



JAVA STARS 2006 - Sun Microsystems Award

Projekttitel: Der Wahlverfahren-Vergleicher

Übersicht

Bundesland	Niedersachsen
------------	---------------

Teamnummer	211
------------	-----

Schulnummer	65043
-------------	-------

Schulname	Käthe-Kollwitz-Schule Hannover
-----------	--------------------------------

Schulform	Gymnasium
-----------	-----------

Name des Teams	Polis12
----------------	---------

Projektname	Der Wahlverfahren-Vergleicher
-------------	-------------------------------

Projektkurzbeschreibung (max. 4 Zeilen)	Unser Programm behandelt die unterschiedlichen Verfahren zur Vergabe der Sitze in einem Parlament. Das Programm errechnet nach der Eingabe verschiedener Parteien und deren Wählerstimmen die Anzahl der Sitze, die sie im jeweiligen Parlament bekommen würde.
---	---

Unterrichtsfach	Politik
-----------------	---------

Gruppengröße	3 Schüler
--------------	-----------

Alter der Schüler/innen	17 Jahre / 17 Jahre / 17 Jahre
-------------------------	--------------------------------

teilnehmende Mädchen	0
----------------------	---

Inhaltsverzeichnis

1	ALLGEMEINE FRAGEN ZUM WETTBEWERB.....	5
1.1	WIE HABEN SIE/HABT IHR VON DEM WETTBEWERB "JAVA STARS 2006" ERFAHREN?.....	5
1.2	HABT IHR AN ANDEREN WETTBEWERBEN TEILGENOMMEN?.....	5
1.3	BEI WELCHEN WETTBEWERBEN HABT IHR BISHER DAS PROGRAMM SCHON VORGELEGT?	5
2	PROJEKTDESCHEIBUNG	6
2.1	PROJEKTDESCHEIBUNG	6
2.2	PROJEKTIDEE	6
2.3	THEMA.....	6
2.4	UNTERRICHTSFACH.....	6
2.5	NUTZEN FÜR DEN UNTERRICHT.....	6
2.6	ZEITAUFWAND	6
2.7	ARBEITSTEILUNG	6
2.8	ARBEITSUMGEBUNG	7
2.9	JAVA-APPLETS.....	7
2.10	TESTUMGEBUNG	7
2.11	PROBELÄUFE.....	7
2.12	INSTALLATION / START DES PROGRAMMS.....	7
3	LÖSUNGSKONZEPT	8
3.1	AUFBAU DER LÖSUNG.....	8
3.2	EINGESETZTE VERFAHREN.....	8
4	PROGRAMM-ARCHITEKTUR	10
4.1	ÜBERSICHT	10
4.2	FUNKTIONSKOMPLEX/KLASSE 1: WAHLENGROB2.JAVA / START.JAVA	12
4.2.1	Zustände.....	12
4.2.2	Methoden.....	12
4.3	FUNKTIONSKOMPLEX/KLASSE 2: STIMMENEINGABE.JAVA	13
4.3.1	Zustände.....	13
4.3.2	Methoden.....	13
4.4	FUNKTIONSKOMPLEX/KLASSE 3:DHONTRECHNUNG.JAVA.....	14
4.4.1	Zustände.....	14
4.4.2	Methoden.....	14
4.5	FUNKTIONSKOMPLEX/KLASSE 4: NIEMEIERRECHNUNG.JAVA.....	15
4.5.1	Zustände.....	15
4.5.2	Methoden.....	15
4.6	FUNKTIONSKOMPLEX/KLASSE 4: AUSWERTUNG.JAVA	16
4.6.1	Zustände.....	16
4.6.2	Methoden.....	16
5	BENUTZERSCHNITTSTELLE	17
5.1	KONZEPT.....	17
5.2	ANZAHL DER PARTEIEN	17
5.3	PARTEIENDEFINITION.....	17
5.4	AUSGABE NACH D'HONDT	17
5.5	AUSGABE NACH HARE/NIEMEYER	17
5.6	AUSWERTUNG.....	17
6	REFERENZEN.....	18
7	ANLAGEN.....	19
7.1	DOKUMENTIERTER QUELLKODE	19

7.2	EIDESSTATTLICHE ERKLÄRUNG	19
7.3	EINVERSTÄNDNISERKLÄRUNG DER ERZIEHUNGSBERECHTIGTEN	19

1 Allgemeine Fragen zum Wettbewerb

1.1 Wie haben Sie/habt ihr von dem Wettbewerb "JAVA STARS 2006" erfahren?

Wir (der Informatik-Kurs 12/13 der Käthe-Kollwitz-Schule Hannover) wurden von unserem Lehrer auf den „JAVA STARS 2006“-Wettbewerb aufmerksam gemacht.

