Gliederung vom Benutzerhandbuch

Grobe Übersicht über den Inhalt des Handbuches

1. **Vorwort:** Dieses System soll die Seminarverwaltung mit Hilfe von Computern vereinfachen und diese Software wurde im Studium als Projekt entwickelt.Das System basiert auf Java-Servlet Technologie sowie einer dynamisch erzeugten Web-Oberfläche.
2. **Anmelden:** Im Rahmen der Anmeldung werden noch einige notwendige Daten des User verlangt, um sie für zukünftige und eindeutige Identifizierung zu speichern.
3. **Lesen der Beiträge:** Die Beiträge werden je nach Benutzer-Berechtigung zu Verfügung gestellt, bestimmte Beiträge werden nur für bestimmte Nutzer verfügbar sein.
4. **Einloggen:** Im Rahmen des Einloggens werden bestimmte Daten von dem Benutzer verlangt, diese werden mit den schon gespeicherten Daten (Daten der schon Angemeldeten Nutzers) verglichen.
5. **Vorlesung:** Vorlesungen werden von den Dozenten verwaltet, der Dozent ist berechtigt Daten zu ändern, Skripte hochzuladen, genauso wie der Tutor, aber für die Studenten wird der Zugang zu bestimmten Daten eingeschränkt.
6. **Seminar:** Seminare werden von den Dozenten verwaltet. Die Dozenten haben die Möglichkeit einen Überblick über die aktuellen Zustand des Seminars zu erhalten, die Studenten haben hier die Möglichkeit sich in einen Seminar einzutragen un umzumelden.
7. **Übung:** Hier werden die Übungsserien durch den Dozenten hochgestellt, und die Studenten haben die Möglichkeit sie downzuloaden. Die Dozenten besitzen volle Rechte, um die Übungen zu verwalten.
8. **Forum:** Der Dozent ist Moderator des Forums, je nach Benutzerrechte können Beiträge geschreiben, gelesen oder entfernt werden.
9. **Abmelden:** Hier meldet sich den Benutzer ab.
10. **Stichwortverzeichnis:** **siehe Glossar**

Die Javadoc Dokumentation soll den Nutzer als Handbuch zur Verfügung stehen, wie eine Online Hilfe. (hier stellen wir ein Beispiel vor, wie diese aussehen soll)

import: Die import-Anweisung am Anfang des Listings dient dazu, die Klassse des Pakets java.*.* bekannt zu machen. Ohne diese Anweisung würden den Packages (oder Pakets) vom Compiler nicht gefunden und es gäbe eine entsprechende Fehlermeldung.
Klassen: Klassen sind das wichtigste Strukturierungsmittel objektorientierter Sprachen. Eine Klasse enthält eine Menge von Variablen, die den Zustand von Objekten dieser Klasse beschreiben, und eine Menge von Methoden, die das Verhalten der Objekte festlegen.
public: Alle Elemente einer Klassendefinition können mit Hilfe der aus C++ bekannten Schlüsselwörter public, private und protected in ihrer Sichtbarkeit eingeschränkt werden.
throws: Schlüsselwort throws deklariert Ausnahmen, die während der Methodenausführung auftreten können.
String: In Java werden Zeichenketten durch die Klasse String repräsentiert. Sie bietet Methoden zum Erzeugen von Zeichenketten, zur Extraktion von Teilstrings, zum Vergleich mit anderen Strings und zur Erzeugung von Strings aus primitiven Typen. Der Compiler erkennt und interpretiert String-Literale und ist in der Lage, String-Objekte zu erzeugen und zu verketten.

Die Javadoc wird in der Syntax /** … */ beschrieben.
Einige Bestimmungen für das Quellcodeaussehen für unser Entwicklerteam:

Die Variablennamen in den Kommentaren werden groß geschrieben, alle Substantive werden in den Kommentaren klein geschrieben, Hauptmethoden des Programmes und wichtige Variablen werden kommentiert.
Am Anfang jeder Klasse und Methode ist eine mindestens zweizeilige Beschreibung der Funktionalität vorgesehen.

Prototyp 1

"

```
import java.io.*;
import java.text.*;
import java.util.*;
import org.w3c.dom.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.xml.parsers.*;
import util.HTMLFilter;


public class demo extends HttpServlet {
  /** Prototyp für die Seminarverwaltung. Funktionalität: einloggen und testen des Namens */
  /** und des Paswortes, benutzerrollenabhängige Anzeige von verschiedenen Daten */

    ResourceBundle rb = ResourceBundle.getBundle("LocalStrings");
    private NodeList Entries = null;
    private Document dok = null;

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
         String [][] daten=new String [10][5];  /** daten-feld zum bearbeiten*/
        int check=0;       /** check-variable */
        Document doc;      /** variable fuer xml auslesen */
…
```

"