JSP Webspiel - Hangman Übung

APR 5BHITS 04.11.2012

Pawlowsky

Pawlowsky

Inhalt

[Aufgabenstellung 2](#_Toc339913073)

[Designüberlegung 3](#_Toc339913074)

[Arbeitsdurchführung 4](#_Toc339913075)

[Resultate/Niederlagen 4](#_Toc339913076)

[Tests 4](#_Toc339913077)

[Quellen 4](#_Toc339913078)

# Aufgabenstellung

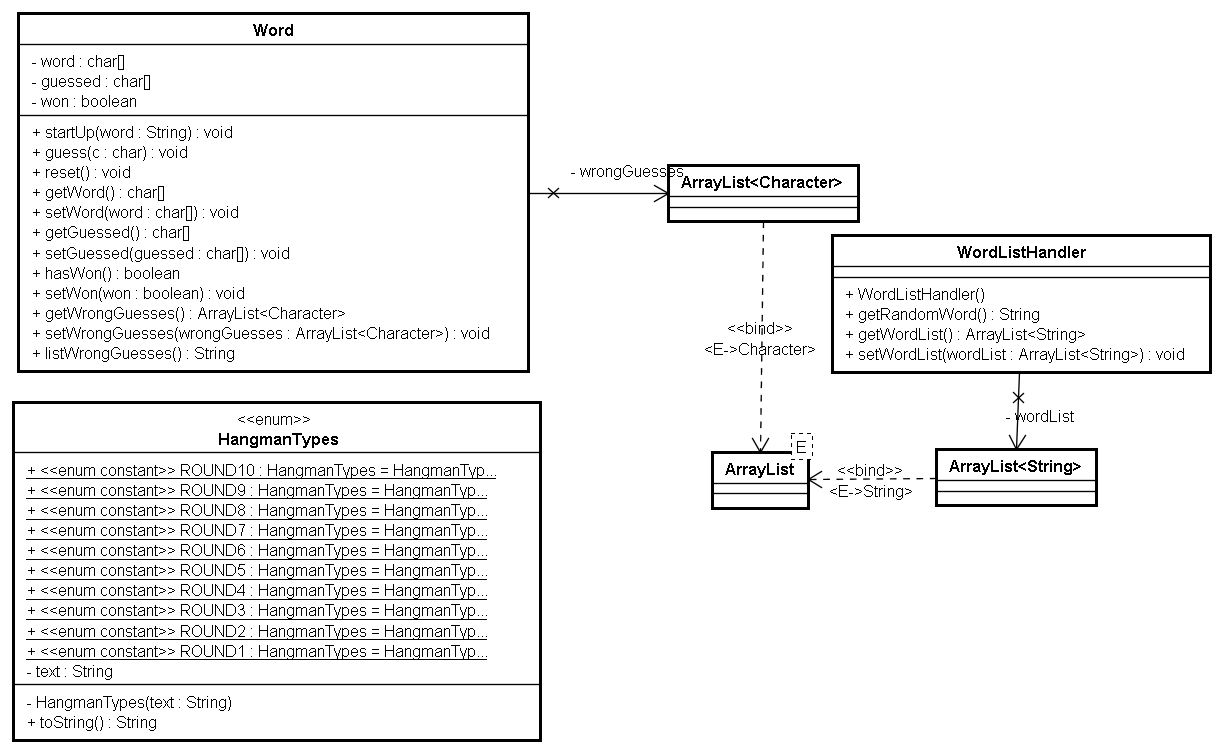
Es ist ein einfaches Web-Spiel mit Hilfe von Servlets, Java Server Pages und dem MVC Pattern zu programmieren. Die Details zum Spiel sind frei wählbar, allerdings sollte ein sinnvoller Spielablauf möglich sein. Welches Spiel zu programmieren ist, ist von der Katalognummer abhängig:

* **Hangman: 1, 4, 7, 10, 13**
* TicTacToe: 2, 5, 8, 11, 14
* Mastermind: 3, 6, 9, 12

Abgabe:  
Gesamtes Projekt inklusive Dokumentation und \*.war  
  
Punkte (16):  
Ausführliche Sourcedokumentation (4), Ressourcennutzung (4), Undeployment (2), Deployment (4), Protokoll (2 .. -4)

