Multimedia im Netz –
Wintersemester 2011/12

Übung 11

Medien Informatik München

Betreuer: Sebastian Löhmann

Verantwortlicher Professor: Prof. Dr. Heinrich Hussmann

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Organisatorisches

Zu Blatt 09

JBoss/Eclipse

- Gab es Probleme bei der Einrichtung der Entwicklungsumgebung?

Java Server Faces

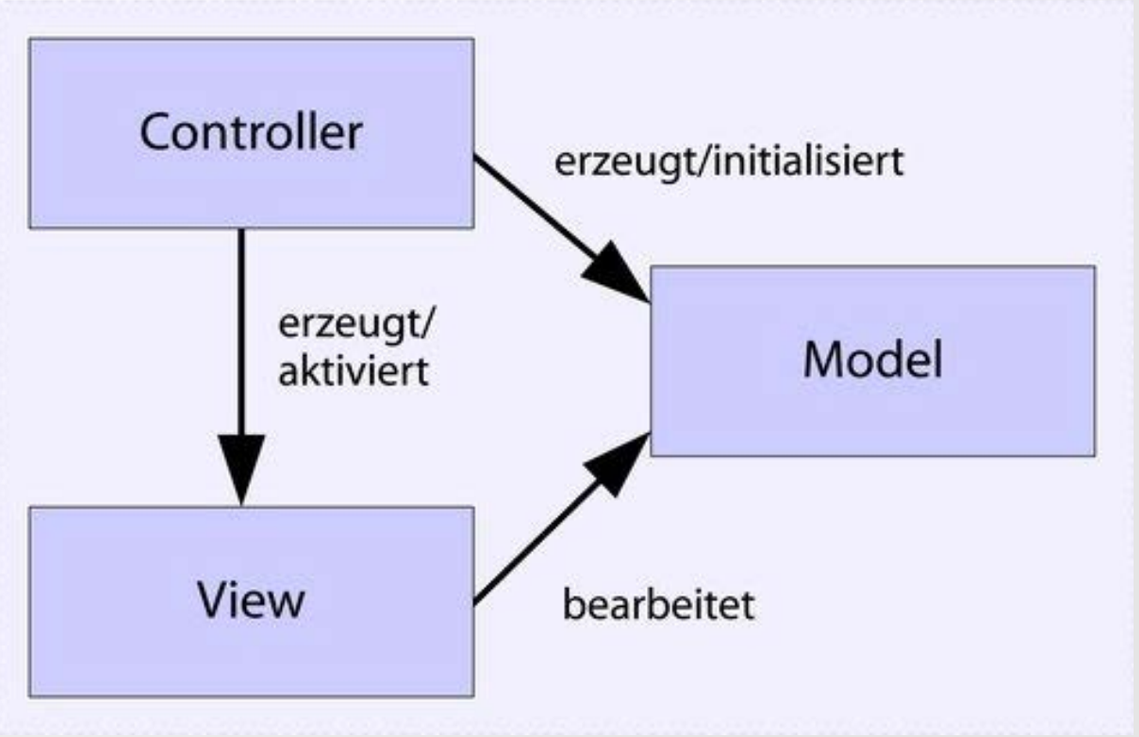
JSF: Wiederholung

- Java Framework für Web-Anwendungen
- aktuelle Version ist 2.0

- Ziel: Trennung von Java-Code und Markup
- folgt der Model-View-Controller-Architektur
- Anwendungslogik kann leicht wiederverwendet werden, View austauschbar

Model-View-Controller-Architektur

- View
  - Präsentation, Formulare, … (z.B. HTML)
  - nimmt Nutzereingaben entgegen (Observer-Pattern)
- Model
  - Datenmodell, Programmlogik (z.B. Java)
  - Konstruktor, Getters, Setters
- Controller
  - Programmsteuerung
  - bearbeitet Nutzereingaben



View: Konzept

- Interface aus Formularen, Buttons und anderen UI-Komponenten
- Unser Beispiel der letzten Woche: hello.xhtml
- In JSF geschrieben, wird aber bei Aufruf vom Server gerendert und als HTML angezeigt
- Test
  - http://mmn.medien.ifi.lmu.de:8080/HelloJSF/hello.jsf
  - Quellcode anzeigen & mit hello.xhtml vergleichen

View: JSF Tag Libraries

- HTML Library (`xmlns:h="http://java.sun.com/jsf/html"`)
  - Forms, input and output
  - Grouping, tables
  - Commands (buttons, links)
  - Messages

- Core Library (`xmlns:f="http://java.sun.com/jsf/core" `)
  - Views, subviews
  - Listeners
  - Data converters
  - Validators
  - Internationalisation

View: JSF HTML Tags

- <h:inputText …><*/h:inputText*>
- <h:commandButton ...<*/h:commandButton*>
- Weitere Komponenten unter

http://docs.oracle.com/cd/E17802_01/j2ee/javaee/javaserverfaces/2.0/docs/pdldocs/facelets/index.html

- oben links die h-Library auswählen

View: Navigation

```
<h:commandButton value="Welcome Me"
    action="welcome"></h:commandButton>
```

- durch `action="welcome"` wird automatisch welcome.jsf aufgerufen

View: Eingabewerte & Bean-Variablen

```
<h:inputText
   value="#{helloBean.name}"></h:inputText>
```

- bei Absenden des Formulars wir automatisch der Setter für die Property „name“ in HelloBean.java aufgerufen und der vom Benutzer eingegebene Wert gesetzt
- helloBean bezeichnet Namen der Java-Bean mit erstem Buchstaben als Lower Case

Managed Bean

```
import javax.faces.bean.ManagedBean;

@ManagedBean

public class HelloBean {
    private String name;
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

Controller & Model

- in unserem (simplen) Beispiel ist Controller & Model in HelloBean.java vereinigt
- In komplexeren Anwendungen befindet sich die Logik in eigenen Klassen (Model)
- ManagedBean ist in der Regel der Controller
- Objekte des Models werden durch den Controller erzeugt und Methoden werden durch den Controller aufgerufen

View: Ausgaben

```
#{helloBean.name}
```

Ist Kurzform für

```
<h:outputText value="#{helloBean.name}" />
```

Listener

```
<h:commandButton
   id="example"
   actionListener="#{controller.myHandler} " />
```

- bestimmt Methode, die im Controller implementiert ist und das Event verarbeitet

```
public void myHandler(ActionEvent ae) { … }
```

- Methode im Controller, die das Event verarbeitet

(Sehr) Hilfreiches Tutorial

- http://www.coreservlets.com/JSF-Tutorial/jsf2/
- Wichtige Abschnitte
  - JSF 2.0: Introduction and Overview
  - JSF 2.0: Programming Basics
  - The JSF 2.0 Expression Language
  - Managed Beans I
  - Handling GUI (Application) Events

Blatt 10

Tic Tac Toe

- Aufgabe: das Spiel Tic Tac Toe in JSF entwickeln
- Auf Trennung von View, Controller und Model achten

Tic Tac Toe: Hinweise

- Spielfeld-Layout:

0	1	2
3	4	5
6	7	8

- gewonnen, wenn ein Spieler 3 Felder in einer Reihe für sich gewonnen hat:

{ {0, 1, 2}, {3, 4, 5}, {6, 7, 8}, {0, 3, 6}, {1, 4, 7}, {2, 5, 8}, {0, 4, 8}, {6, 4, 2} }

Danke ☺ Fragen?

Medien Informatik München

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN