raibledesigns.com/rd/entry/my_comparing_jvm_web_frameworks

My Comparing JVM Web Frameworks Presentation from Devoxx 2010 This week, I've been having a great time in Antwerp, Belgium at the Devoxx Conference. This morning, I had the pleasure of delivering my Comparing JVM Web Frameworks

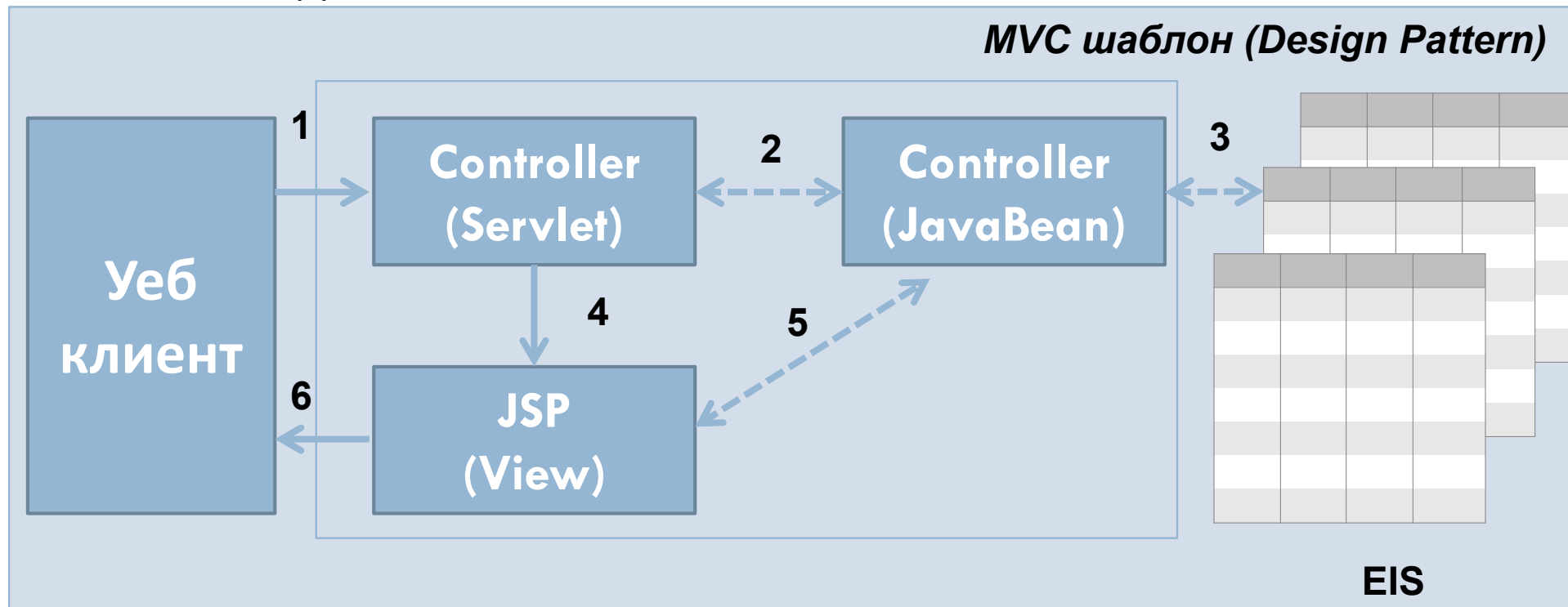


Предимства

- възможност за генериране на съдържание не само в HTML
- **MVC** софтуерна архитектура
- използва **EL**
- лесна конфигурация
- възможност за създаване на нови компоненти

MODEL-VIEW-CONTROLLER (MVC)

