

# JSP vs. PHP

Ein persönlicher Vergleich



## Agenda

- JSP-Ursprung
- Warum dann noch PHP ansehen?
- Der erste Eindruck
- Ein Beispiel in JSP & PHP: mein Flugbuch
- Der Versuch eines Vergleichs
- Fazit



# Bemerkung

- alles persönliche & subjektive Einschätzung
- kein „Glaubenskrieg“ für/gegen JSP/PHP
  - oftmals äußere Zwänge entscheidend
  - persönliche Vorlieben der Entwickler
  - „Wert“ der Einarbeitung und des verfügbaren Know-How nicht unterschätzen
  - Qualität entsteht nur durch qualitativ gute Arbeit der Entwickler (Werkzeuge, Technologien unterstützen dies nur)

# Warum mache ich JSP

- war zu diesem Zeitpunkt auf OpenVMS verfügbar
- bietet alle Vorteile, solche Aufgaben zu realisieren und dabei unabhängig von der Plattform zu werden
- Aufbau von neuem Know-How (Java, Tomcat, Apache, JSP, ...)
- JSP ist eine „gute“ Technologie (= Java!) und damit für mich geeignet

# Warum PHP ?

- große Anzahl PHP-Anwendungen im Netz
- PHP als Alternative zu JSP einsetzbar (?)
- PHP besser als JSP (?)
- PHP inzwischen auch für OpenVMS verfügbar
- über eigenen Tellerrand schauen

# Infrastruktur

- PHP:
  - Apache
  - PHP (4.3.x)
  - Texteditor
- IDE (z.B. PHPEdit), Debugger, Tools
- Dokumentation ok
- alles im Netz verfügbar, ohne Probleme Download
- Einfacher zu Installieren
- JSP:
  - Apache
  - Tomcat
  - Java
  - Texteditor
- IDE (z.B. NetBeans), Debugger, Tools
- Dokumentation ok
- alles im Netz verfügbar
- größere „Installation“
- Erfahrung Installation notwendig

# PHP zum 1.

- „andere“ Sprache (z.B. „.“ anstelle „+“)
- damit auch eine neue/zusätzliche Sprache
- viele Konstrukte übernommen (++, strcmp)
  - Einarbeitung einfacher (C, Java angenommen)
- unheimlich viele Funktionen
  - viele Dingen einfach machbar
  - viel zu unübersichtlich
- Objekte/Klassen (PHP4/PHP5)
- Bücher: VIELE – u.a. auch „schlechte“
  - scheint aber Gewohnheiten zu bilden

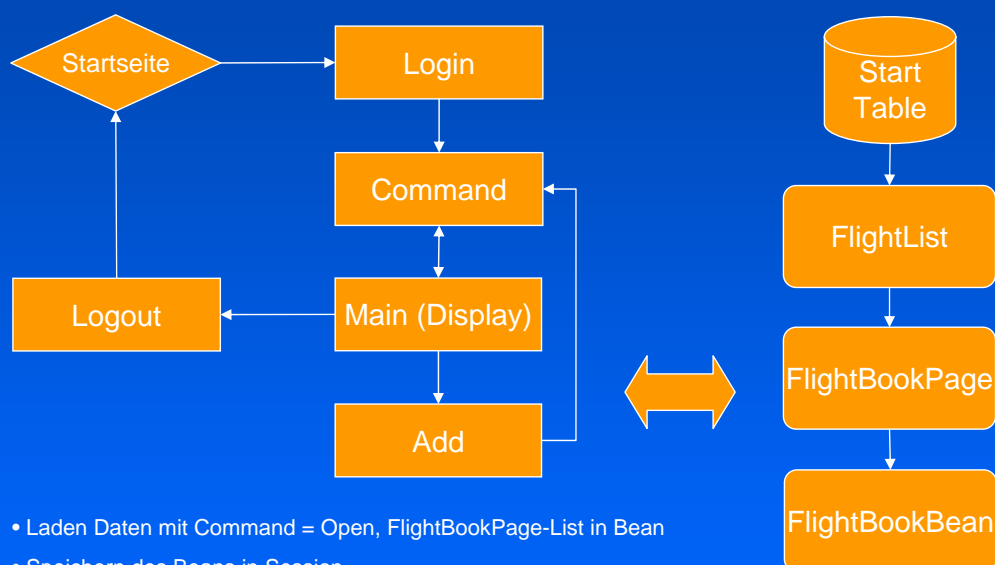
## Beispiel Flugbuch

- Rahmen war meine „Spielwiese“ aus unseren Projekten
- wollte etwas sinnvolles machen, was aber nicht zu groß wird
- Anzeige Flugbuch (Daten waren gerade da)
- als „Fingerübung“
- [Jörg's Flightbook: Login](#) PHP
- [Jörg's Flightbook: Login](#) JSP

# Beispiel Flugbuch

- MS SQL Datenbank mit Tabelle Starts
- Java-Klasse Flight
- Java-Klasse FlightList – DB-Interface
- Java-Klasse FlightBookPage – Struktur der Liste mit Navigationsmöglichkeiten
- Java-Klasse FlightBookBean – Interface zwischen HTML-/Web-Interface und Daten (FlightBookPage(s))
- Startseite Flightbook.html
- Aktionsseiten: Command.JSP, Main.JSP, Add.JSP
- Servlets: Login, Logout

# Beispiel Flugbuch



- Laden Daten mit Command = Open, FlightBookPage-List in Bean
- Speichern des Beans in Session
- Zerstörung des Beans mit Logout
- Command: Interface Request <-> Bean
- Main = Anzeige

# Und das ganze als PHP

- Java-Klassen -> PHP class objects
  - reines Umschreiben (Syntaxanpassung)
  - PHP4-Klassen mit Einschränkungen
- JSP -> PHP
  - anderes Handling Session (Cookies & Firewall!)
  - gleiches Prinzip, andere Syntax
  - JSP-Tags -> „Scriptcode“ in PHP
- Servlets -> PHP
  - analog JSP -> einfach umzuschreiben (hier auch „keine“ Funktionalität)
- Umschreiben: nicht immer optimaler PHP-Code, aber schnell und einfach machbar

# Vergleich JSP - PHP

- beides verwendbar, gleiche Plattformen
- JSP mehr Infrastruktur
  - größerer Aufwand Installation, Wartung
  - größere Projekte machbar (EJB, ...)
- JSP = Java!
  - eher Wiederverwendbarkeit
  - **sauberes Klassenkonzept** (trotz PHP5)
  - Java auch außerhalb JSP verwendbar (-> Anzahl Sprachen im Projekt)
  - Konfiguration Anwendung für mehrere Instanzen (web.xml)

# Fazit

- Es entscheiden eher äußere Faktoren, was man nimmt!
  - Verfügbarkeit
  - Anzahl Sprachen im Projekt
  - Kenntnisse des Teams
  - Vorlieben (wenn man es sich leisten kann)
- PHP – Prototyping-System (?)
  - Beispiel (?) -> [update\\_account.php.03.txt](#)
  - Beispielhaft – nicht unbedingt
  - Trennung Darstellung – Logik nicht im Vordergrund
- JSP – Profisystem (?)
  - klares Konzept Darstellung – Logik & Support

# Fragen & Diskussionen

- Fragen
- Hinweise
- Tipps

