



JAVA STARS 2005 - Sun Microsystems Award

Projekttitel: Deutschlandspiel

Übersicht

Teamnummer	134
------------	-----

Schulnummer	182140
-------------	--------

Schulname	Börde Berufskolleg
-----------	--------------------

Schulform	Berufskolleg
-----------	--------------

Name des Teams	IT324
----------------	-------

Projektname	Deutschlandspiel
-------------	------------------

Projektkurzbeschreibung (max. 4 Zeilen)	Eine Deutschlandkarte in einem webinterface über die man "spielerisch" geographische Aspekte wie z.B. Bundesländer, deren Hauptstädte, Flüsse, deren Lage usw. kennen lernen kann.
--	--

Unterrichtsfach	Anwendungsentwicklung
-----------------	-----------------------

Gruppengröße	4 Personen
--------------	------------

Altersgruppe	21-24
--------------	-------

Inhaltsverzeichnis

ALLGEMEINE FRAGEN ZUM WETTBEWERB.....	5
WIE HABEN SIE/HABT IHR VON DEM WETTBEWERB "JAVA STARS 2005" ERFAHREN?.....	5
HABT IHR AN ANDEREN WETTBEWERBEN TEILGENOMMEN?.....	5
PROJEKTBESCHREIBUNG.....	6
PROJEKTBESCHREIBUNG.....	6
PROJEKTIDEE.....	6
THEMA.....	6
UNTERRICHTSFACH.....	6
NUTZEN FÜR DEN UNTERRICHT.....	6
ZEITAUFWAND.....	6
ARBEITSTEILUNG.....	7
ARBEITSUMGEBUNG.....	7
SYSTEMANFORDERUNGEN.....	7
TESTUMGEBUNG.....	7
PROBELÄUFE.....	7
INSTALLATION / START DES PROGRAMMS.....	8
LÖSUNGSKONZEPT.....	12
AUFBAU DER LÖSUNG.....	12
EINGESETZTE VERFAHREN.....	12
PROGRAMM-ARCHITEKTUR.....	13
FUNKTIONSKOMPLEX: PROGRAMMSTART / INITIALISIERUNG.....	13
ZUSTÄNDE.....	13
METHODEN.....	13
FUNKTIONSKOMPLEX: DATENBANKVERBINDUNG.....	14
ZUSTÄNDE.....	14
METHODEN.....	14
FUNKTIONSKOMPLEX: „ANTWORTEN“.....	15
ZUSTÄNDE.....	15
METHODEN.....	15
FUNKTIONSKOMPLEX: „KARTENAUFBAU“.....	16
DIE ZUSTÄNDE:.....	16
ERLÄUTERUNG BUNDESLÄNDER_MAPPING_PAGE.....	16
FUNKTIONSKOMPLEX: „FRAGEBEREICH“.....	17
ZUSTÄNDE.....	17
METHODEN.....	17
BENUTZERSCHNITTSTELLE.....	18
KONZEPT.....	18
FUNKTIONSBEREICH 1 – FRAGE.....	19
FUNKTIONSBEREICH 2 – INTERAKTIVE KARTEN.....	19
FUNKTIONSBEREICH 3 – INFORMATIONEN / LÖSUNG.....	19
FUNKTIONSBEREICH 4 – PROGRAMMHILFE.....	19
REFERENZEN.....	20
ANLAGEN.....	21
DOKUMENTIERTER QUELLKODE.....	21
EIDESSTATTLICHE ERKLÄRUNG.....	21
EINVERSTÄNDNISERKLÄRUNG DER ERZIEHUNGSBERECHTIGTEN.....	21

Allgemeine Fragen zum Wettbewerb

1.1 Wie haben Sie/habt ihr von dem Wettbewerb "JAVA STARS 2005" erfahren?

Unser Projektteam hat durch unseren Berufsschullehrer Herrn Oberreuter Wettbewerb Java Stars erfahren. Er selbst ist Lehrer der gymnasialen Oberstufe für Mathematik und Informatik. Er begleitete schon im Jahr 2004 eine Oberstufenklasse durch diesen Wettbewerb. Unsere Klasse ist somit der Nachfolger im Jahre 2005

Habt Ihr an anderen Wettbewerben teilgenommen?

Nein, dies ist unser erstes Projekt. Das Team kannte sich vorher auch nur aus dem Berufsschulunterricht.

Projektbeschreibung

Projektbeschreibung

Das Projekt ist eine serverseitige Anwendung, welche auf Servlets zurückgreift. Das Programm läuft mit Hilfe eines Servlet Containers (in unserem Fall: Tomcat) und ist von jedem Client mit Hilfe eines Browsers (IE, Mozilla) nutzbar.

Das Projekt ist ein Spiel, in dem Bundesländer und andere geographische Punkte in Deutschland spielerisch rausgefunden werden können.

Eine Datenbank (bei uns: Firebird) enthält die Informationen über diese Punkte und diese werden Dynamisch über Servlets erstellt.

Projektidee

Die Projektidee hatte unser Mitschüler aber Nicht-Projektteilnehmer Sebastian Stroer. Hierzu an dieser Stelle ein Dankeschön!

Thema

Das Thema lautet: Geographie – spezifisch für Deutschland.

Unterrichtsfach

Wir empfehlen den Einsatz des Programms im Unterricht aller Klassen der Sekundarstufe I bzw. der Grundschule in den Fächern Erkunde und Geographie.

Nutzen für den Unterricht

Aufgrund der graphisch animierten Darstellung der Deutschlandkarte, werden die Schülerinnen und Schüler auf spielerische Art und Weise mit den geographischen Gegebenheiten ihrer Heimat vertraut gemacht.

Mit Hilfe der Weboberfläche werden die Schülerinnen und Schüler dazu begeistert verschieden Aufgabenstellungen intuitiv zu bearbeiten.

Außerdem bietet das Programm die Möglichkeit, dass sowohl der aktuelle Spieler als auch weitere Zuschauer mit Tips und Hilfen jederzeit während einer Raterunde interaktiv am Geschehen teilnehmen können.

Da durch unser Programm eine visuell bessere Vorstellung der Bundesländer und Ländergrenzen ermöglicht wird, wird somit durch das bildliche Denken ein höherer Lerneffekt erzielt.

Zeitaufwand

Neben unserer eigentlichen beruflichen Tätigkeit als Auszubildende in verschiedenen IT-Unternehmen, haben wir uns über den Berufsschulunterricht hinaus jede Woche und am Wochenende privat getroffen. Somit ist jedes Projektmitglied auf einen Zeitaufwand von 70 Stunden gekommen. Gemeinsam haben wir also etwas mehr als 280 Stunden in unser Projekt investiert.

Arbeitsteilung

Zuständigkeiten wurden flexibel eingeteilt. So, dass jedes Teammitglied für das andere einspringen könnte, falls jemand verhindert bzw. erkrankt ist.

Folgende Zuständigkeiten kristallisierten sich jedoch heraus:

Ingo Meyer	Entwicklung des Gesamtkonzeptes für die Java Umsetzung, Design der Klassen, DB-Handling, start_mapping, DB+Entwurf
Denis Piepenbreier	DB anlegen + Füllung, question Klassen, Doku
Joachim Riepegerste	HTML-Seiten, grafisches Kartenmaterial, Koordinaten, Answer Klasse
Christian Pohle	Gestaltung, Doku, DB-Admin, Koordinaten, Mapping Klasse

Arbeitsumgebung

Genutzt wurde eine Windows 2000 SP4 Plattform. Als Java Editor wurde NetBeans in der Version 4.1 mit integriertem Tomcat Server genutzt.

Zu Beginn wurde BlueJ genutzt, was sich jedoch im Laufe des Projektes als zu unkomfortabel bewies. Dies ist vorallem durch die fehlende Servletunterstützung und somit nötige händische Installation von der Sun Servlet API und der Konfiguration des Tomcat Containers zu begründen.

Systemanforderungen

Das Programm ist eine Servlet Anwendung und kann von aktuellen Browsern wie z.B. Internet Explorer ab der Version 5, Mozilla Firefox oder Opera ab der Version 7 genutzt werden.

Testumgebung

Die Testumgebung sind die unter 2.8 Geräte.

Probeläufe

Die ersten Probeläufe waren nur von mässigem Erfolg gekrönt.

Ein Hauptproblem war das Einbinden der externen Library, um eine Verbindung zur Firebird Datenbank aufbauen zu können.

Des weiteren stellte sich heraus das ein Debuggen wie bei einem „normalen“ Programm nicht möglich war, da es nicht möglich war in die Unterklassen zu springen im Debugging Modus von NetBeans.

Somit musste ein Workaround her, um zu sehen wo es bei den Unterklassen zu Problemen kommt.

Dieser bestand darin, das alle Fehlermeldungen als Parameter zurückgegeben wurden und es somit möglich war sie im Browser auszugeben beim durchlaufen des Servlets.

Installation / Start des Programms

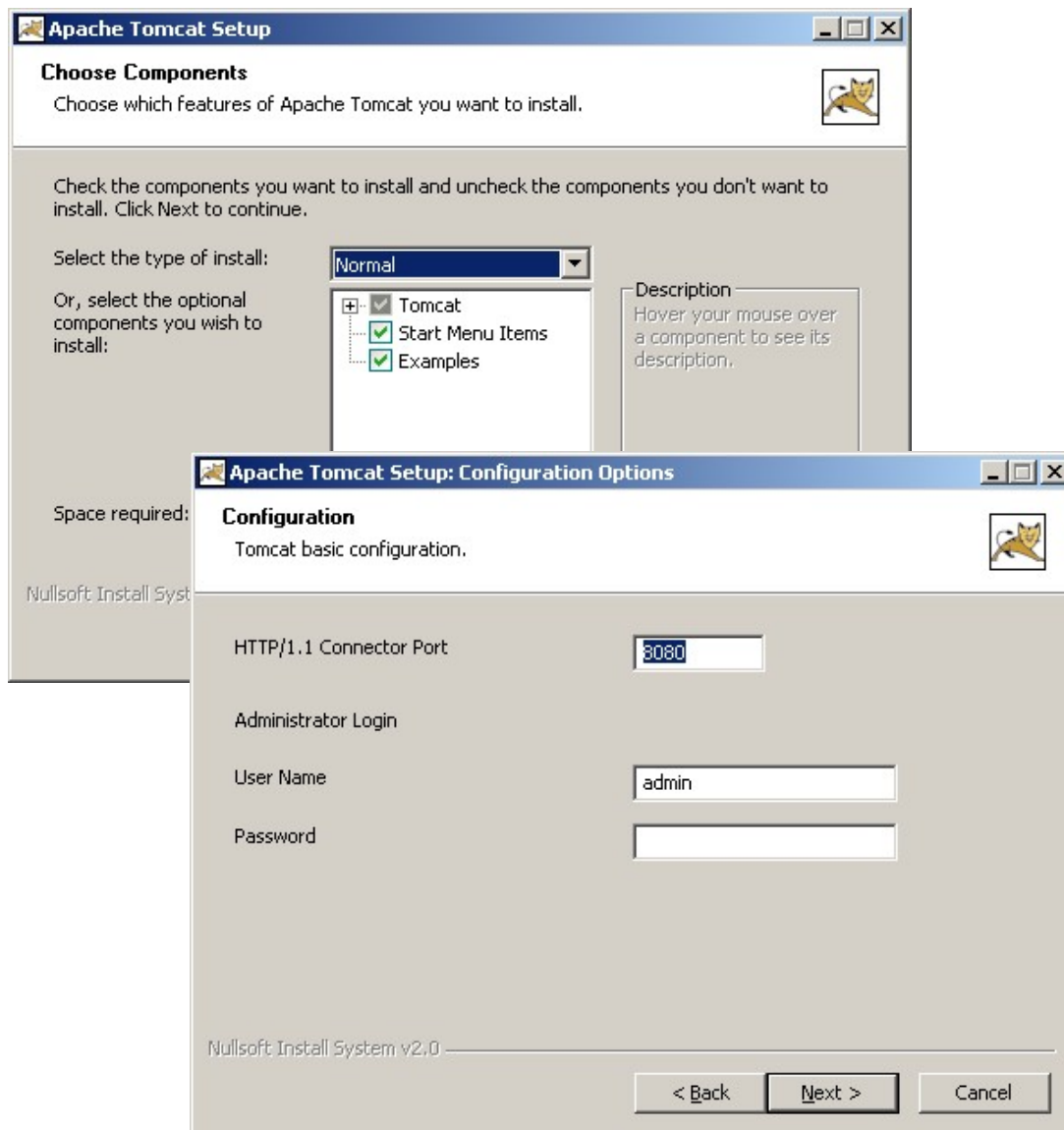
Server Installation:

Die Installation des Servers benötigt einen Servlet Container, in dem die Java-Klassen abgelegt werden.

