

Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinich, Sven Pohl, Karsten Krug, Sebastian Neudert
Software-Praktikum, Thema <<Übungsbetrieb>>

Designbeschreibung WebAssign Version 1.9.B20020918:

1.

Allgemeines

WebAssign bietet eine umfangreiche Plattform für den Betrieb einer Universität unterstützendes System. Studierende, Betreuer und Korrekturkräfte werden gleichermaßen unterstützt. Die Erstellung von Aufgaben, das Einsenden von Lösungen, deren Korrektur sowie die Veröffentlichung von Ergebnissen erfolgen über das Internet.

Wichtig ist das dem Student für alle Bereiche der Universität Lehre wirksame Werkzeuge zur Verfügung stehen. Es muss ihm möglich sein sämtliche Studienrelevanten Informationen abzurufen. Dazu zählen die Scripte, Übungsaufgaben, alte Klausuren aber auch Hinweise zum Studium oder Literatur Empfehlungen. Außerdem sollte der Kontakt zu den Korrektoren und/oder Professoren möglich sein um evtl. fragen zum Stoff oder den Übungsaufgaben zu klären. Im WebAssign-System greifen alle beteiligten Nutzer über eine Web-Schnittstelle auf ein zentral installiertes Anwendungssystem zu. Dort werden alle anfallenden Daten und Dokumente in einer Datenbank verwaltet.

Es werden verschiedene Nutzer angelegt:

Der Student greift in der Regel lesend auf die Datenbank zu indem er Script, Übungsaufgaben oder Ergebnisse abrufen. Nur zum Lösungs- Upload soll er schreiben können.

Der Korrektor ruft eingesendete Aufgaben ab und kann die Lösungen nach Korrektur uploaden. Er hat einen begrenzteren Einblick als der Student.

Der Betreuer besitzt Admin ähnliche zugriffsrechte: er verwaltet nicht nur die Aufgabenstellungen sondern kann auch Studenten und Korrektoren einschreiben. Er kann die Lösungen der Studenten und die Korrekturen einsehen.

2.

Produktübersicht

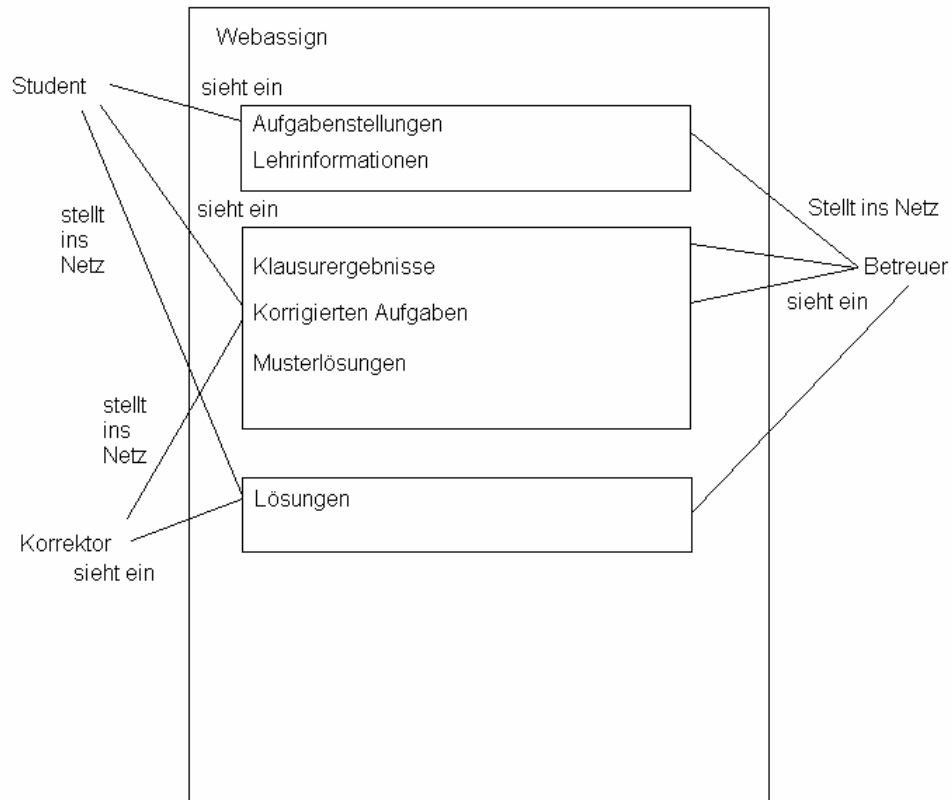
Webassign stellt die Datenbank- Anwendungssoftware um den Studenten zu ermöglichen Lösungen in die Datenbank zu schreiben oder abzurufen.

