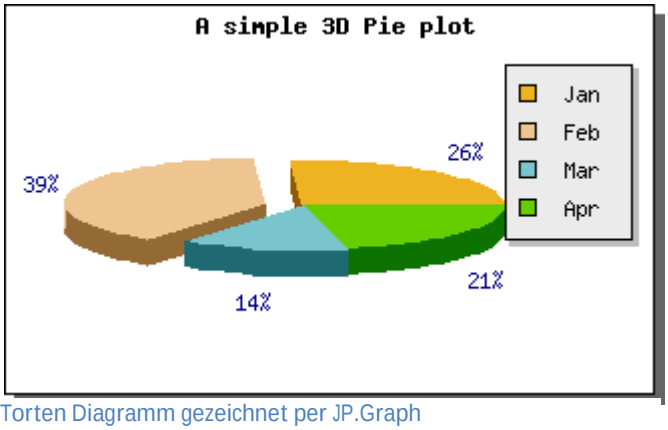
FRIEDENSTRASSE
WILHELMSHAVEN

2009

Grafische Darstellungsmöglichkeiten von Datenbankinhalten



Torten Diagramm gezeichnet per JP.Graph


Google Visualization API

Nikolas Kittler – Timm Schütte
BBS Friedenstrasse
07.05.2009

Schule:	BBS Friedenstrasse Wilhelmshaven
Kurs:	Datenbanken
Fach:	Informationstechnik
Ausgabetermin des Themas:	06.03.2009
Abgabetermin des Themas:	08.05.2009

. . . . . .

Unterschrift Schülerin / Schüler

. . . . . .

Unterschrift Fachlehrer / Fachlehrerin

Die vorliegende Arbeit wurde am . . . . . eingereicht.

Note: . . . . . KMK-Punkte: . . . . . .

Erklärung

Hiermit erkläre ich, dass ich damit einverstanden bin, wenn die von mir verfasste Facharbeit der schulinternen Öffentlichkeit zugänglich gemacht wird.

. . . . . .

Ort, Datum

. . . . . .

Unterschrift

Nikolas Kittler – Timm Schütte

Grafische Darstellung von Datenbankinhalten

Inhaltsverzeichnis

1 Einleitung

„Welche Rolle spielt eigentlich visuelle Darstellung von Datenbankinhalten heute?“

Im heutigen Informationszeitalter sind Datenbanken gar nicht mehr weg zudenken, ganz egal ob der einfache Kioskbesitzer seine Ein- und Verkäufe via Datenbank managet oder ob an der Börse aktuelle Werte von Aktien durchlaufen und schon im Sekundentakt ausgewertet und als Diagramme dargestellt werden. **Zeitlich variable Werte sollen dabei möglichst anschaulich in einem Diagramm ausgegeben werden um den Betrachter einen aktuellen Überblick über die Lage zu geben.** Datenbanken beziehungsweise Tabellen stellen dabei die Basis und Speicherplatz dieser Informationen dar, dabei werden diese Werte ausgelesen und in Diagramme umgewandelt.

1.1 Inhaltsübersicht

Inhaltliche Trennung in theoretische und praktische Arbeit.

Innerhalb unserer Facharbeit soll es dabei klar werden wie Datenbankinhalte zu einen darauf genierten Diagramm führen, so dass es für einen HTML Programmier und jenen schlichten Benutzer der eine Internetseite im World Wide Web auf die Schnelle programmieren möchte klar ist wie er Werte in einer Datenbank, die er vielleicht anlegen möchte, visuell darstellen kann. Wir haben uns daher im praktischen Teil der Arbeit lediglich auf eine Darstellungsmöglichkeit begrenzt via die sogenannte Programmbibliothek „JPGraph“ begrenzt. Andererseits gibt es weitaus mehr Darstellungsmöglichkeiten über andere Programmier- und Schriftsprachen die Diagramme und auf variablen Werten realisiert werden können. Auf die wir im theoretischen Teil näher eingegangen sind jedoch haben wir uns nur dort nur auf **Google Visualization API** festgelegt und angeschnitten welches Schema der Programmierer dort anwenden muss um damit ein Diagramm erstellen zu können.

Dabei wird generell eine Ähnlichkeit von dem Verlauf der Programmiersprachen beziehungsweise der Programmbibliotheken aufgewiesen: variable Inhalte der Datenbank werden via Script „genommen“ und eine Grafik basiert auf den Werten wird generiert, der Benutzer muss aber dabei klar die Grenzen dieses Vorgangs vorher abgrenzen können damit keine Fehler entstehen können.#

Im Inhalt der Facharbeit haben wir stark getrennt zwischen praktischer Arbeit, Anhand eines sachlichen Beispiels unter JPGraph und unter theoretischer Arbeit, die tiefer auf Hintergrundinformationen zu verschiedenen Systemen der Darstellung von Datenbankinhalten eingeht.

1.2 Problemstellung

„Was war eigentlich die Aufgabenstellung?“

Laut der Aufgabenstellung sollten wir uns eher auf praktischer Basis mit der visuellen Darstellung von Datenbankinhalten befassen, dabei konnten wir anhand einer fertigen Bibliothek namens „JPGraph“ die Ergebnisse einer Wahl anhand eines Säulen und 3D-Torten Diagrammes darstellen. Je nach des Umfangs der verwendeten Bibliothek sollten Diagramm Beispiele ausgewählt und aufgezeigt werden wie man diese programmieren kann.