1.2 Habt Ihr an anderen Wettbewerben teilgenommen?

In dieser Gruppenkonstellation haben wir noch nie an einem vergleichbaren Wettbewerb teilgenommen. Ein Mitglied unserer Gruppe, Markus Rothenhöfer, hat bereits am „Bundeswettbewerb Informatik“ teilgenommen.

1.3 Bei welchen Wettbewerben habt Ihr bisher das Programm schon vorgelegt?

Wir haben unser Programm bisher bei keinem weiteren Wettbewerb eingereicht.

2 Projektbeschreibung

2.1 Projektbeschreibung

Ein Programm für den Politik-Unterricht soll anschaulich und in Interaktion mit dem Benutzer zeigen, wie Verfahren zur Sitzverteilung nach einer Wahl funktionieren und welchen Einfluss sie auf die Mehrheitsverhältnisse in einem parlamentarischen Gremium haben.

Die Gruppe will mit einem Programm für den Politik-Unterricht anschaulich demonstrieren, wie Verfahren zur Sitzverteilung nach einer Wahl funktionieren und welchen Einfluss sie auf die Mehrheitsverhältnisse in einem parlamentarischen Gremium haben. Der Benutzer soll dabei auch aktuelle Daten eingeben können.

2.2 Projektidee

Die Idee zu unserem Projekt ist im Rahmen unseres Informatik-Kurses entstanden und war somit ein Produkt aller teilnehmenden Schüler.

2.3 Thema

Bestimmung der Sitzverteilung in einem Parlament nach einer Wahl.

2.4 Unterrichtsfach

Das Programm eignet sich ideal für den Politik-Unterricht, da es die Sitzverteilung in einem Parlament nach dem Hare/Niemeyer-Verfahren, als auch nach dem d'Hondt-Verfahren errechnet.

2.5 Nutzen für den Unterricht

Oftmals ist es schwer für die Politik-Lehrkraft ein geeignetes Medium zu finden, um den Schülern die verschiedenen Sitzverteilungs-Verfahren nahe zu bringen. Hier bietet unser Programm einen großen Vorteil, da es auf anschauliche Art und Weise die Unterschiede zwischen dem Hare/Niemeyer-Verfahren und dem Verfahren nach d'Hondt aufzeigt. Häufig entscheidet die Wahl des Verfahrens über die Mehrheit der Sitze in einem Parlament. Unser Programm veranschaulicht bei solchen Situationen wie sehr die Wahl des Verfahrens das Endergebnis beeinflusst.

2.6 Zeitaufwand

Für das gesamte Projekt benötigten wir, bei 3 Arbeitsstunden in der Woche, ca. ein halbes Jahr.

2.7 Arbeitsteilung

Ursprünglich wollten wir die komplette Arbeit auf alle drei Teilnehmer gleichermaßen aufteilen. Nach einigen Überlegungen entschieden wir uns jedoch dafür, die Entwicklung des eigentlichen Programmes auf zwei Schüler aufzuteilen und die Dokumentation vom dritten Mitglied ausführen zu lassen.

2.8 *Arbeitsumgebung*

Das Programm wurde von uns auf einem Desktop-PC mit Microsoft Windows XP erstellt. Die Entwicklung verlief unter der aktuellsten JAVA-Version (1.6).

2.9 *Java-Applets*

Das Programm wurde als Applet programmiert und ist in allen gängigen Browsern lauffähig.

2.10 *Testumgebung*

Das Programm wurde unter dem Betriebssystem Microsoft Windows XP ausgiebig getestet. Allerdings funktioniert es nur mit der neuesten JAVA-Version einwandfrei.

2.11 *Probeläufe*

Der Benutzer wird erst einmal dazu aufgefordert die Anzahl der Parteien und anschließend die zugehörigen Wählerstimmen einzugeben. Anschließend berechnet das Programm die zugeteilte Anzahl der Sitze im Parlament. Zuerst werden dem Benutzer die Ergebnisse nach dem d'Hondt-Verfahren aufgezeigt und anschließend die Ergebnisse des Hare/Niemeyer-Verfahrens. Zu guter Letzt zeigt das Programm noch eine fertige Auswertung der verschiedenen Verfahren.

2.12 *Installation / Start des Programms*

Das Programm lässt sich ganz einfach per Doppelklick im Browser starten. Eine Installation ist nicht nötig.

3 Lösungskonzept

3.1 Aufbau der Lösung

Dem Anwender präsentiert sich nach Ausführung des Programms der erste Schritt „Einleitung“. Nach dem Betätigen des „Weiter“-Buttons wird der Anwender zur Eingabe der Parteienanzahl aufgefordert.

Nun gelangt der Anwender zum zweiten Schritt. Dieser verlangt vom Anwender eine Benennung der einzelnen Parteien und die jeweiligen Wählerstimmen. Außerdem lassen sich in diesem Schritt noch immer Parteien hinzufügen oder entfernen.

Wurden nun alle Eingaben gemacht und der nächste Schritt durch Drücken des „Weiter“-Buttons geöffnet, gelangt der Benutzer auf die nächste Seite. Hier werden die Ergebnisse nach dem d'Hondt-Verfahren tabellarisch aufgezeigt. Die Sitzverteilungsreihenfolge wird mit Hilfe einer Animation für alle Schüler leicht verständlich gemacht. Es entsteht ein Überblick über den Wahlausgang, wenn er nach dem d'Hondt-Verfahren ausgeführt worden wäre.

Im nächsten Schritt erhält der Benutzer Ausgabe der Ergebnisse nach dem Hare/Niemeyer-Verfahren. Auch hier bietet eine übersichtliche Tabelle den nötigen Überblick.

Im letzten Schritt schließlich erhält der Benutzer eine Auswertung und einen Vergleich der beiden angewandten Auszählverfahren. Hier werden eventuelle Unterschiede verdeutlicht und die Beeinflussung der Wahl durch die Wahl des richtigen Auszählverfahrens aufgezeigt.

Wir haben uns für diese Art der Veranschaulichung entschieden, da wir sie für übersichtlich und gut strukturiert halten. In den einzelnen Schritten begleiten den Benutzer kurze und erklärende Texte zu seinem jeweiligen Schritt. Das Programm wurde von uns so konzipiert, dass nicht viel Know-How benötigt wird um es zu verstehen. Damit ist es ideal für den Einsatz in Schulklassen geeignet.

3.2 Eingesetzte Verfahren

Wir haben uns für eine übersichtliche Gestaltung durch einen modularen Aufbau entschieden. Das Hauptprogramm (`wahlengrob2.java`) verwaltet den Programmablauf und fügt die einzelnen Panels (Navigationsleiste oben, Weiterbutton unten und Inhalte) ein. Die eigentlichen Inhalte der einzelnen Schritte/Seiten werden in Klassen ausgelagert und vom Hauptprogramm hinzugeladen. Dies gewährt – auch wenn wir anfangs mit vielen Problemen dadurch zu kämpfen hatten – eine optimale Übersichtlichkeit und einen professionellen, strukturierten Aufbau.

Meistens wird von Programmteil zu Programmteil eine (oder mehrere) Klasse(n) mit übergeben. Das ermöglicht einen leichten Zugriff auf alle Methoden und Attribute.

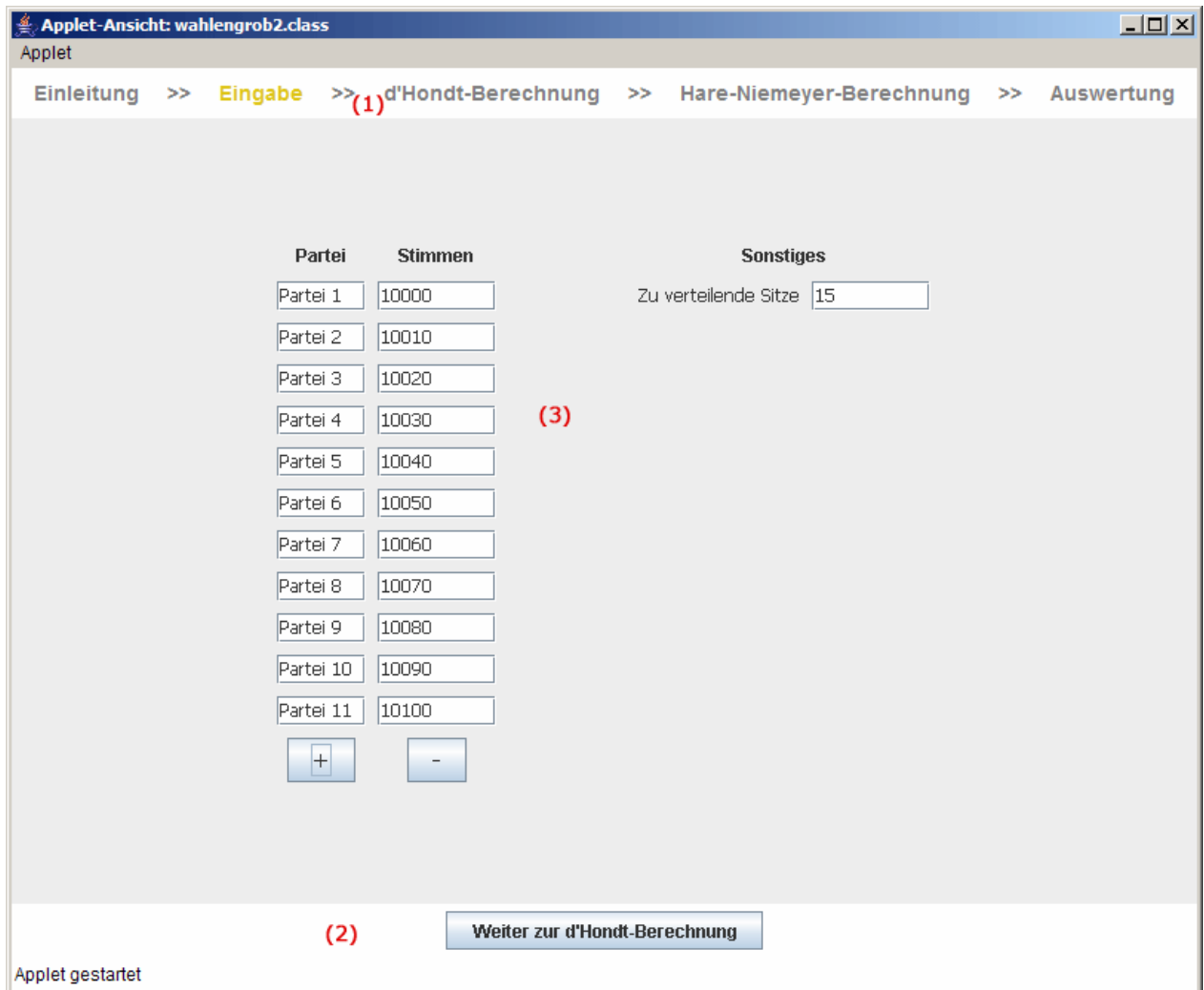
Zur Sortierung von Arrays haben wir den gängigen Algorithmus modifiziert und auf ArrayList-Objekte umgestaltet. Die konsequente Verwendung von ArrayList-Objekten hat uns viele Schleifen erspart und den Code um einiges übersichtlicher gemacht.

Um den Schülern das d'Hondt-Verfahren zu veranschaulichen, haben wir uns entschieden eine Animation einzubauen. Diese wird über einen Timer gesteuert und ersetzt HTML-Code und generiert ihn neu.

Besonders Aufwändig war die Realisierung von dynamischen Eingabefeldern bei der Eingabeseite. Es ist möglich während der Laufzeit Parteien hinzuzufügen oder wieder zu entfernen. Somit ist das Programm auf maximale Flexibilität ausgelegt.

4 Programm-Architektur

4.1 Übersicht



Der obige Screenshot soll den grundlegenden Aufbau verdeutlichen. Die Navigationsleiste (1) und die Leiste mit dem Weiterbutton (2) sind fest in der Hauptklasse (wahlengrob2.java) definiert und deshalb in jedem Teil des Programmes sichtbar. Sie sind auch optisch klar von dem Inhalt getrennt. Indem wir für sie einen weißen Hintergrund ausgewählt haben. In der Mitte befindet sich der in andere Klassen ausgelagerte Inhalt (3).