Hier ist vor allem wichtig das die Rechte ordentlich und wohldefiniert verwaltet werden. (Es sollte nicht passieren das die Musterlösungen dem Studenten vor Abgabetermin dem Studenten zugänglich sind). Die Ausgabe wird ebenfalls formatiert, z.b. werden bei den Klausurergebnissen die Namen ausgeblendet und nur das Ergebnis und die Artikelnummer wird sichtbar sein.

Ausserdem werden noch organisatorische Aspekte zur Verfügung gestellt, die im web verfügbar sein sollen. Hier kann der Betreuer Einschreibungen starten oder wichtige Termine veröffentlichen.

Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinrich, Sven Pohl, Karsten Krug, Sebastian Neudert
Software-Praktikum, Thema <<Übungsbetrieb>>



3.

Grundsätzliche Design-Entscheidungen

Softwahrtechnische Anforderungen:

Für den Betrieb ist ein Webserver von Nöten sein auf dem das Java Servlet und die Servlet Engine laufen wird, ausserdem sollt dieser Server eine Relationale Datenbank ansprechen können. Ein weiterer Linux Server wird benötigt auf dem die https laufen (Apache-Server) und der eine Datenbank anspricht (wahlweise mySql oder informix). Aufgrund der Trennung von Datenbank und Anwendungssoftware kann aber auch eine andere Datenbank benutzt werden.

Hardwaretechnische Anforderungen:

Die Hardware Anforderungen hängen stark mit der zu erwarteten Frequentierung ab. Der Webserver sollte also ausreichende Rechenleistung bringen um die nutzer zu verwalten- der Datenbank Server ist ebenfalls von den Erwartungen abhängig (vor allem die Kapazität).

Für einen Testbetrieb mit max. 10.000 Einsendungen pro Woche reicht ein 300 MHz SPARC und 512 MB Hauptspeicher als Webserver.

Anforderungen an die Servlet Architektur:

Es wird eine Java Servlet Architektur benutzt, das heisst das jeder Benutzer ein Java Script serverseitig aufruft. Um das zu gewährleisten muss für jeden Benutzer auf dem Server ein Verzeichnis angelegt werden in dem dies für jeden nutzer isoliert passieren kann. Die Endwerte der URL werden dabei als Parameter genutzt. Die Ergebnisse werden an WebAssign zur weiteren Nutzung weitergeleitet.

Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinrich, Sven Pohl, Karsten Krug, Sebastian Neudert
Software-Praktikum, Thema <<Übungsbetrieb>>