1.3 Abgrenzung des Themas

„Praktisch und theoretische Arbeit?, „Nochmal wie weit werden welche Themen aufgegriffen?“

In unserem Fall haben wir uns im praktischen Teil der Facharbeit auf die Bibliothek „JPGraph“ beschränkt und uns klar dabei auf ein sachliches und sogar politisch aktuelles Beispiel im Wahljahr 2009 bezogen. Dabei haben wir eine Internetseite programmiert auf der per Mausklick möglich ist eine zufällige Anzahl von Wähler per Klick zur *virtuellen Wahlurne* zu bitten um dabei zufällig seine Stimme auf eine der großen Parteien abzugeben. Basis dieser Zufälligen generierten Automatik ist dabei der sogenannte „Randombefehl“ der dabei lediglich Anzahl der Wähler, Wahlteilnahme und die von „ihm“ abgegebene Stimme auf eine der großen Parteien zufällig bezieht. Die Werte werden daraufhin per Bibliothek „JPGraph“ ausgelesen und dann in ein Balkendiagramm für die Wahlteilnahme und ein Tortendiagramm für das Wahlresultat umgewandelt. Dieses Beispiel macht es anschaulich wie die der *PHP-Laie* seine Tabellen die er visualisieren kann hat und wie die Verwendung von „JPGraph“ abläuft.

Im theoretischen Teil der Arbeit sind wir etwas breit gefächerter an das Thema, der Darstellungsmöglichkeiten von Datenbankinhalten eingegangen. Wir haben verschiedene Möglichkeiten der Darstellung näher dargelegt und PHP- Bibliotheken näher beschrieben, und sind auch auf die Aspekte eingegangen warum es sich besonders heute eignet als Programmier oder auch Laie PHP- Bibliotheken zu verwenden. Dabei lässt sich von vorn herein feststellen das es heutzutage ein beliebter Dreh und Angelpunkt ganzer Branchen ist, Datenbankinhalte oder Datenbanksysteme zu vermarkten. Wobei sich dabei ein Anbieter von Datenbanksystemen auch gleich seinen Script der visuellen Darstellung seiner

Datenbankensysteme anpasst und sie als Co-Produkt seines eigentlichen Datenbankmanagementsystems anbietet.

1.4 Nennung und Begründung der gewählten Arbeitsweisen und Methoden

Aufbau eines praktischen Problems bis hin zur Quellcode Kommentierung.

Im praktischen Teil der Arbeit wurde ein „praktisches“ Problem erschaffen um anhand dessen Techniken der PHP- Bibliothek JPGraph aufzuweisen und durch eine ausführliche Quellcode Kommentierung erkenntlich zumachen. Durch diese Methode ist der Leser dieser Facharbeit auch in Zukunft in der Lage Datenbankinhalte visuell darstellen zu können.

Im theoretischen Teil der Facharbeit sind wir auf andere neuere Felder der Erschließung Datenbank Visualisierungen eingegangen, und haben bei einer anderen Bibliothek namens **Google Visualization API** kurz „angeschnitten“ wie es funktioniert und durch welches Schema der Anwender gehen muss um bei durch diese Bibliothek Diagramme von seinen Datenbankeinträgen zu erstellen. Vor- und Nachteile wurden einzeln zu den Bibliotheken angeführt und zeigen besonders gut Probleme von alten beziehungsweise neueren Bibliotheken auf. Vor und Nachteile beziehen sich dabei darauf wie gut die Handhabung ist, ob viel oder wenig Vorkenntnisse erfordert werden oder ob es noch weitere besonderes gute Punkte den Anwender bestechen kann.

2 Darstellungsmöglichkeiten von Datenbankinhalten

“PHP- Bibliotheken”, “Java” und “Flash” und viele mehr.

2.1 PHP- Bibliotheken

Der einfache Weg ursprünglich komplexere Befehle beliebig oft und vereinfacht anzuwenden.

PHP- Bibliotheken bieten viele Vorteile für den Autonormalprogrammier von Internetseiten, ohne große Vorkenntnisse kann er schon vorher programmierte Klassen und Befehle von einer Drittperson übernehmen. Wichtig dabei zu wissen ist das eine Bibliothek in diesem Fall „JPGraph“ verschiedene Klassen und Befehle enthalten kann. Der Programmier kann dadurch direkt Befehle und vorhandene Techniken benutzen. Dies Prinzip ist besonders vorteilhaft und erspart den Programmier ein lästiges erstellen von Techniken da vorhandene Befehle beliebig oft übertragbar auf verschiedene Problematiken sein kann. Dieses zu nutzen nehmen von Bibliotheken ist besonders gängig in der Objektorientieren Programmiersprache, die mittlerweile eine große Rolle in der heutigen Zeit.

2.2 Vor und Nachteile von PHP- Bibliotheken

Wodurch bestechen die PHP- Bibliotheken in der Anwendung auf Visualisierungen und wodurch nicht.

Vorteile	Nachteile
• Dies ist besonders positiv für den Laien und Hobby Programmierer ohne viele Vorkenntnisse. • Sind belieb oft anwendbar ohne große Probleme. • Im Internet lassen sich viele verschiedene Einführungen für diese Bibliotheken finden im Fachjargon nennt man diese **Tutorials.** • Fragen können mit einer Community [1] im Forum geklärt werden. • Die meisten Bibliotheken enthalten ein umfassendes Specktrum an verschiedensten Diagrammformen. • Manche Bibliotheken sind erweiterbar um Klassen und Diagrammarten.	• Eben dadurch, dass komplexe Befehle vereinfacht werden nimmt der Anwender dieser nicht mehr Bezug zum eigentlichen Ablauf und verliert diesen aus den Augen. • Für fortgeschrittene Anwender sind die Bibliotheken nur schlecht erweiterbar, um z.B. seine eigene Klassen und Objekte anzuhängen. • Manche sind mit Kosten verbunden. • Teilweise werden viele Vorkenntnisse vom Anwender abverlangt. • Die meisten Tutorials liegen in Englisch vor somit wird ein guter Umgang mit der englischen Sprache vorausgesetzt

2.3 Fazit zu PHP- Bibliotheken

„Wer sollte PHP- Bibliotheken verwenden und wer nicht?“

Durch diese Gegenüberstellung lässt sich schließen das PHP- Bibliotheken die für die Visualisierung von Datenbankinhalten verwendet werden sich besonders für Laien anbietet, da vieles vereinfacht dargestellt wird. Viele Diagrammformen sind bereits enthalten und Einführungen in diese

Bibliotheken sind im Internet zu finden, jedoch lässt sich ganz ohne Fachkenntnisse ein Diagramm durch PHP nicht erstellen. Alles in allem sind Bibliotheken für den Laien brauchbar der keinen Bezug zur dahinter stehenden, viel tiefer gehenden Programmierung nehmen möchte, jedoch sind sie nicht für jene Art von Programmieren zu empfehlen die schon eine große Erfahrung in der Programmierung von Diagrammen haben und diese aus dem „FF“ programmieren können. Da macht es wenig Sinn die Bibliotheken nur ansatzweise anzurühren.

[1] Community, eine offene Gemeinschaft an Leuten die über verschiedenste Arten wie Foren, Chats oder auch Blogs über Problematiken, Fragen von Mitgliedern der Gemeinschaft fachsimpelt, sich austauscht oder einfach nur berät

2.4 Java und Flashdarstellungen von Datenbankinhalten

„Neue Alternativen im Internet?“

Java und Flash spielen im Internet zunehmend eine große Rolle ganz egal ob bei dem neuartigen IPTv[2] oder bei anderen kleineren Flashspielchen oder sogar bei einer dynamischen Werbeanzeige auf großen Internetplattformen. Flash und Java setzen dabei eine große Fachkenntnis an den Programmierer voraus und erfordern viel Zeit und Arbeit um einfache Grafiken oder gar Statistiken anzuzeigen. Nachteilig ist auch das bei einer vorhanden Java bzw. Flash Grafik die man im Internet anderen Menschen zugänglich machen möchte ein sogenannten Plug-In für den auf Betrachterseiten vorhanden Browser vorhanden sein muss, um die Grafik auch für ihn anschaubar zumachen.

Die Thematiken und Strukturen wie Java oder auch Flash auf eine Datenbank zugreift haben wir nicht detailliert in dieser Arbeit nicht angeführt da wir uns im praktischen Teil näher mit der Bibliothek „JPGraph“ beschäftigt haben. Jedoch haben wir noch zwei weitere Bibliotheken angeführt wobei **Google Visualization API** nur unter Anwendung von der Scriptsprache JavaScript die über 3 Ecken auf eine Datenbank zugreifen kann, dadurch werden viele Vorkenntnisse benötigt in unter anderen 3 Programmiersprachen wie JavaScript, PHP und JSON.

3 Google Visualization API

Die neue Möglichkeit von Google Datenbanken auszuwerten

3.1 Hintergrundinformationen zu Google

„Woher kommt eigentlich Google?“

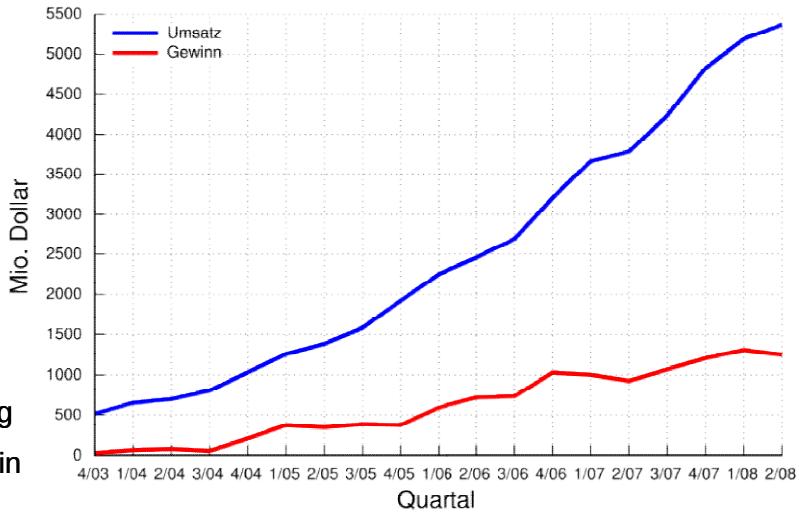
Im Internet und im Alltag jedes Einzelnen ist heute dieses Schlagwort **Google** nicht mehr wegzudenken. Sogar im Duden wurde schon 2003 das Wort googlen mit aufgenommen. Dabei war die Idee wenn man genauer darüber nachdenkt so leicht mit der im 7. September 1998 von *Larry Page* und *Sergei Brin* ein Suchmaschinenunternehmen gestartet wurde:

„Das Ziel von Google besteht darin, die Informationen der Welt zu organisieren und allgemein nutzbar und zugänglich zu machen.“[3]

[2] IPTv ist eine Art über einen Flashplayer sich im Internet von anderen Usern aufgenommen Serien oder gar Filme direkt im Browserfenster anzuschauen.
[3] Zitat, von wikipedia.org unter dem Artikel google.inc

– Google Unternehmensinformationen

Im Verlauf der Zeit nahmen die Marktanteile und Gewinne von Google immer mehr zu und Google erschloss sich mit den hohen Gewinnen immer mehr Markt- bzw. Dienstfelder. Eine der größten Quellen der Google Gewinne fliest über die Werbegelder die Firmen an Google bezahlen um beispielsweise einen unterlegten Eintrag oder ganz weit obenstehenden Eintrag in der Suche bekommen wenn der Benutzer irgendwas ähnliches das der Firma nahe kommt bei Google sucht oder anders gesagt googelt.



Umsatz und Gewinnentwicklung von Google *Quelle: Wikipedia.org*

Heute gibt es neben der ursprünglichen Suchmaschine viele weitere Dienste: Google.mail – ein Email Dienst der den Benutzer umsonst bis zu 2 Gigabyte Speicherplatz bietet für seine E-Mails, Google.maps der kommerziell einen Routenplaner bietet ohne dabei Kosten vom Benutzer zu fordern, Google.video – Eine Video Plattform angelehnt an das große Vorbild YouTube jedoch mit nur mäßigen Erfolg, und Google Chrome einen Internetbrowser der jedoch sich nur schwer im schon beherrschten Markt von den beiden Explorertypen „Firefox“ und „Internetexplorer“ vordringen kann. Es gibt noch viele weitere Dienste von Google jedoch möchten wir nun an dieser Stelle weiter auf den Dienst Google Visualization API eingehen.

3.2 Google Visualization API in der Anwendung

„Wie gestaltet sich die Handhabung von Google Visualization API?“

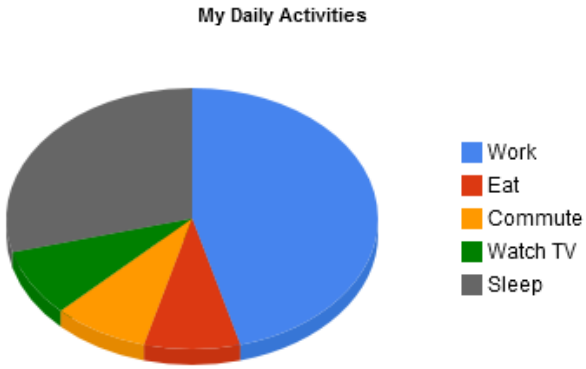
Bei Google Visualization API hat sich Google.inc an der Programmiersprache JavaScript bedient. Google hat jedoch auch wie bei der Bibliothek JPGraph alle benötigten Dateien in einem Paket zusammen geschnürt um den Programmierer möglichst eine unproblematischen Start mit Google Visualization API anzubieten. Jedoch werden viele Teile der Bibliothek nur online angeboten d.h. man muss sie direkt in der PHP- Datei als online Verknüpfung einbinden somit wird vorausgesetzt dass eine Internetverbindung vorhanden ist.

Hauptbestandteil von Google API Visualization ist eine enggeknöpftes Schema mit der man ein Diagramm basierend auf Datensätze die vorher schon angelegt wurden oder noch innerhalb bei der Erstellung des Diagramms eingefügt werden können, anlegen kann:

1. Zu aller erst muss der Laie oder auch Programmierer im Quellcode der PHP- Datei in der sich das Diagramm befinden soll muss folgender Quellcode Implementiert werden damit **Google Visualization API** problemlos geladen werden kann:

   ```
   <!--Load the AJAX API-->
   <script type="text/javascript" src="http://www.google.com/jsapi"></script>
   ```

2. In diesem Schritt werden nun die benötigten Pakete zusätzlich dazu geladen, außerdem können von Benutzern die im Internet Pakete für Google Visualization API geschrieben haben zusätzlich hinzugefügt werden.
3. In diesem Schritt geht es an das „Eingemachte“. Die Visualisierung der Grafik soll in 4 Unterschritten eingeleitet werden d.h.:
   a. Zuerst muss man seine Datensätze bzw. Datenbankinhalte dafür vorbereiten und genau bestimmen was man anzeigen möchte das. Kann entweder durch den klassischen query() Befehl zu einer anderen PHP- Seite geschehen oder die Daten können bei der Erstellung der Grafik hinzugefügt werden.
   b. Unter diesem Unterschritt wird definiert welche Diagrammart benutzt werden soll
      und ob es sich um eine HTML, GIF oder Sogar mov. Datei handeln soll, da man durch JavaScript viele weitere Darstellungsmöglichkeiten in die Grafik einbinden kann wie z.B. das wenn man über ein Stück eines Tortendiagrammes mit dem Mauscursor geht das die korrekte %-Zahl eingeblendet wird sowie noch einmal zusätzlich die Legende und Erklärung der Grafik.



eine solche Grafik lässt durch Google Visualization API erstellen.

   c. Unter diesem Unterschritt kann noch zusätzlich entschieden werden ob der erstellen der Grafik noch *handlingEvents* oder *ReadyEvents* hinzufügen möchte. Der Unterschied ist dabei simpel, *handlingEvents* werden durch den Betrachter der Grafik ausgelöst z.B. wenn er den Cursor über einen bestimmten Teil der Grafik führt.
      Auf der anderen Seite gibt es die *ReadyEvents* die mit dem Aufruf der Grafik ausgeführt werden können.
   d. Der finale Schritt erzeugt das Diagramm bzw. die Grafik. Durch den vielleicht schon aus Java bekannten Draw() Befehl, aber dafür wird zuerst eine *DataTable* benötigt, d.h. das Array welches aus den Datenbankinhalten die

angezeigt werden sollen müssen hier eingebunden werden, zusätzlich können hier Parameter dazu übergeben werden.

3.3 Vor- und Nachteile von Google Visualization API

„Läuft Google mit dem Trend?“

Vorteile	Nachteile
• Eine eindeutig große Anzahl an Skripten für Grafiken wird angeboten. • Ein gutes und ausführliches Tutorial wird auf einer Internetseite angeführt. • Die Bibliothek ist durch Anwender erweiterbar dank einer *opensource* Technik. • Fehler können direkt an Google übermittelt werden. • In Foren können sich Anwender dieses Systems austauschen über Probleme oder Neuerungen. • Visuelle Effekte werden durch *handlingEvents* sowie *ReadyEvents* dank JavaScript gut ausgenutzt.	• Ein großes Fachwissen ist von Nöten, in Sprachen wie JavaScript, PHP, JSON sowie HTML. • Englisches und nicht deutsches Tutorial ist vorhanden. • Eine Internetverbindung wird für manche Bestandteile der Bibliothek vorausgesetzt. • Das *„Java Plug-In“* wird zum anzeigen der Grafiken benötigt.

3.4 Fazit zu Google Visualization API

„mehr etwas für den erfahrenen Anwender“

Bei dieser PHP- Bibliothek die von Google.inc veröffentlicht wurde werden große Ansprüche an den Anwender gestellt, in unter 4 „Sprachen“ muss er ein gutes Wissen aufweisen. Allerdings wenn er das nötige Wissen sich angeeignet hat kann man damit schon beeindruckende Diagramme und Grafiken erstellen. Diese Bibliothek ist daher nur besonders für erfahrene Programmier zu empfehlen die beispielsweise in Onlineredaktionen Statistiken zu aktuellen Nachrichten besonders unterlegen möchten. Zum Beispiel bei einer Neuen Arbeitslosen Quote oder den Wahlergebnissen.

4 „JPGraph”

Hauptbestandteil der Facharbeit – Instandsetzung und Visualisierungsmöglichkeiten

Um ein direktes Beispiel bieten zu können, um Datenbanken visualisieren zu können, beschränken wir uns auf die Klassenbibliothek [4] JPGraph. Diese Bibliothek ist im Internet öffentlich zugänglich, von daher kommen für die Anschaffung keine Kosten auf. Mit JPGraph ist es möglich, mithilfe von einfachem Programmiercode PHP-Daten und/oder Datenbankinhalte auf diverse Grafiken abzubilden und diese zu bearbeiten. Dieser Bearbeitungsvorgang obliegt dem Host, das heißt, der User kann die Grafiken auch ohne überhaupt die Bibliothek auf seinem Rechner zu haben auf dem Browser darstellen lassen. JPGraph benötigt PHP-Programmiercode und zu funktionieren. Die Informationen der Datenbanken werden lediglich mit PHP ausgelesen, und nicht direkt von Datenbank auf Grafik übertragen. Der Vorteil gegenüber anderen Bibliotheken wäre, dass es bereits komplexe Visualisierungsmöglichkeiten gibt, und die Grafiken, wie Linien-, Balken-, 2D und 3D-Tortendiagramme, sich fast beliebig manipulieren und bearbeiten lassen. Außerdem lässt sie sich in vielen Versionen (mitunter auch den alten) von PHP kompilieren, denn JPGraph passt sich an die aktuell benutzte Version an. Was sich allerdings als nachteilhaft erweist wäre die Tatsache, dass diese Bibliothek noch auf älteren Datensätzen basiert. So ist die Programmierung nicht mehr so einfach und variabel wie andere Bibliotheken. Updates werden zwar immer wieder zur Verfügung gestellt, jedoch in unregelmäßigen Abständen, und auch nur um die Basisstruktur etwas zu verbessern. Ebenfalls sind die verfügbaren Grafiken sehr schlicht gehalten, was bedeutet, dass diese Grafiken ungeeignet sind für Homepages mit aufwendigerem Design. Die einzelnen Grafiken zu ersetzen erfordern einen direkten Eingriff auf die Bibliotheken, welche sehr schwer zu überschauen sind. Alleine die Hauptdatei beträgt fast 6000 Zeilen Quellcode, welche auch noch mehrfach verschachtelt ist. Für simple Demonstrationszwecke, einfache Aufgaben und Problemlösungen ist JPGraph dennoch sehr gut geeignet, welches auch Ausschlag gebend war für unsere Wahl.