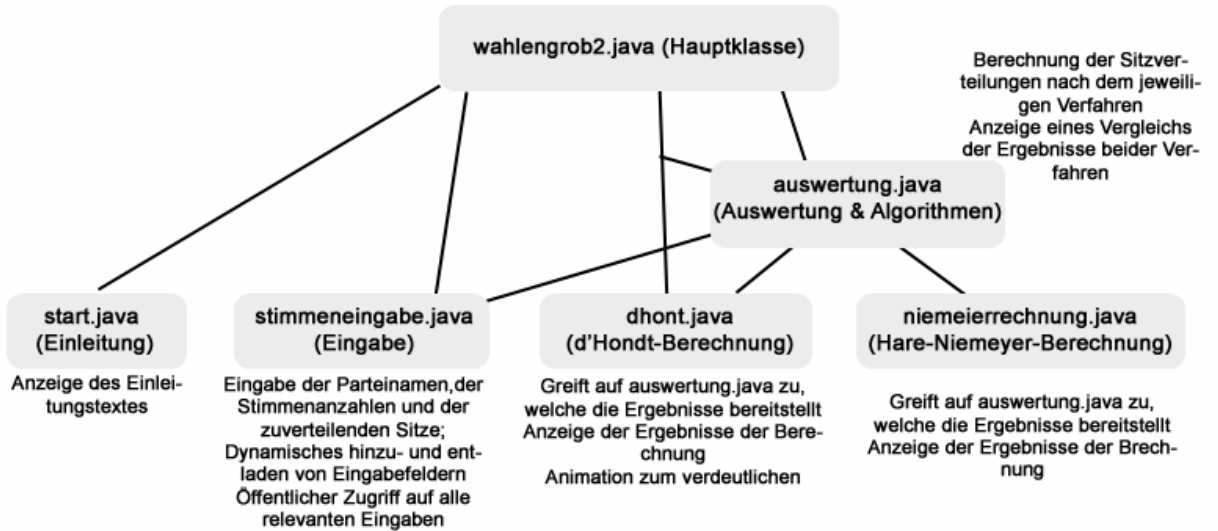
Eine Übersicht über die Programmarchitektur wird mit Hilfe des auf der nächsten Seite folgenden Diagramms geboten. Neben den einzelnen Klassen (welche grau unterlegt wurden) sind die wichtigsten Funktionen der Klasse beschrieben.

Die Verbindungsstriche stellen Schnittstellen/Zugriffe zwischen den einzelnen Klassen dar. So ist z.B. ersichtlich, dass auf auswertung.java von dhont niemeyerrechnung und wahlengrob2 zugegriffen wird. Auswertung.java greift selbst auf stimmeingabe.java zu.

Programmarchitektur

Die Verbindungsstriche stellen wichtige Verbindungen bzw Schnittstellen zwischen Klassen dar.

Grundgerüst
Anzeige der kompletten Navigations
Hinzuladen und entladen der einzelnen Klassen
Hilft bei der Übergabe von Objekte



4.2 Funktionskomplex/Klasse 1: wahlengrob2.java / start.java



Wahlengrob2.java ist auf jedem Bild vorhanden. Da dies die Hauptklasse ist. Start.java stellt weder Methoden noch Zustände zu Verfügung. Sie stellt nur den Text dar.

4.2.1 Zustände

Leider ist uns nicht ersichtlich, was mit „Zustände“ gemeint wird. Eine Anfrage an den Support blieb unbeantwortet.

4.2.2 Methoden

Wahlengrob.java stellt folgende Methoden zur Verfügung:

- void SchrittRefresh():aktualisiert die Navigation
- void callframe():sorgt dafür, dass die richtigen Seiten bzw. Klassen geladen und angezeigt werden.