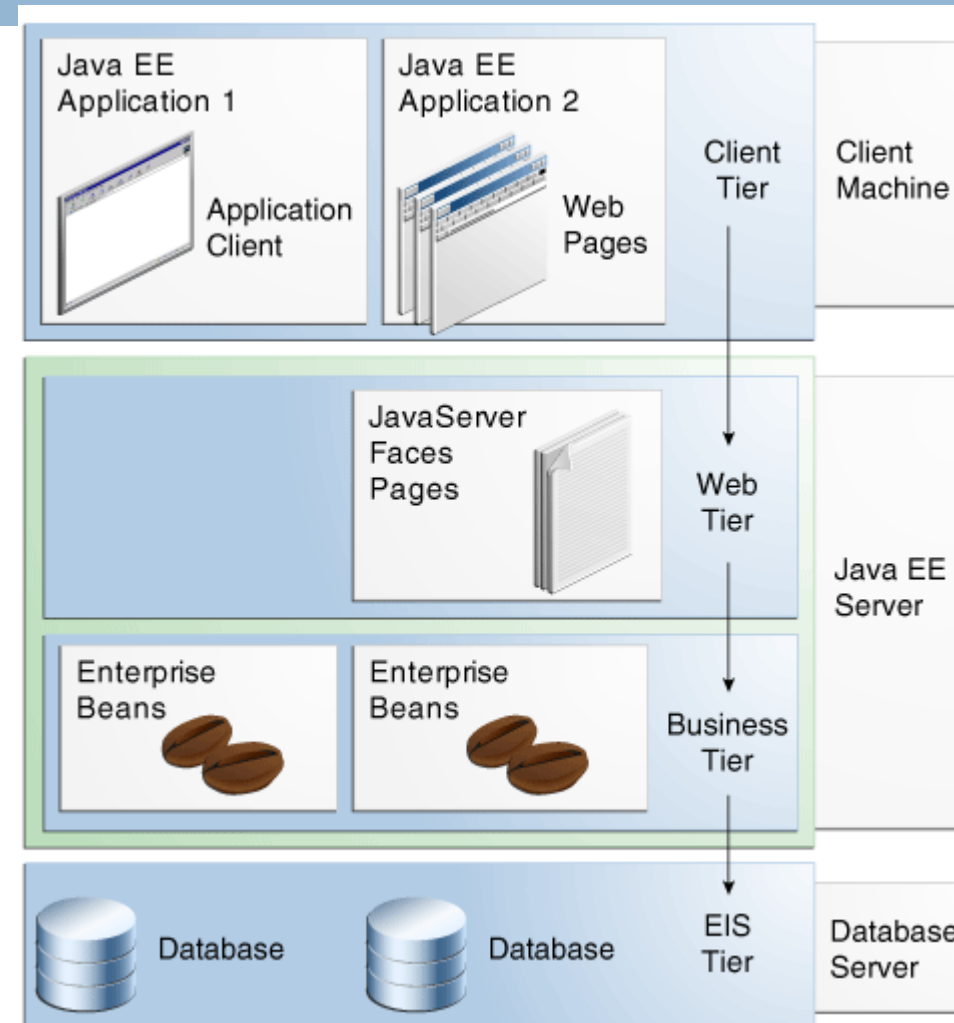
- Архитектурен модел, който използва набор от **програмни шаблони за проектиране** (design patterns), в които основното е логическото разделяне на едно приложение на следните части:



JavaEE7 (sept. 2013)