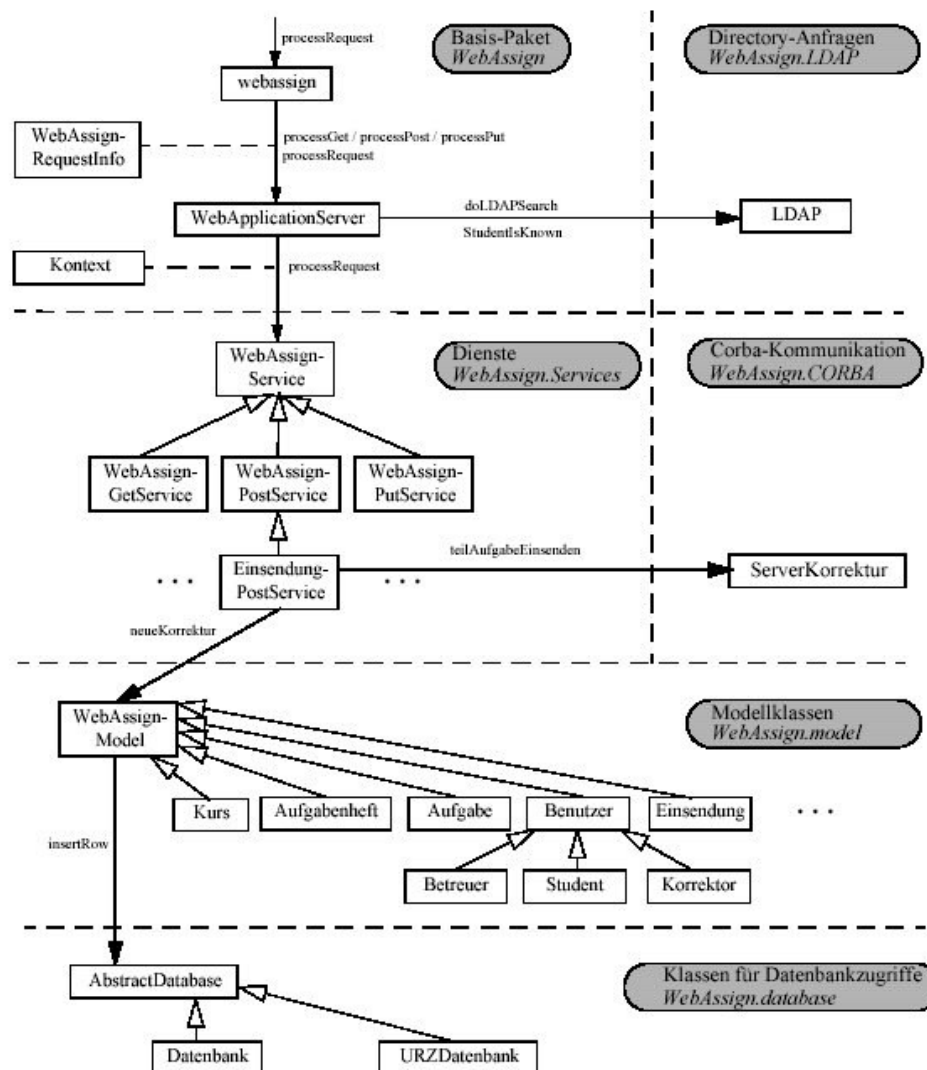
Das Zentrale Servlet:

Die Requests der zentralen Servlets werden zunächst in dem Objekt „WebAssignRequestInfo“ gekapselt und an den WebApplicationServer geleitet. Dort werden die Eingabedaten extrahiert, auf Gültigkeit geprüft und der Nutzer (Korrektor/Student/Betreuer) wird geprüft, die Attribute des Nutzers hinzugefügt.

WebAssign teilt die Basisfunktionalität für alle gewünschten Operationen. Die Services teilen sich auf in Get/Put/Post services.

Der Service Klasse EinsendungPostService kommt eine besondere Bedeutung zu:

Nachdem die Eingesandten Daten in der DB gespeichert wurden soll das Korrektur Modul aufgerufen werden. Dazu wird eine CORBA Schnittstelle (WebAssignCorba) mit dem Korrektur Modul aufgebaut.



Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinrich, Sven Pohl, Karsten Krug, Sebastian Neudert
Software-Praktikum, Thema <<Übungsbetrieb>>

4.

Paket- und Klassenstruktur

Paket WebAssign.database

- AbstractDatabase
 - abstrakte Oberklasse zur Anbindung verschiedener Datenbanken
 - (Informix, MySQL, Oracle)
 - stellt die Basisoperationen für die elementaren DB-Operationen bereit
- DatabaseCheckThread
 - innere Klasse von AbstractDatabase
 - prüft in regelmäßigen Abständen die DB-Connection und restauriert sie, falls nötig
- HISDatenbank
 - bietet zentralen Zugriff auf studentische Stamm- und Leistungsdaten
 - zur Laufzeit darf nur eine Instanz existieren
- URZDatenbank
 - bietet zentralen Zugriff auf die Stammdaten des URZ; insbesondere zur Überprüfung von Belegungen
 - zur Laufzeit darf nur eine Instanz existieren
- InformixDatenbank
 - implementiert AbstractDatabase bezüglich einer Datenbankanbindung zu Informix Datenbankmanagementsystemen
- MysqlDatenbank
 - implementiert AbstractDatabase bezüglich einer Datenbankanbindung zu MySQL Datenbankmanagementsystemen
- OracleDatenbank
 - implementiert AbstractDatabase bezüglich einer Datenbankanbindung zu Oracle Datenbankmanagementsystemen
- Tupel
 - Schnittstelle einer geordneten Menge von Paaren
- QueryEvaluato
 - Interface

Paket WebAssign.CORBA

Die in diesem Package enthaltenen Klassen stellen die Anbindung zu CORBA bereit, und werden hier nicht weiter aufgeführt.

