Projekt
im Praktikum zur OOS
im Sommersemeter 2016

Projektbeschreibung

&

Programmierrichtlinien Ergänzung (javadoc)

auf

Homepage Download-Bereich

Anforderungen

Funktionale Anforderungen

- **FR1:** Es soll eine Nutzerverwaltung implementiert werden, die
  - neuen Benutzern ermöglicht, sich einzutragen, wobei sie eine UserId und ein Passwort wählen können.
  - bereits eingetragenen Benutzern ermöglicht, sich unter Eingabe der UserId und des Passwortes anzumelden.
- **FR2:** Der Zugang zum System soll sowohl remote, als auch auf dem lokalen Rechner erfolgen.

Technische Anforderungen

- **TR1:** Das System muss auf allen Plattformen ausführbar sein.
- **TR2:** Außerdem darf für die Zugangssysteme lediglich folgendes vorausgesetzt werden:
  - Es ist ein (java-fähiger) Web-Browser installiert.
  - Das System ist mit dem Internet verbunden.

Design- und Implementierungsentscheidungen

- **FR1 und FR2** -> 3-Schichten-Architektur:
  Client, Server, Persistenzschicht
  - remote -> Client/Server-Architektur
  - Benutzer persistent halten -> Persistenzschicht
- **TR1 ->** Java SDK: möglichst große Plattformunabhängigkeit
- **TR2 ->** Zugangssystem als Java-Applet implementiert
  oder mit WebFramework
- zusätzlich Folgendes festgelegt:
  Systemadministrator wird ermöglicht:
  - bereits eingetragene Benutzer zu löschen
  - Datenhaltung in der Persistenzschicht zu initialisieren.

3-schichtige Architektur moderner Informationssysteme

- ***plattformunabhängig***
- ***beliebige Endgeräte***
- ***kein Konfigurationsaufwand***

Client läuft im Browser

Applet

darum:

Sockets

RMI in Verteilte Systeme

- ***plattformunabhängig***
- ***objektorientierte Architektur***

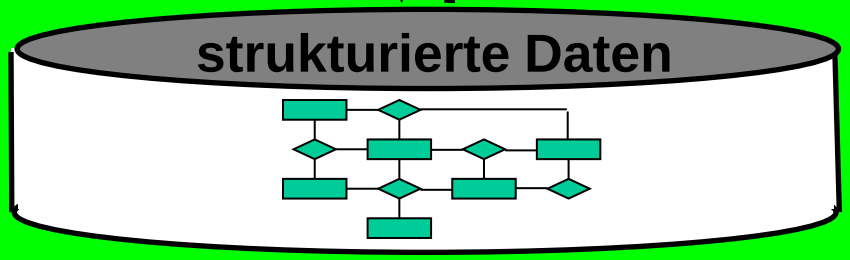
Server: Anwendungslogik

JDBC

- ***verteilte Datenbank:***

hier zu aufwändig

-> hier einfach Dateisystem des PCs



Versuch 2.1: Server

Bereitstellung der Methoden für eine Nutzerverwaltung:

- **`benutzerEintragen(Benutzer benutzer)`**
- **`benutzerOK(Benutzer benutzer)`**
- **`benutzerLöschen(Benutzer benutzer)`**

hierzu Struktur eines Benutzers durch Speicherung von:

- **`UserID`**
- **`Passwort`**

Unterscheidung zwischen Diensten, die jeder nutzen darf (in Interface) und

Diensten, die nur ein Systemadministrator nutzen darf.

Versuch 2.2: Persistenz

Implementierung der persistenten Datenhaltung der Nutzerverwaltung auf dem Dateisystem mittels:

- Serialisierung (Objekt -> Datei)
- Deserialisierung (Datei -> Objekt)

von Objekten

Versuch 3: Oberflächenprogrammierung, Ereignisverarbeitung

Implementierung der Fensteroberfläche durch

Implementierung von drei Swing-Frames

Reaktion auf Ereignisse, wie Mouseklick, ...

durch Java-Ereignissteuerung

Versuch 4: Verteilung des Systems

Trennung in:

- Client
- Server

Implementierung der Kommunikation zwischen Client und Server über Java-Sockets

Versuch 5: Web-Fähigkeit des Systems

Bereitstellung des Systems als WebService

Ausführung in Browser

Multi-User-Fähigkeit

Vorgehensweise

inkrementell, iterativ:

- Grobarchitektur bei Beginn festgelegt
- aber: nicht alle Probleme schon bei Beginn gelöst
- deshalb:
  Änderungen an bereits implementierten Code jederzeit möglich
- darum:
  so programmieren, dass man später leicht ändern kann