Die Installation wird in diesem Dokument mit Hilfe eines Tomcat Containers erläutert:

Programm Download: <http://jakarta.apache.org/tomcat/>

Die Installation selbst ist nicht schwierig. Eine Standard Installation mit den in den Screenshots angegebenen Daten ist ausreichend.



Ein Kennwort ist nicht nötig. Der Port ist für den späteren Aufruf vom Webclient wichtig.

Stellen Sie bitte sicher, wenn das Programm über das Internet angeboten werden soll, dass etwaige Sicherheitseinstellungen an ihrer Firewall geändert werden müssen.

Im lokalem Netzwerk (LAN) ist dies zu 99,9 % kein Problem. Wenn Sie sich unsicher sind fragen sie bitte den zuständigen Netzwerkverantwortlichen.

Prüfen Sie bitte die korrekte Installation des Tomcats. Dazu öffnen sie einen Browser und tragen folgende Adresse in die Adresszeile ein:

<http://localhost:8080>

Die „8080“ ist die Portnr. die sie im Tomcat konfiguriert haben.

Wenn die Installation erfolgreich war, dann sollten sie folgenden Bildschirm sehen:



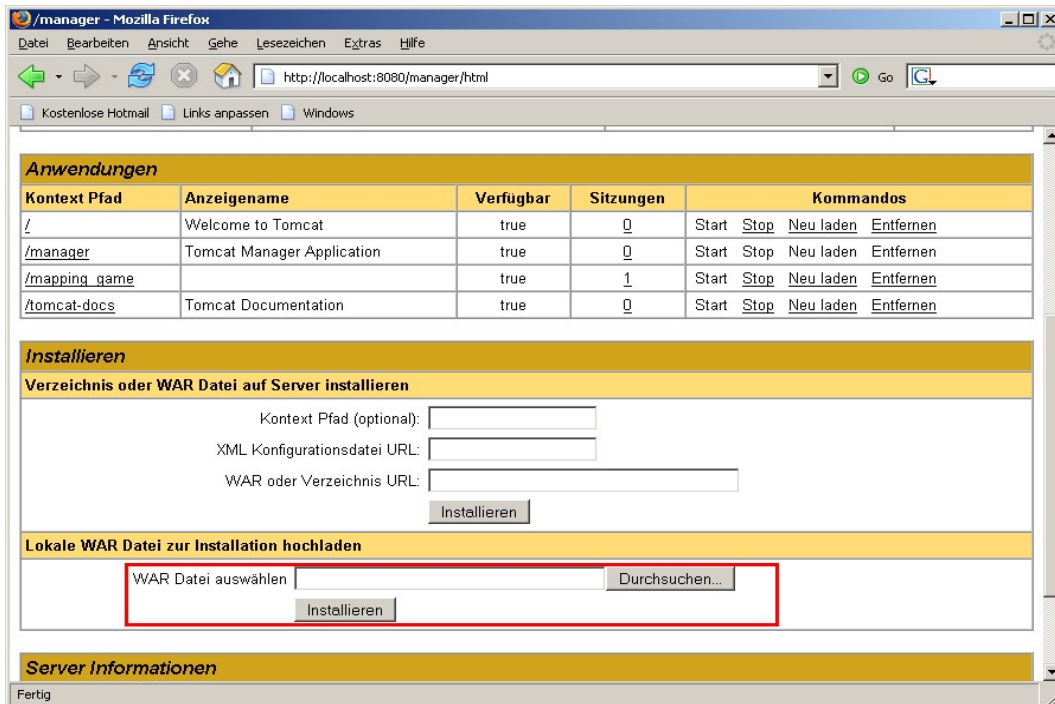
Sollte es Probleme geben, stellen Sie im System Tray sicher, dass der Tomcat Dienst auch gestartet ist. Dieser Fehler tritt meist auf.