4.1 Vorbereitung: JPGraph instandsetzen

Bevor überhaupt Grafiken entstehen können, muss JPGraph zunächst vorbereitet werden. Vorausgesetzt wird mindestens für die aktuelle Version:

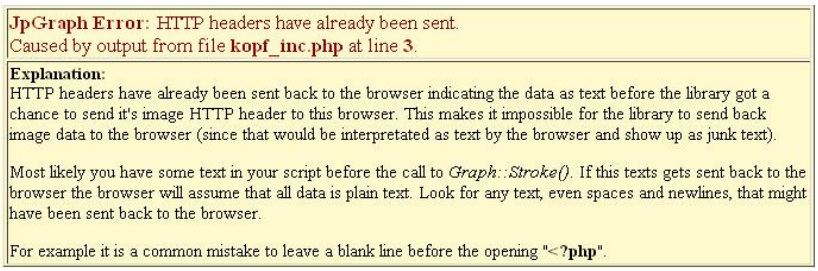
- Apache Server (andere Server sind auch möglich, jedoch nehmen wir dies als Beispiel)
- PHP mit der Version 4.1 oder höher (ältere Versionen sind mit älteren JPGraph-Bibliotheken sind ebenfalls möglich)
- JPGraph Library

Wir gehen jetzt davon aus, dass der Apache Server und PHP bereits installiert wurden. Nun sollte zunächst ein Verzeichnis für die Homepage erstellt werden. In unserem Falle nehmen

[4] Klassenbibliothek, bedeutet für einen Programmierer das mehrere Klassen, Objekte oder gar einzelne Befehle in einem Paket zusammen gefasst werden und auf belieb viele Problematiken angewandt werden können.

wir *C:\xampp\htdocs\jpgraph*. In diesen Ordner wird nun die JPGraph Library entpackt. Benötigt wird allerdings nur der *src*-Ordner. Nun könnte man schon loslegen.

Zunächst muss man wissen, dass die ausgegebene Grafik selbst eine php-Datei ist und nicht wie vielleicht angenommen eine .jpg-Datei oder dergleichen. Allerdings wird diese PHP-Datei wie eine Grafik behandelt. Das bedeutet, wenn man das erstellte Diagramm anzeigen lassen will, nicht durch einen *include*-Befehl aufrufen kann, sondern durch den HTML Befehl <img src="diagrammname.php">.
Wenn man den include Befehl verwendet, wird eine Fehlermeldung ausgegeben.


JpGraph Error: HTTP headers have already been sent.
Caused by output from file kopf_inc.php at line 3.

Explanation:
HTTP headers have already been sent back to the browser indicating the data as text before the library got a chance to send it's image HTTP header to this browser. This makes it impossible for the library to send back image data to the browser (since that would be interpretated as text by the browser and show up as junk text).

Most likely you have some text in your script before the call to *Graph::Stroke()*. If this texts gets sent back to the browser the browser will assume that all data is plain text. Look for any text, even spaces and newlines, that might have been sent back to the browser.

For example it is a common mistake to leave a blank line before the opening "<?php".

Der Grund dafür ist, dass JPGraph eine vollständige Seite kompiliert, und dafür HTML Header verwendet, und gleichzeitig auch schließt. Verwendet man nun den include-Befehl auf der Seite, auf welcher die Grafik angezeigt werden soll, kann man keinen weiteren Header mehr benutzen. Es kann nur ein Header pro Seite verwendet werden, von daher kann man entweder die Grafik allein anzeigen lassen oder den Rest der Seite mit dieser Fehlermeldung.
Empfehlungswert wäre also, dass man für jede Grafik eine eigenständige PHP-Datei anlegt, auch um der Übersicht willen.

4.2 Basisbefehle

Jede Darstellungsmöglichkeit ist nach dem gleichen System aufgebaut. Zwar weichen einige Befehle von Diagramm zu Diagramm ab oder sind gar komplett unterschiedlich, jedoch haben sie einige grundlegende Befehle gemein.