4.3 Funktionskomplex/Klasse 2: stimmeneingabe.java

Partei	Stimmen	Sonstiges
Partei 1	<input type="text" value="10000"/>	Zu verteilende Sitze <input type="text" value="15"/>
Partei 2	<input type="text" value="10010"/>	
Partei 3	<input type="text" value="10020"/>	
Partei 4	<input type="text" value="10030"/>	
Partei 5	<input type="text" value="10040"/>	

Applet gestartet

Ermöglicht die Eingabe der Parteien, der Stimmenanzahlen und der zu verteilenden Sitze. Außerdem ist es möglich dynamisch, während die Klasse angezeigt wird, neue Eingabefelder hinzuzufügen. Die Werte der Eingabefelder werden dann fast allen anderen Klassen zu Verfügung gestellt.

4.3.1 Zustände

Leider ist uns nicht ersichtlich, was mit „Zustände“ gemeint wird. Eine Anfrage an den Support blieb unbeantwortet.

4.3.2 Methoden

- void parteiwegActionPerformed(): entfernt dynamisch das letzte Partei- und Stimmeneingabefeld
- void parteihinzuActionPerformed(): fügt dynamisch ein neues Partei- und Stimmeneingabefeld unten hinzu.

4.4 Funktionskomplex/Klasse 3:dhontrechnung.java

Applet-Ansicht: wahlengrob2.class

Applet

Einleitung >> Eingabe >> **d'Hondt-Berechnung** >> Hare-Niemeyer-Berechnung >> Auswertung

D'Hondt-Verfahren

Bei Verwendung des d'hondtschen Höchstzahlverfahrens teilt man die Zahl der erhaltenen Stimmen einer Partei nacheinander durch eine aufsteigende Reihe natürlicher Zahlen (1, 2, 3, 4, 5, ... n). Die dabei erhaltenen Bruchzahlen werden als Höchstzahlen bezeichnet. Die Höchstzahlen werden absteigend nach ihrer Größe geordnet. Die so ermittelte Reihenfolge gibt die Vergabereihenfolge der Sitze an. Die Vergabereihenfolge wird anhand der **roten Zahlen** in den Klammern nach und nach angezeigt.

Divisor (n)	Partei 1	Partei 2	Partei 3
1	5535.0 (5)	3423.0 (8)	23434.0 (1)
2	2767.5	1711.5	11717.0 (2)
3	1845.0	1141.0	7811.33 (3)
4	1383.75	855.75	5858.5 (4)
5	1107.0	684.6	4686.8 (6)
6	922.5	570.5	3905.67 (7)
7	790.71	489.0	3347.71
8	691.88	427.88	2929.25
9	615.0	380.33	2603.78

Und so weiter...

Weiter zur Hare-Niemeyer-Berechnung

Applet gestartet

Anzeige des d'Hondt-Verfahrens inkl. Erklärung. Es werden 9 Reihen der Höchstzahlen als Beispiel generiert und in einer Tabelle angefügt. Zudem wird mit einer Animation die Vergabereihenfolge deutlich gemacht.

4.4.1 Zustände

Leider ist uns nicht ersichtlich, was mit „Zustände“ gemeint wird. Eine Anfrage an den Support blieb unbeantwortet.

4.4.2 Methoden

- void animationstart(): Startet die Animation, welche die Vergabereihenfolge verdeutlicht. Die Animation selbst wird in einer Timer Subklasse gemacht.

4.5 Funktionskomplex/Klasse 4: niemeierrechnung.java

Applet-Ansicht: wahlengrob2.class

Applet

Einleitung >> Eingabe >> d'Hondt-Berechnung >> **Hare-Niemeyer-Berechnung** >> Auswertung

Hare-Niemeyer-Berechnung

1. Schritt: Grundverteilung

Die Stimmen der Parteien werden durch die Gesamtstimmenzahl aller Parteien (ohne ungültige Stimmen und Enthaltungen) dividiert und mit der Gesamtsitzzahl multipliziert (=Quote). Der abgerundete Teil der Quote wird als Sitzzahl direkt zugeteilt.

2. Schritt: Restsitzverteilung

Die Restsitze werden in der Reihenfolge der größten Nachkommateile der Quoten den Parteien zugeteilt.

Partei	Stimmen	Quote	Sitze
Partei 1	5535	2,56313	2
Partei 2	3423	1,58511	1 + 1
Partei 3	23434	10,85175	10 + 1

[Weiter zur Auswertung](#)

Applet gestartet

Anschauliche Anzeige der Sitzverteilung nach dem Hare-Niemeyer-Verfahren inkl. Erklärung.