Einrichtung des Tomcat Servers mit WAR-Datei

Wählen sie auf der Startseite im Menu „Administration“ den „Tomcat Manager“ aus.

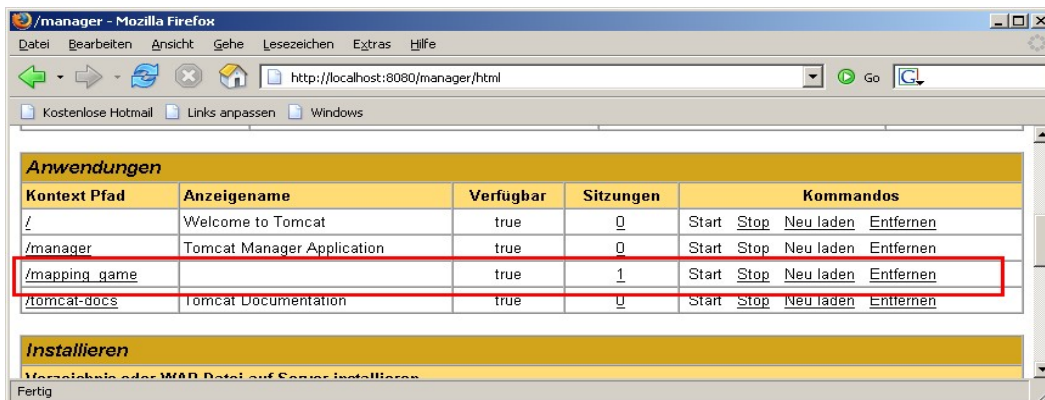
Sie werden anschließend nach Benutzernamen und Kennwort gefragt. Dies sind die aus der Installation angegebenen Daten!



Klicken sie anschließend auf „Durchsuchen“ und wählen sie die Datei „MAPPING_GAME.WAR“ aus. Installieren sie diese anschließend mit einem Klick auf „Installieren“.

Anschließend wird unter „Anwendungen“ ein Link zu der Applikation angezeigt.

Dieser Link ist die Startadresse des Programmes.



Datenbank Server Installation:

Laden sie unter <http://sourceforge.net/projects/firebird> den entsprechenden Datenbankserver für ihr Betriebssystem herunter.

In dieser Dokumentation ist die Konfiguration und Installation für Windows beschrieben, bei anderen Betriebssystemen sollte dieses ähnlich sein.

Wir empfehlen die Installation des Firebird 1.5, da das Deutschlandspiel auf dieser Datenbankversion basiert.

Rufen Sie das „Setup“ auf und installieren Sie den Firebirddatenbank Server mit den vorgegebenen Einstellungen!

Weitere Einstellungen sind nicht nötig.



Datenbank Anbindungs Installation

Die mitgelieferte Opensource JDBC Connection Library "firebirdsql-full.jar" benötigt eine Implementierung im Java Runtime Environment.

Hier zu kopieren sie diese Datei in den Ordner „<JRE>\lib\ext.

Somit ist die Bibliothek in Java eingebunden.

Datenbankdateiablage

Kopieren sie die MAPPING_DATA.GDB in das Verzeichnis

C:\Programme\mappinggame !

Dies ist unbedingt erforderlich!

Programmstart:

Nutzen sie die unter „Einrichtung des Tomcat Servers mit WAR-Datei“ konfigurierte Startadresse

Lösungskonzept

Aufbau der Lösung

Die Lösung besteht aus drei Kernbereichen:

- Fragebereich, in der eine dynamische Frage generiert wird
- Antwortbereich, in der Antwort bzw. weitergehende Informationen angezeigt werden
- Kartenbereich, in der Interaktiv eingegriffen werden kann

Eingesetzte Verfahren

Genutzt wurden unterschiedliche Verfahren:

Servlets:

Die Übergabe von Parametern läuft über die Servlet Parameterübergabe in Java.

HTML:

Der Seitenaufbau der 3 Kernbereiche funktioniert über die Beschreibungssprache HTML, in dieser wurde auch eine unsichtbare Tabelle implementiert.

Polygon Koordinaten:

Um die Bereiche der Grafik zu spezifizieren wurden in HTML Polygon Koordinaten eingebettet. Diese wurden mit dem Shareware Tool „MapThis“ erzeugt.