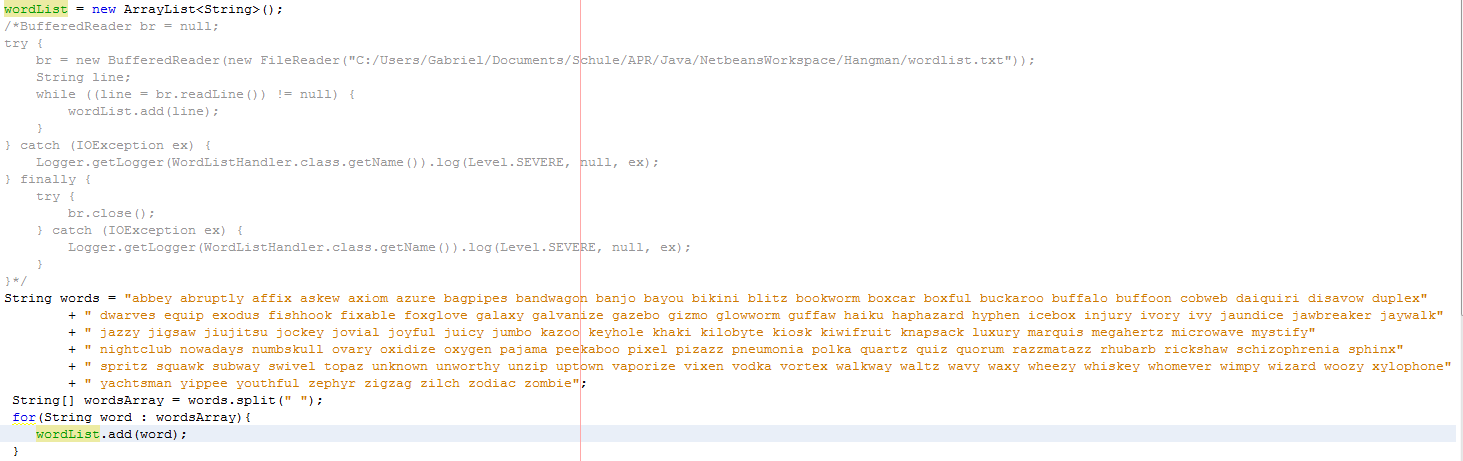
# Designüberlegung

Mein grundsätzliches Klassen-Design sieht in etwa so aus:



Wie man sehen kann habe ich eine Klasse die das zu erratende Wort (und damit den Großteil des Spiels) darstellt. Weiters habe ich einen WordListHandler, der dafür zuständig ist die möglichen Wörter nach denen es zu raten gilt zu initialisieren. Und zu guter Letzt gibt es ein enum, dass das Aussehen des Hangmans zu den unterschiedlichen Zeiten während des Spiels beinhaltet.

Um innerhalb einer JSP-Applikation auf eine Datei zugreifen zu können, muss diese Datei auf den Localhost liegen. Ich habe diese Möglichkeit ausprogrammiert, allerdings nehme ich nicht an, dass Sie mein Textfile mit den Wörtern beim Test meiner Applikation auf den Localhost legen werden und daher habe ich die Möglichkeit auskommentiert und nun einen String mit allen Wörtern in die Applikation geladen.



Auf das saubere Undeployment musste ich bei dieser Applikation nicht achten, da ich keinen Thread oder eine ähnliche Struktur benutzt habe die das saubere Undeployment gefährden könnte.

**Aufwandsabschätzung**

Protokoll:1,5h

Implementierung der Oberfläche: 2,5h

Implementierung der Spiellogik: 1,5h

**Gesamt: 4h**

# Arbeitsdurchführung

**Pawlowsky:**

01.11.2012 Implementierung der Oberfläche 2,5h

01.11.2012 Implementierung der Spiellogik 1,5h

05.11.2012 Protokoll 1,5h

**Gesamt: 5,5h**

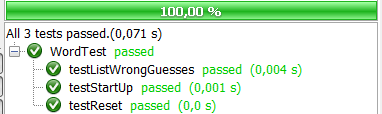
Resultate/Niederlagen

Ich hatte anfangs das Problem, dass bei einem Neustarten der Session noch ein Klick auf ein anderes Objekt notwendig war, bevor die Session wirklich neu gestartet und alle zuvor getätigten Einstellungen zurückgesetzt wurden.

Dies habe ich durch eine Reset-Methode in meiner Word-Klasse gelöst. Nun muss ich die Session nicht mehr neu starten, da bei einem Klick auf den Reset-Button einfach nur alle spielbezogenen Einstellungen auf ihren Startwert gesetzt werden und ein neues Wort aus der Wortliste geholt wird.

# Tests

Hier die Bestätigung, dass alle Tests erfolgreich durchführbar waren:



# Quellen

<http://netbeans.org/kb/docs/web/quickstart-webapps.html>