Paket WebAssign.korrektur

Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinich, Sven Pohl, Karsten Krug, Sebastian Neudert
Software-Praktikum, Thema <<Übungsbetrieb>>

Paket WebAssign.korrektur.assistent

- Server
 - Hauptklasse des Korrekturassistenten
 - baut ServerSocket auf und hält Request-übergreifende Informationen vor
- Authorization
 - verkapselt Authorisierungsinformationen, die über das http- Protokoll ausgetauscht werden
- DataTransferStatus
 - protokolliert Staus des laufenden Datenaustauschs mit dem WebAssign-Server
- DataTransferClientInfo
 - verkapselt für einen DownloadClient sichtbaren Callback-Methoden des DataTransferStatus
- Slave
 - zentraler Request-Handler für Korrekturserver
 - verarbeitet GET/ PUT /POST – Requests
- MainServerRequest
 - Oberklasse für alle Requests an den WebAssign-Server (MainServer)
- DownloadClient
 - Schnittstelle der Slave-Klasse beim Download von Korrekturen
- KorrektorDownloadRequest
 - verkapselt einen Request an den WebAssign-Server, der neue Korrekturen anfordert
- UploadClient
 - Schnittstelle der Slave-Klasse beim Upload von Korrekturen
- KorrektorUploadRequest
 - verkapselt einen Request an den WebAssign-Server, der fertige Korrekturen hochlädt
- Einsendungen
 - verkapselt alle zu einem Kurs gehörigen Einsendungen, die über ein spezielles Directory identifiziert werden
- KursInfo
 - verkapselt die Informationen zu einem Kurs
- ConfirmationRequest
 - Bestätigungsrequest, der nach erfolgreichem Download von Korrekturen an den WebAssign-Server zurückgesendet wird
- Config
 - verkapselt die Informationen aus einer kursspezifischen Ini-Datei

Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinrich, Sven Pohl, Karsten Krug, Sebastian Neudert
 Software-Praktikum, Thema <<Übungsbetrieb>>

- Methoden zum Laden und Speichern der Datei sowie Getter und Setter Methoden zum Zugriff auf einzelne Attribute
- ConfigLoader
 - verwaltet mehrere kursspezifische Konfigurationen
 - Methoden zum Wechseln zwischen den Konfigurationen sowie zur Ausgabe einer Html – Selection Box mit allen vorhandenen Konfigurationen
- HTTPConnect
 - stellt http Verbindung her
 - erhält Parameter dafür von einer Instanz von HTTPClient
- HTTPConector
 - baut SSL-fähige Socket-Verbindung auf
 - als Thread realisiert, damit der Aufrufer Timeout setzen kann

WebAssign.korrektur.server

- AbstractKorrekturServer
 - fasst alle gemeinsamen Eigenschaften von Korrekturservern zusammen

WebAssign.korrektur.server.lotse

- WortKorrekturServer
 - implementiert die abstrakte Klasse AbstractKorrekturServer

WebAssign.korrektur.server.pascal

- Test
 - ein Objekt dieser Klasse steht für einen Testlauf des erzeugten Pascalprogramms mit einem konkreten Testfall
 - TestWaechter
 - verwaltet zwei Listen von Tests: alte Tests und neue Tests
 - jeder neu angemeldete Test wird zunächst der Liste neue Tests hinzugefügt
 - bei jedem Ablauf der Wartezeit werden alle Tests abgebrochen sofern sie nicht schon auf normalen Weg abgeschlossen wurden und neue Tests neue Tests werden zu alten Tests
 - auf diese Weise kann kein Test länger laufen als 2*warteZeit
 - PascalBewerter
 - PascalEinsendungsBearbeiter
 - PascalKorrekturServer
 - PascalTextschablone
- Diese Klassen werden bei der automatischen Korrektur von Übungsaufgaben mittels Pascal benötigt.

WebAssign.korrektur.smalltalk

- SmalltalkClass
 - bietet Methoden zur Korrektur von Smalltalk Aufgaben

Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinrich, Sven Pohl, Karsten Krug, Sebastian Neudert
Software-Praktikum, Thema <<Übungsbetrieb>>

- SmalltalkKorrekturServer
 - implementiert die abstrakte Klasse AbstractKorrekturServer

WebAssign.korrektur.utils

- Dateiwaechter Methoden zur Existenzüberwachung
- Dateiwarter einer aktuell bearbeiteten Datei