Zunächst muss in jedem PHP-Dokument die allgemeine Bibliothek aufgerufen werden. Diese Basisdatei heißt *jpgraph.php*. Aufzufinden ist diese Datei in dem vorherig entpackten Verzeichnis *src*. Da diese Datei in einem Unterverzeichnis unseres Projektordners befindet, wird sie wie folgt aufgerufen im PHP-Dokument:

```
<?php
include ("src/jpgraph.php");
?>
```

Es könnte auch dort das exakte Verzeichnis stehen, da der Host schließlich nur darauf zugreifen muss und nicht der User. Dennoch ist es empfehlenswert nur das Unterverzeichnis zu benennen. Gründe hierfür wären ein Verzeichniswechsel oder weitergeben der Dateien an einen anderen Server, der nicht den gleichen Dateipfad verwendet wie vorgegeben.

Nun müsste man sich für eine Diagrammart entscheiden. Denn jetzt muss die jeweilige Diagramm-Datei in die Datei eingebunden werden. In unserer Facharbeit haben wir uns auf 4 Darstellungsmöglichkeiten beschränkt, die wie folgt den Dateinamen haben:

jpgraph_line.php / für ein Liniendiagramm

jpgraph_bar.php / für ein Balkendiagramm

jpgraph_pie.php / für ein 2D-Tortendiagramm

jpgraph_pie3d.php / für ein 3D-Tortendiagramm

Hat man sich für ein Diagramm entschieden, bindet man sie einfach mit dem `include`–Befehl ein. Es gibt auch Verbindungsmöglichkeiten mit verschiedenen Diagrammen, doch dies ist von der Diagrammart abhängig. Ein Linien- und Balkendiagramm könnten zusammen dargestellt werden auf einer Grafik. Jedoch wäre es zum Beispiel nicht möglich ein Tortendiagramm mit einem Liniendiagramm zu verbinden, da es logischerweise keine Skala besitzt wie das Liniendiagramm. Nun muss zunächst eine Grafik gebildet werden. Dies wird mit folgendem Befehl ausgeführt:

```
$graph = new Graph([Breite],[Höhe],[Format]);
```

Die Textbausteine in den eckigen Klammern werden durch Werte oder Variablen ersetzt. Beim Format gibt es aber auch eine Möglichkeit, ein automatisches Format auszuwählen mit dem Attribut `"auto"`. Der Graph muss auch nicht zwingend `$graph` benannt werden. Man kann auch jeden beliebigen Namen auswählen. Jetzt muss ein Maßstab festgelegt werden.

```
$graph->SetScale("[x-Achse][y-Achse]");
```

Auch hier muss man die Textbausteine in den eckigen Klammern ersetzen, in dem Falle mit der Art der Skala. Man beachte, die beiden Skalen-Arten zusammengeschrieben werden. Folgende Skala-Arten sind möglich:

- lin – Lineare Skala
- log – Logarithmische Skala
- int – Integer Skala
- nur für x-Achse: text – Textskala

Nun kann man mit der Grafik-Formatierung beginnen. Die Formatierungsbefehle sind abhängig von der Diagramm-Art, von daher werden wir darauf später eingehen. Wenn die Formatierung abgeschlossen ist, kann der Graph mit `$graph->Stroke();` ausgegeben werden. All diese genannten Befehle sind in jedem Grafikcode vorhanden. Nun werden wir uns den einzelnen Diagrammen zuwenden. Bei den folgenden Beispielen zeigen wir zunächst Diagramme ohne Datenbank-Verknüpfung aus Platzgründen. Wie die Graphen dann mit Datenbankwerten gefüttert werden, zeigen wir später in unserem direkten Beispiel „Bundestagswahl-Generator“, wo wir auch das Balkendiagramm vorstellen. Denn die

Datenbankinhalte werden wie bereits in der Einleitung beschrieben mit PHP abgerufen, es erfolgt also keine direkte Kompilierung von Datenbank zu JPGraph.

4.3 Liniendiagramm

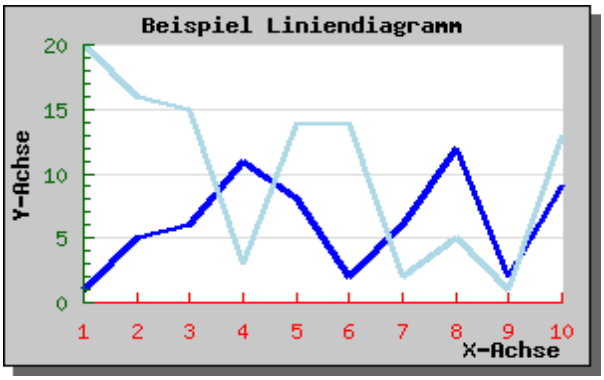
Damit eine bessere Übersicht geschaffen ist über die einzelnen Befehle für die Diagrammarten, verwenden wir zunächst den gesamten Quellcode und erläutern ihn im Detail.

```
include ("src/jpgraph.php");
include ("src/jpgraph_line.php");                    //1

$graph = new Graph(300,200,"auto");                  //2
$graph->SetScale("textlin");

$graph->title->Set("Beispiel Liniendiagramm");       //3
$graph->title->SetFont(FF_FONT1,FS_BOLD);

$graph->xaxis->title->Set("X-Achse");                //4
$graph->yaxis->title->Set("Y-Achse");
$graph->xaxis->title->SetFont(FF_FONT1,FS_BOLD);
$graph->yaxis->title->SetFont(FF_FONT1,FS_BOLD);
$graph->img->SetMargin(40,20,20,40);                 //5

$ydata = array(1,5,6,11,8,2,6,12,2,9);               //6
$ydata2 = array(20,16,15,3,14,14,2,5,1,13);
$lineplot=new LinePlot($ydata);
$lineplot2=new LinePlot($ydata2);

$lineplot->SetColor("blue");                         //7
$lineplot->SetWeight(2);
$lineplot2->SetColor("lightblue");
$lineplot2->SetWeight(2);

$graph->xaxis->SetColor("red");                      //8
$graph->yaxis->SetColor("darkgreen");
$graph->SetShadow();                                 //9
$graph->Add($lineplot);                              //10
$graph->Add($lineplot2);
$graph->Stroke();                                    //11
```