Die Koordinaten wurden anschließend in den HTML Tag eingefügt.

Grafik:

Die Grafik wurde von Joachim Riepegerste mit Hilfe von Boardmitteln von Windows bearbeitet (Paint!!!). Schattierungen und Bereiche wurden von ihm eingefärbt.

Datenbank:

Eine Firebird Datenbank (OpenSource) realisiert die Informationsverwaltung. Mit Hilfe dieser können die Seiten dynamisch gestaltet und Informationen ausgegeben werden.

SQL:

Durch SQL (Structured Query Language) wurden Informationen aus der Datenbank abgefragt. Durch SQL konnten spezifisch Informationen ermittelt werden.

Programm-Architektur

Funktionskomplex: Programmstart / Initialisierung

start_mapping
+id : int -id_ad : int -status : string -login_status : string -id_answer_typ : int -user_id : int -user_name : string -user_pw : string
+processRequest() +doPost(Zoll Servlet req, aus Servlet res) +doGet() +getServletInfo()

Zustände

Die eigentliche Servlet-Klasse und Hauptkomponenten im Klassengefüge. Sie fängt die do und get Anfragen an den HTTP -Server ab. Und leitet diese zur processRequest Methode in der Klasse. Dort werden dann die Übergabe Parameter verarbeitet und somit ja nach Ablauf eine völlig neue Seite generiert.

Methoden

Do und Get Methode fangen die Anfragen des Clients an den HTTP Server ab und leiten sie an die processRequest Methode weiter. In dieser Methode werden die Übergabeparameter ausgewertet und dann ja nach Ablauf auf die Unterklassen Question, Mapping und Answer zugegriffen bzw. Sie als Objekte eingebunden.

Funktionskomplex: Datenbankverbindung

DbHandle
+Status : String
+Wert : String
+Colum_count : int
+row_count : int
-treiber_db_load() : bool
-db_connection_on() : bool
-db_connection_off() : bool
+frage() : string
+antwort_text() : bool
+random_id() : int
+Antwort_kordinaten() : string
+Antwort_infos() : string
+pruefe_antwort() : string
+Antwort_image() : string
+user_login() : string
+answer_typ_class() : string
+question_typ_class() : string
+mapping_typ_class() : string

Zustände

Diese Klasse ist die Schnittstelle zwischen der Datenbank und dem Servlet bzw. Den Page_builder Klassen. Alle benötigten Daten werden mit Hilfe der Methoden in dieser Klasse aus der Datenbank gelesen.

Methoden

Die Methoden sind weitestgehend selbst erklären und haben immer einen Rückgabewert welcher dann von den Page-Bulider Klassen weiter Benutzt wird.

Funktionskomplex: „Antworten“

Hauptstadt_answer_page
-answer_page : string
-antwort : string
-status : string
+bulid_answer_page() : <nicht spezifiziert>

Bundeslaender_answer_page
-answer_page : string
-antwort : string
-status : string
+bulid_answer_page() : <nicht spezifiziert>

Fluss_answer_page
-answer_page : string
-antwort : string
-status : string
+bulid_answer_page() : <nicht spezifiziert>

Zustände

Die Klassen zum Antwortbereich „<Lokation>_answer_page.java“ erstellen mit Hilfe des DB_handling Funktionskomplexes je nach Status eine HTML Seite. Diese HTML Seite zieht eine Frage aus der Datenbank und generiert somit dynamisch den Webseitenanteil für den Fragebereich.

Die Attribute haben folgende Bedeutungen:

answer_page	beinhaltet den HTML Quellcode für den Webseitenanteil
antwort	beinhaltet den Text zur richtigen Antwort aus der Firebird Datenbank (Inhalte wie Einwohnerzahlen, Schulden, Parteien, ...)
status	beinhaltet 3 unterschiedliche Stati zur Kontrolle (neuer Aufbau, Richtig geklickt, falsch geklickt)

Methoden

In dieser Klasse existiert nur eine Methode. Diese baut auf Basis der Zustände den Webseitenanteil für den Antwortenbereich auf.

Funktionskomplex: „Kartenaufbau“

Hauptstadt_mapping_page
-mapping_page : string
-status : string
+bulid_answer_page() : <nicht spezifiziert>

Bundeslaender_mapping_page
-mapping_page : string
-status : string
+bulid_answer_page() : <nicht spezifiziert>

Fluss_mapping_page
-mapping_page : string
-status : string
+bulid_answer_page() : <nicht spezifiziert>

Die Zustände:

Um den Status des aktuellen Spiels herauszufinden, haben wir uns für 3 unterschiedliche Status entschieden.

„**neu**, **weiter_richtig** und **weiter_falsch**“

Steht der Status auf „**neu**“, wird per Zufallsgenerator eine neue Frage aus der Datenbank gestellt und die Deutschlandkarte wird neu und leer aufgebaut.

Ist der Status „**weiter_richtig**“, stimmen Frage ID und Koordinaten ID überein. Das geklickte Bundesland wird rot eingefärbt und die korrekte Antwort aus der Datenbank dargestellt.

Bei „**weiter_falsch**“ stimmen Frage ID und Antwort ID nicht überein. Ist dies der Fall, wird der Status auf „**neu**“ gesetzt und das Spiel beginnt von vorne.

Erläuterung Bundeslaender_Mapping_Page:

Die **Bundeslaender_Mapping_Page** besteht aus dem Aufbau der Deutschlandkarte vor und nach der Eingabe des Users. Beim Start des Spiels wird die leere Karte mit den Koordinaten der Bundesländer erstellt.

Nachdem der User auf einen Bereich in der Deutschlandkarte geklickt hat, werden die angeklickten Koordinaten mit den Koordinaten in der Datenbank verglichen. Diese Koordinaten haben eine feste ID, welche mit der ID aus der **Bundeslaender_question_page** verglichen wird. Stimmen die IDs überein, wird das Bundesland rot eingefärbt und die richtige Antwort ausgegeben.

Stimmen die beiden IDs nicht überein, so wird das Spiel erneut gestartet.

Funktionskomplex: „Fragebereich“

Hauptstadt_question_page
-question_page : string
-frage : string
-status : string
+bulid_answer_page() : <nicht spezifiziert>

Bundeslaender_question_page
-question_page : string
-frage : string
-status : string
+bulid_answer_page() : <nicht spezifiziert>

Fluss_question_page
-question_page : string
-frage : string
-status : string
+bulid_answer_page() : <nicht spezifiziert>

Zustände

Die Klassen zum Fragebereich „<Lokation>_question_page.java“ erstellen mit Hilfe des DB_handling Funktionskomplexes je nach Status eine HTML Seite. Diese HTML Seite zieht eine Frage aus der Datenbank und generiert somit dynamisch den Webseitenanteil für den Fragebereich.

Die Attribute haben folgende Bedeutungen:

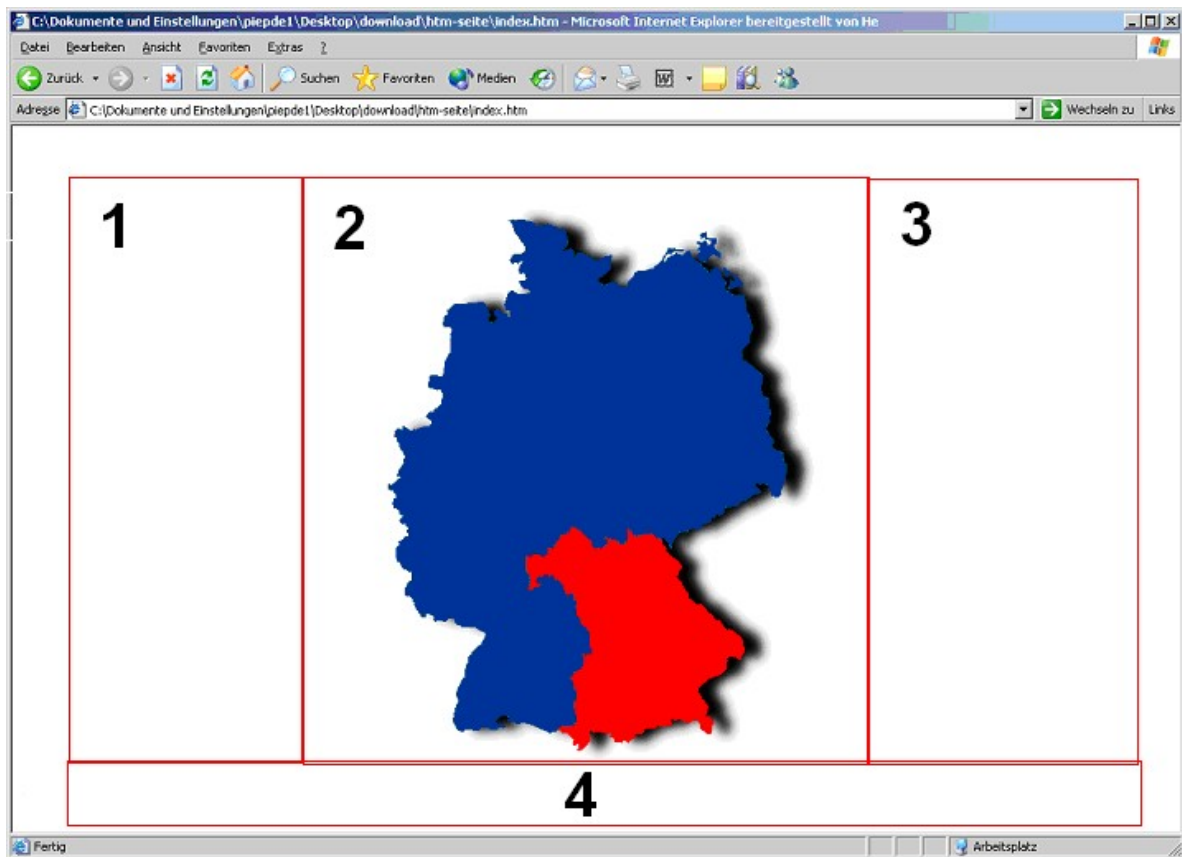
question_page	beinhaltet den HTML Quellcode für den Webseitenanteil
frage	beinhaltet die Frage aus der Firebird Datenbank
status	beinhaltet unterschiedliche Status zur Kontrolle

Methoden

In dieser Klasse existiert nur eine Methode. Diese baut auf Basis der Zustände einen Webseitenanteil für den Fragebereich auf.

Benutzerschnittstelle

Konzept



Die Benutzerschnittstelle ist ähnlich einem Webauftritt von Konzernen und Firmen (siehe Abbildung oben). Sie besteht aus mehreren Teilbereichen, ähnlich HTML-Frames.

Jeder Bereich steht für eine bestimmte Aufgabe. Diese werden nun im Detail beschrieben:

Funktionsbereich 1 – Frage

In diesem Bereich wird eine Frage gestellt. Dies könnte beispielsweise eine Frage nach einer bestimmten Stadt sein.

Funktionsbereich 2 – interaktive Karte

In diesem Bereich ist der Benutzer dazu aufgefordert interaktiv am Geschehen teilzunehmen. Er soll mit dem Mauszeiger auf die richtige Lösung zeigen und mit einem Klick bestätigen.

Funktionsbereich 3 – Informationen / Lösung

Nach dem ein Benutzer einen Mausklick abgegeben hat wird im Bereich 3 eine Information ausgegeben, ob die „Antwort“ wahr oder falsch ist.

Ist die Antwort richtig, so kann man mit einem Klick auf „weitere Informationen“ Daten über die Lokation angezeigt werden.

Mögliche Daten sind: Stadtgröße, Flusslänge o.ä.

Funktionsbereich 4 – Programmhilfe

In diesem Bereich kann eine weitere Online Hilfe angezeigt werden. Ein Klick auf „Hilfe“ genügt.

Referenzen

[1] www.wikipedia.de

[2] Java4U Buch (EINS Verlag)

[3] JSP Cookbook O'Reilly Verlag

Anlagen

Dokumentierter Quellcode

<separate Datei einreichen. Den Programmtext bitte ausdrucken, dabei aber auf *nicht* selbst geschriebene Teile (z.B. verwendete Funktionen der Entwicklungsumgebung, automatisch generierter Programmtext, etc.) verzichten.>

Eidesstattliche Erklärung

<separater Ausdruck einreichen>

Einverständniserklärung der Erziehungsberechtigten

<separater Ausdruck einreichen>