- EausNBewerter
 - erzeugt neuen 1-aus-N-Bewerter

- Einzelbewerter
 - erzeugt einen neuen Einzelbewerter

- FileAccess
 - Hilfsklasse, die Methoden für einen einfacheren Dateizugriff bereitstellt
 - liefert FileAccessException

- FileAccessException
 - Fehlerbehandlung bei Dateizugriffen

- Intervall
 - Verwaltung eines Intervalls positiver Zahlen

- Keys
 - spaltet String nach mehreren Schlüsselwörtern auf

- LotseZahlenbewerter
 - erzeugt neuen LotseZahlenbewerter

- ProgrammEinzelbewerter
 - erzeugt neuen Einzelbewerter (String oder Zahl)

- StringBewerter
 - erzeugt neuen StringBewerter

- StringUtils
 - bietet Methoden für den Umgang mit Strings

- WordGroups
 - stellt Methoden für den Umgang mit Wortlisten bereit

- WordList
 - erzeugt eine neue Wortliste

- XausNBewerter
 - erzeugt neuen X-aus-N-Bewerter

- ZahlenBewerter
 - erzeugt neuen Zahlenbewerter

Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinich, Sven Pohl, Karsten Krug, Sebastian Neudert
Software-Praktikum, Thema <<Übungsbetrieb>>

Paket WebAssign.services

- Klasse WebAssign Service
 - Basisklasse aller Services.
 - Stellt sämtliche übergeordnete Funktionalität zur Verfügung, insbesondere den Dollar-Variablen-Expander

- Steht in Beziehung zu:

- Klasse WebAssignRequestInfo
 - Zuständig für Login (inkl. Fehlermeldung bei missglückten Login)
- Klasse Kontext
 - Ein Kontext-Objekt speichert alle Informationen aus der Datenbank, die auf bestimmten HTML-Seiten (mit diesem Kontext) bekannt sind.
 - Von der Art des Kontexts hängt es ab, welche Datenbankinformationen bekannt sind (und welche Variablen benutzt werden dürfen).
 - Im Kontext werden allerdings aus Effizienzgründen nur einzelne Tupel gespeichert. Werden ganze Relationen benötigt, so muss der jeweilige Service sich diese selbst aus der Datenbank holen.
- PrintStream
- ByteArrayOutputStream

Paket WebAssign.services.get

- Klasse WebAssign.Get.Service
 - Oberklasse aller GetServices
 - Bildet „Vorbild“ für alle funktionalen Klasse, die mir Services zu tun haben, d.h. das Paket WebAssign.services.get beinhaltet alle Funktionalitäten (wie der Name schon sagt) welche einen Informationsausgabe ermöglichen
 - Zum Beispiel:
 - Klasse AufgabenGetService
 - Ermöglicht die Ausgabe von Aufgabentexten, Musterlösungen, Korrekturhinweisen u.s.w
 - Klasse StudentDataGetService
 - Ermöglicht die Ausgabe von WebAssign-lokale Stammdaten
 - Klasse KursForumZensurGetService
 - Forumsverwaltung
 - Schnittstelle zum Bearbeiten/Loeschen von Forumseiten anzeigen.
 - Klasse VeranstalterGetService
 - Kurse und Querverweise zu allen Veranstaltern auflisten
 - Ermöglicht auch den Zugriff auf DB-Daten
 - Zum Beispiel:
 - Klasse DBExportGetService
 - Formular für den Export von Korrekturen aus der DB anzeigen
 - Ermöglicht auch die statistische Informationsausgabe

Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinich, Sven Pohl, Karsten Krug, Sebastian Neudert
Software-Praktikum, Thema <<Übungsbetrieb>>

- Zum Beispiel:
 - Klasse ErgebnisUebersichtGetService
 - ermöglicht animierte Übersicht über alle Aufgabenhefte und alle Einsendeformen anzeigen