4.5.1 Zustände

Leider ist uns nicht ersichtlich, was mit „Zustände“ gemeint wird. Eine Anfrage an den Support blieb unbeantwortet.

4.5.2 Methoden

Diese Klasse stellt keinerlei Methoden zur Verfügung außer den Konstruktor.

4.6 Funktionskomplex/Klasse 4: auswertung.java

Applet-Ansicht: wahlengrob2.class

Applet

Einleitung >> Eingabe >> d'Hondt-Berechnung >> Hare-Niemeyer-Berechnung >> **Auswertung**

Auswertung & Vergleich

Hier kannst du die Ergebnisse des Sitzverteilung beider Verfahren im Vergleich sehen. Unterschiede sind **rot markiert**.

	Sitze nach d'Hondt	verglichen mit	Sitze nach Hare-Niemeyer
Partei 1	2	ebenso viel	2
Partei 2	1	weniger als	2
Partei 3	12	mehr als	11

Zurück zur Einleitung

Applet gestartet

Diese Klasse beinhaltet die Algorithmen beider Wahlverfahren. Außerdem erstellt sie einen übersichtlichen Vergleich beider Verfahren.

4.6.1 Zustände

Leider ist uns nicht ersichtlich, was mit „Zustände“ gemeint wird. Eine Anfrage an den Support blieb unbeantwortet.

4.6.2 Methoden

- `int[] hareniemeyer(stimmeneingabe)`: Berechnet die Sitzverteilung nach dem Hare-Niemeyer-Verfahren. Die Sitze der Parteien werden in einem `int`-Array ausgegeben.
- `int[] dhondt(stimmeneingabe pstimmeneingabe)`: Berechnet die Sitzverteilung nach dem d'Hondt-Verfahren. Die Sitze der Parteien werden in einem `int`-Array ausgegeben.
- `double runden(double d, int n)`: einen Wert `d` auf `n` Nachkommastellen runden

5 *Benutzerschnittstelle*

5.1 *Konzept*

Das Programm wurde von uns aus Gründen der Übersichtlichkeit in fünf Seiten gegliedert. Jede Eingabe-Seite stellt klare Forderungen an den Benutzer, die es zu erfüllen gilt. So muss z. B. die Anzahl der Parteien und deren Wählerstimmen eingegeben werden. Wir haben uns bemüht, die Benutzeroberfläche so benutzerfreundlich und das Programm so lehrreich (mit Animation etc.) wie möglich zu gestalten.

5.2 *Anzahl der Parteien*

In diesem ersten Schritt wird der Anwender dazu aufgefordert, die Anzahl der Parteien einzugeben.

5.3 *Parteiendefinition*

Im zweiten Schritt geht es darum, zu definieren welche Parteien an der Wahl teilgenommen haben und wie viele Stimmen die einzelnen Parteien erhalten haben. Des Weiteren lassen sich noch die Anzahl der zu vergebenen Sitze und Sperr-Klauseln in die Rechnung mit einzubeziehen.

5.4 *Ausgabe nach d'Hondt*

Im dritten Schritt bekommt der Benutzer die Ergebnisse nach d'Hondt präsentiert. Zusätzlich wird in einem kurzen Text das Rechenverfahren erläutert und eine Animation abgespielt.

5.5 *Ausgabe nach Hare/Niemeyer*

Im vierten Schritt werden die Ergebnisse nach dem Hare/Niemeyer-Verfahren aufgezeigt. Auch hier wird das Verfahren anhand eines kurzen Textes und einer Tabelle erklärt und dem Benutzer so nahe gebracht.

5.6 *Auswertung*

Der letzte Schritt besteht aus einer übersichtlichen Auswertung beider Verfahren. Dem Benutzer wird hier klar gemacht, wie die Auswahl eines bestimmten Verfahrens das Endergebnis beeinflussen kann.

6 *Referenzen*

[1] freie Enzyklopädie Wikipedia (de.wikipedia.org)

7 Anlagen

7.1 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.>

7.2 Eidesstattliche Erklärung

<separater Ausdruck einreichen>

7.3 Einverständniserklärung der Erziehungsberechtigten

<separater Ausdruck einreichen>