Source: <http://docs.oracle.com/>

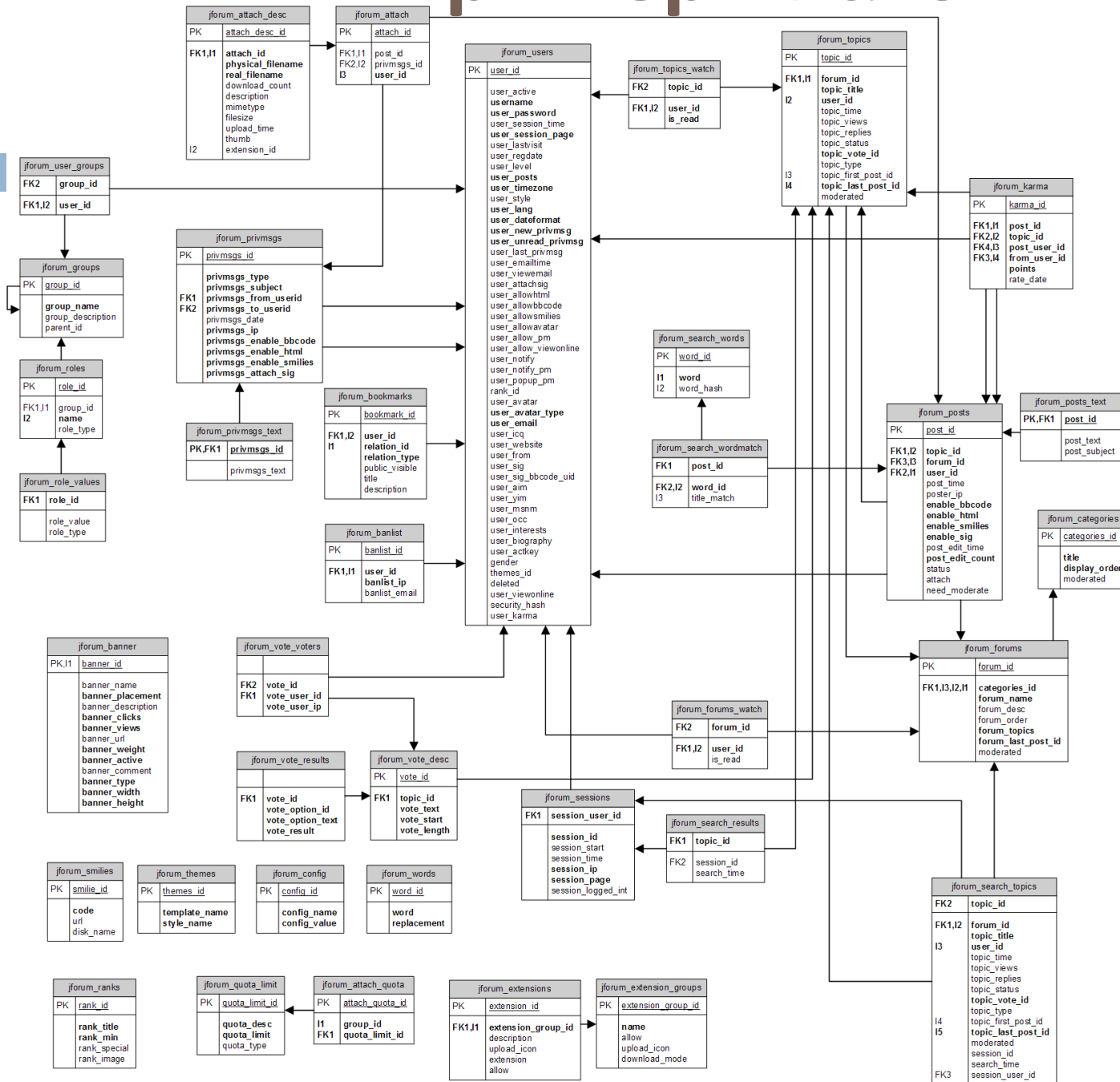
[Description of "Figure 1-1 Multitiered Applications"](#)



Слой 1: модел (Model) - характеристики

- съдържа моделът на данните, с които работи приложението;
- отделните класове (логически единици и/или таблици) и връзките между тях;
- разделяне от графичното представяне;

Пример 1: слой модель (1/2)



□ jforum.net

Слой 2: Изглед (View)

- интерфейс, чрез който потребителите взаимодействат с приложението;
- целта му е да визуализира обработените данни от приложението (посредством **JSP**, **AWT**, **Swing**, **JavaFX**,...) и да предостави възможност на потребителя да изпълнява команди (да променя данните);
- в този слой не трябва да има бизнес-логика, която да дефинира и определя поведението на приложението;

Слой 3: Контролер (Controller)

- Осъществява връзката между модела (model) и изгледа (view)
- Контролерът е мястото където се намира логиката на приложението
- Извлича необходимата информация от модела (например, от база данни) и предава резултата към изгледа, който се грижи за външното му оформление
- Получава команди от потребителя, посредством изгледа, и обработва данните
- Избира следващия "екран" на приложението (страница, диалогов прозорец, т.н.)