Paket WebAssign.services.post

- Klasse WebassignPostServices
 - Oberklasse
 - Infrastruktur zum Verarbeiten von Post-Requests
 - Enthält u.a. Funktionen zum Email senden, zum Empfangen von Anmeldungsemails von Studenten
 - Zum Beispiel:
 - Klasse EMailPostService
 - E-mail an den WebAssign-Admin senden
 - Klasse AufgabenErstellungPostService
 - ermöglicht:
 - Aufgabe generieren/analysieren
 - Teilaufgabe erzeugen
 - Lösung/Hinweise/Korrekturseite/Quittung erzeugen
 - Änderungen sichern
 - Klasse ForumPostService
 - Beitrag zum Forum einsenden
 - Klasse EMailVersandPostService
 - E-mail an alle Kursbeleger versenden
 - Klasse KorrektorSearchPostService
 - Detailinfos zu einem Korrektor ausgeben
 - Ermöglich auch Anmeldevorgänge
 - Für WebAssign anmelden
- Klasse WebAssignFilePostService
 - Oberklasse
 - Infrastruktur zum File-Upload
 - Zum Beispiel:
 - Klasse EinsendungRTFUploadPostService
 - RTF-Datei ins System hochladen, nach HTML konvertieren und als "Einsendung" speichern
 - Klasse KursRessourceUploadPostService
 - KursRessourcen ins System hochladen, gezippt oder nicht gezippt
- Klasse WebAssignMultipartPostService
 - Oberklasse
 - Infrastruktur zur Behandlung von Multipart-Post-Requests (Kombination von Datei- und normalen Eingabefeldern)

Paket WebAssign.services.put

- Klasse WebAssignPutService

Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinrich, Sven Pohl, Karsten Krug, Sebastian Neudert
 Software-Praktikum, Thema <<Übungsbetrieb>>

- Oberklasse aller PutServices
- Zum Beispiel
 - Klasse KorrektorOnlineUploadPutService
 - Online-Korrekturen in die DB eintragen
 - Klasse KorrektorUploadService
 - Korrekturen vom Korrekturserver entgegen nehmen
 - Klasse KursRessorcePutService
 - Eine neue "Rohseite" ins System hochladen

Paket WebAssign.util

- Enthält u.a. Klassen zur Datenidentifizierung
 - Zum Beispiel
 - Klasse Data
 - Bildet Oberklasse
 - Zum Beispiel
 - Klasse FileData
 - Klasse StringData
- Enthält Klassen zur HTML-Seitenmodellierung
 - Zum Beispiel
 - Klasse HTML-Teilaufgabe
 - Klasse PageInfo
- U.a.
 - Klasse DataAccess
 - implementiert einige nützliche (statische) Datumsfunktionen
 - Klasse DataCache
 - Verwaltet Referenzen auf Objekte vom Typ "Data" über mehrere Requests hinweg. Data-Objekte registrieren sich selbständig und bekommen dabei einen hashCode zugewiesen. Über diesen Wert kann das Objekt später wieder abgerufen
 - Klasse DateTime
 - Kapselt ein WebAssign-Datum. Daten werden als String in der DB gespeichert, müssen aber zur Laufzeit verglichen werden können. DateTime übernimmt die Konvertierung von und nach String und stellt Methoden für den Vergleich von Daten zur Verfügung
 - Klasse ThreadManager
 - Verwaltet alle Threads, die unter WebAssign laufen und beendet sie beim Shutdown des Servlets

Paket WebAssign.util.fileupload

- Enthält u.a. Klasse FileUploadResourceConstants
 - Überwacht und verwaltet den Dateiupload
 - Speichert konstante Daten in den vorhandenen Ressourcen des FileuploadServlets
- Klasse UploadResults

Ueb 14

Christian Witte, Lars Thielecke, Claudia Heinrich, Sven Pohl, Karsten Krug, Sebastian Neudert
Software-Praktikum, Thema <<Übungsbetrieb>>

- Für jede hochgeladenen Datei wird eine Instanz der Klasse erzeugt welche die Transaktion verfolgt und auf dem Server darstellt. UploadResults beinhalten Attribute der Datei
- Klasse UploadException
 - Ausnahmen welche während des Datei uploads passieren werden Teil der Multipart-Message

Paket WebAssign.work

- Beinhaltet die Aufgabenverwaltung
- Enthält u.a. Klasse Aufgabe
 - Repräsentiert eine Aufgabe im WebAssign- System für die automatische Generierung der LOTSE- Aufgaben
 - Oberklasse der „Aufgabenklassen“
 - Zum Beispiel
 - Klasse Begriffe
 - Repräsentiert eine Aufgabe, bei der ein oder mehrere Begriffe überprüft werden
 - Klasse Zuordnung
 - Oberklasse für alle Zuordnungsaufgaben
- Enthält u.a. Klasse AufgabenSeite
 - Die Klasse steht für eine Aufgabenseite mit den dazugehörigen Eigenschaften
- Enthält u.a. Klasse AufgabenTypen
 - Verwaltung der verschiedenen Aufgabentypen und deren Aufgabennamen