

Τεχνολογίες Εφαρμογών Διαδικτύου

Ύλη Προετοιμασίας για Εξέταση Διάλεξης-Εργαστηρίου από Βιβλίο

Servlets και Σελίδες Διακομιστή Java, Εκδόσεις Κλειδάριθμος

Ενημερώθηκε: 20/12/2016

Κεφ 3 Βασικά Στοιχεία Μικρουπηρεσιών

(3.1 εως 3.6) (Όχι 3.7 – 3.8)

Κεφ 4 Χειρισμός Αιτήσεων Πελάτη : Δεδομένα Φόρμας

(4.1 εως 4.6) (Όχι 4.7 – 4.8)

Γενικά μπορείτε να αποφύγετε από το κεφάλαιο αυτό ύλη που αναφέρεται σε κόκκους (Java Beans). Όχι γιατί δεν είναι σημαντική αλλά γιατί αναφερθήκαμε σε κόκκους αργότερα.

Κεφ 9 Παρακολούθηση Συνεδρίας (HttpSessions)

(9.1 έως 9.6)

Κεφ 10 Επισκόπηση της Τεχνολογίας JSP

(10.1 έως 10.2)

Κεφ. 11 Κλήση κώδικα Java με Στοιχεία Σεναρίου JSP

(11.1 έως 11.13) (Όχι ό,τι αναφέρεται σε σύνταξη XML)

Σημείωση:

Τις JSP Παραστάσεις (<%= ... %> τις αναφέραμε στην τάξη σαν JSP Εκφράσεις (Expressions)

Τα μικροσενάρια (<% ... %> (servlets) τα αναφέραμε στην τάξη σαν τμήματα κώδικα

***Δώστε έμφαση στο 11.12 (ειδικά στις μεταβλητές request, response, out, session)
Η παράγραφος 11.13 βοηθά στην κατανόηση των διαφορών στα JSP στοιχεία.***

Κεφ 12 Ελεγχος της δομής των Μικρουπηρεσιών : Η οδηγία Page

(12.1 εως 12.2)

Κεφ 13 Συμπερίληψη αρχείων και μικροεφαρμογών σε σελίδες

(13.1 και 13.2) (σελ. 409-412 και 416-418) Όχι ότι αναφέρεται σε XML

Κεφ 14 Χρήση στοιχείων JavaBeans σε έγγραφο JSP

(14.1 και 14.2) (σελ. 435-438)

Κεφ 15 Ολοκλήρωση Μικρουπηρεσιών και JSP Η αρχιτεκτονική Model View Controller (MVC)

(15.1 και 15.2) (σελ. 469-478)

ΟΧΙ το Κεφ 16 (Γλώσσα Παραστάσεων 2.0).

Κεφ 17 Προσπέλαση Βάσεων Δεδομένων με την JDBC

(17.1 & 17.4)

Σαν παράδειγμα για το JDBC μπορείτε να μελετήσετε το παραδειγμα που παρουσιάστηκε στην τάξη Students.java – βρίσκεται και στην ασύγχρονη)

Κεφ 19 Δημιουργία και Επεξεργασία Φορμών HTML

Περιέχει πολλές λεπτομέρειες για τα στοιχεία φορμών. Εσείς θα πρέπει να γνωρίζετε ότι χρησιμοποιήσαμε στην τάξη (σε παραδείγματα και ασκήσεις). (πχ. Για κάθε συστατικό χρησιμοποιήσαμε μόνο τις ιδιότητες type, name και value).

Παραδείγματα – Ασκήσεις στην (νέα) Ασύγχρονη θεωρούνται ύλη.

Ορισμένες Διαφορές μεταξύ Βιβλίου και Κώδικα που εξετάσαμε στην Τάξη.

Το βιβλίο είχε εκδοθεί όταν η τρέχουσα έκδοση της προδιαγραφής των Servlets ήταν η **2.4** ενώ στην τάξη χρησιμοποιήσαμε την έκδοση **3.1** που υποστηρίζεται από τον Apache Tomcat 8.0

Επίσης το βιβλίο ήταν γραμμένο όταν η τρέχουσα έκδοση της Java ήταν η 1.4 ενώ στην τάξη χρησιμοποιήσαμε την 1.8 ή Java 8.

Στην τάξη χρησιμοποιήσαμε τα παραδείγματα του βιβλίου προσαρμοσμένα για τις πρόσφατες εκδόσεις (όπως παρέχονται και από τους συγγραφείς στην ιστοσελίδα του βιβλίου pdf.coreservlets.com).

Αναμεσα στις διαφορές σημειώνουμε ενδεικτικά τις ακόλουθες σε περίπτωση που κάτι δεν είναι ξεκάθαρο κατά την ανάγνωση:

@WebServlet

Κάναμε χρήση της Java δήλωσης (annotation) @WebServlet για καθορισμό του ονόματος που θα χρησιμοποιηθεί σε ένα URL (διεύθυνση εγγράφου) για κλήση ενός servlet.

Αντί αυτού του ονόματος το βιβλίο χρησιμοποιεί πολλές φορές ένα όνομα της μορφής “/servlet/ονομα_κλασης”. Η μέθοδος δεν παρέχεται από τις πρόσφατες εκδόσεις του Apache Tomcat.

Επίσης ονόματα κλήσης (αντιστοιχία URL σε servlet) μπορούν να δοθούν σε αρχεία **web.xml** (μέσα στον κατάλογο WEB-INF) που περιέχουν XML κώδικα της μορφής:

```
<web-app>
  <servlet>
    <servlet-name>Servlet1</servlet-name>
    <servlet-path>foo.Servlet</servlet-path>
  </servlet>
  <servlet-mapping>
    <servlet-name>Servlet1</servlet-name>
    <url-pattern>/enroll</url-pattern>
  </servlet-mapping>
</web-app>
```

Η λειτουργία αυτή εξακολουθεί να παρέχεται από τους περιέκτες servlets όπως ο Tomcat αλλά στην τάξη βασιστήκαμε κυρίως στη χρήση της δήλωσης (annotation) @WebServlet

Enumeration

Στη σελ. 145 χρησιμοποιείται η σύνταξη

```
Enumeration paramName = request.getParameterNames();
```

Στο κώδικα που εξετάσαμε και εξηγήσαμε χρησιμοποιείται.

```
Enumeration<String> paramNames = request.getParameterNames();
```

Η διαφορά οφείλεται στο ότι η Java 1.5 έχει εισάγει την έννοια των Generics (όπως εξηγήσαμε στην τάξη).

StringBuffer και StringBuilder

Στη σελίδα 161 χρησιμοποιείται

```
StringBuffer filtered = new StringBuffer
```

Στον αντίστοιχο κώδικα που εξετάσαμε στην τάξη είδαμε:

```
StringBuilder filtered = new StringBuilder(input.length());
```

Η διαφορά βρίσκεται στο ότι η κλάση `StringBuilder` προστέθηκε σαν μια πιο γρήγορη κλάση γιατί δεν χρησιμοποιεί συγχρονισμό νημάτων (`thread synchronization`)

ShowItems

Στη Σελ. 317 η υλοποίηση της κλάσης `ShowItems` έχει ορισμένες διαφορές με την πρόσφατη έκδοση που είδαμε στην τάξη όπου χρησιμοποιούνται `List<String>` και `ArrayList<String>`.