Слой 3: Контролер (примерен код)

<html>

...

<body>

Вие сте на: \${ 2013-1974 } години!

</body>

</html>



Динамично (сървърно)
изчисление (аритметичен
израз).

Слой 3: Контролер (примерен код)

EL Expression	Result
$\${1}$	1
$\${1 + 2}$	3
$\${1.2 + 2.3}$	3.5
$\${1.2E4 + 1.4}$	12001.4
$\${-4 - 2}$	-6
$\${21 * 2}$	42
$\${3/4}$	0.75
$\${3 \text{ div } 4}$	0.75
$\${3/0}$	Infinity
$\${10\%4}$	2
$\${10 \bmod 4}$	2
$\${(1==2) ? 3 : 4}$	4

MVC често се описва чрез формализмът на UML

- Пример

Структура на JSF приложение

- Набор от страници за визуализация на съдържание (**JSP**, **Facelets**, ...)
- Набор от **JavaBeans**, които дефинират свойства (property) и методи, използвани от компонентите на страницата
- Конфигурационен файл, който описва правилата за навигация между страниците, използваните **JavaBeans** обекти, потребителски компоненти и други
- Дескриптор за внедряване (deployment descriptor), чрез който приложението се представя пред **WEB** сървър
- Набор от компоненти, създадени от потребителя – валидатори, конвертори, тагове и т.н.

Примерна страница (JSF2+)

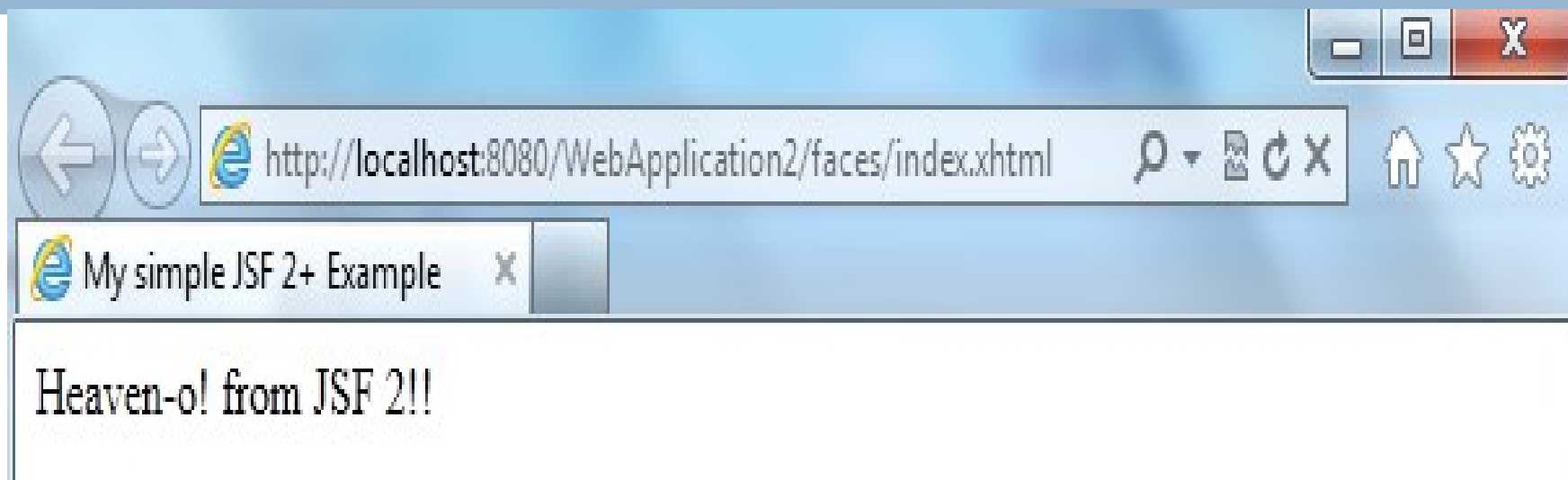
```
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
4 <html xmlns="http://www.w3.org/1999/xhtml"
5   >
6 </html>
7 <h:head>
8   <title>My simple JSF 2+ Example</title>
9 </h:head>
10 <h:body>
11   <f:view >
12     <h:outputText value="Heaven-o! from JSF 2!!" />
13   </f:view>
14 </h:body>
15 </html>
```

1. Включване на библиотеки с префикси h и f

2. Изгледите се обявяват с f:view

3. Html компоненти – например за изход:
`<h:outputText... />`

Примерен изход (JSF 2)



В лекцията бяха разгледани

- ▣ предимства и недостатъци от използването на софтуерна рамка (фреймуърк)
- ▣ какво е MVC архитектура
- ▣ въведение в понятията на JSF
 - основните слоеве на JSF архитектурата
 - връзка между сървлети, JSP и JSF технологии
 - примерна JSF страница и основните и елементи

Упражнение 2



Задачи за изпълнение

- **Задача 1:** направете JSF страницата, дадена в примера (която визуализираща съобщение)
- **Задача 2:** Подобно на примера направете JSF страница – `myprofile` – в която да има информация за вас (например имена, факултетен номер, курс, специалност, снимка)
- **Задача 3:** инсталирайте `tomcat` и разгледайте кода на примерите за JSF/JSP
- **Задача 4:** По зададени а) рожденна година и текуща година – напишете да се изчислява на колко години сте – към **задача 2**). б) изведете коя зодия сте (по зададен месец на раждане) с) изведете вашите години / месеци / дни.

Упътване към задачите (1/3)

1. **Към задача 2.1.** Инсталирайте/стартирайте среда за разработка (например: NetBeans 7.3+/tomcat7+), създайте уеб проект, създайте jsf/jsp страница (**lab02_01_jsf_fn9999.html**); – от **слайд 32*** лекцията (*може номера на слайда да се различава от дадения) ; стартирайте проекта;
2. **Към задача 2.2.** Създайте html (**lab02_02_profile_fn9999.htm**) страница с вашият профил и я стартирайте в създаденият в задача 1 проект .

Упътване към задачите (2/3)

3. към Задача 3: инсталирайте tomcat и разгледайте кода на примерите за JSF/JSP (**lab02_03_tomcat8screenshot_fn9999.jpg**); адресът за сваляне е <http://tomcat.apache.org>

Упътване към задачите (3/3)

4. към Задача 4 страница с пресмятане на годините (**lab02_04_my_years_fn9999.htm** вашят профил и я стартирайте в създаденият в задача 1 проект . **Забележка: по подразбиране** адреса на tomcat сървъра е <http://localhost:8080>
- ▣ <http://localhost:8080/examples/jsp/jsp2/el/basic-arithmetic.jsp> (основни операции)
 - ▣ <http://localhost:8080/examples/jsp/jsp2/el/basic-arithmetic.jsp.html> (код)

Ретроспекция: HTML (към упражнението)

- Основни тагове:
 - html,
 - head,
 - title,
 - body, h1, h2, h3
 - p,
 - br
- Well formatted (отварящ/затварящ таг; валидно влагане: единствен основен елемент; всяко дете има родител, без основният елемент)

Разгледани теми

- Софтуерни рамки - определение
- Шаблони в софтуерните архитектури
- Въведение в JSF
- MVC
- Слоеви на архитектурата – цел и задачи
- Примери за уеб приложение – сравнение между JSP и JSF.