Diese Grafik sollte bei der Ausgabe im Browser erscheinen. Hier die Erläuterung des Quellcodes:

1) Einbinden der Liniendiagramm-Datei.
2) Festlegung der Graph-Skala und –Größe.
3) Benennung der Graphüberschrift und Festlegung von dessen Schrift.
4) X- und Y-Achsen Beschriftung einfügen mit Schriftformatierung .
5) Festlegung des Randes. (links, rechts, oben, unten)
6) Festlegung der Werte mittels festgelegten Arrays. Diese Arrays können durch generierte Arrays aus der Datenbank ersetzt werden. Mit dieser Methode werden die Datenbankinhalte dann ausgegeben. Wie genau sowas aussehen könnte wird dann in unserem Direktbeispiel behandelt.

7) Formatierung der einzelnen Linien. Zunächst wird jeweils die Farbe festgelegt und dann die Liniendicke. Die Farben können ausgewählt werden über Presets[5] oder über manuelle Eingabe eines Hexadezimalcodes.
8) Einfärben der einzelnen Achsen.
9) Schatten der Grafik einfügen. Dient rein der Gestaltung.
10) Einfügen der Linie auf den Graphen.
11) Ausgabe des Graphen



Es gibt noch etliche weitere Funktionen und Gestaltungsmöglichkeiten für das Liniendiagramm. Das Beispiel soll lediglich einen groben Überblick geben, wie die Struktur dieser Diagrammart aufgebaut ist und einen Kontrast von der Programmierung her zu den anderen Darstellungsmöglichkeiten zu verdeutlichen. Aufzufinden sind die weiteren Befehle in der Dokumentation von JPGraph, welche sich auf der JPGraph Homepage befinden. Oder im *doc* Ordner, welcher bei der Installation der JPGraph Library mit entpackt wurde.
Der Vorteil bei der Verwendung eines Liniendiagrammes gegenüber anderen Diagrammen wäre eine bessere Übersicht über Steigungen und Verläufe über einem bestimmten Zeitpunkt. In unserem Beispiel „Bundestagswahl-Generator“ haben wir damit eine Übersicht über die Anzahl der Nichtwähler über die Jahre hinweg geschaffen. Die Wahl des Diagramms ist aber abhängig vom Geschmack.

4.4 2D-Tortendiagramm

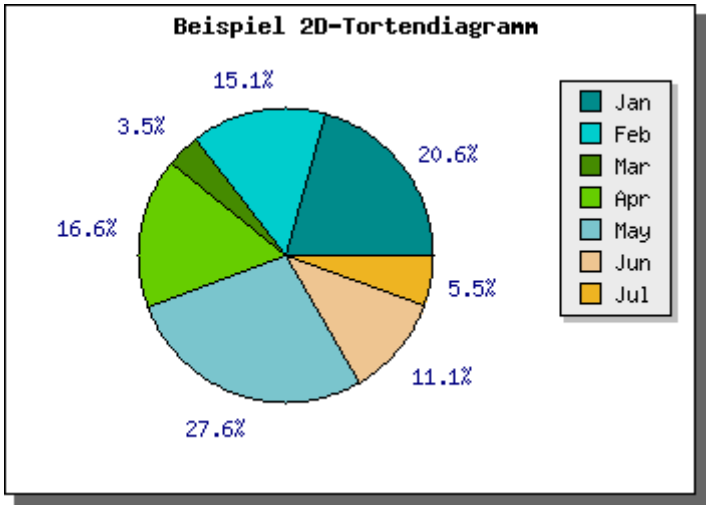
Hierbei möchten wir mehr auf die Unterschiede eingehen gegenüber dem Liniendiagramm, als auf die Struktur des Tortendiagramms direkt. Deshalb ist dieses Beispiel auch sehr einfach gehalten.

```
include ("src/jpgraph.php");
include ("src/jpgraph_pie.php");

$data = array(41,30,7,33,55,22,11);

$graph = new PieGraph(350,250,"auto");
$graph->SetShadow();

$graph->title->Set("Beispiel 2D-Tortendiagramm");
$graph->title->SetFont(FF_FONT1,FS_BOLD);

$format = new PiePlot($data);                          //1
$format->SetLegends($gDateLocale->GetShortMonth());    //2
$format->SetCenter(0.4);
```

[5] Ein Preset ist nicht weiter als eine vorgegebene Einstellung, die nur abgerufen werden müssen. In diesem Falle wäre blue ein Preset für den Hexadezimalcode #0000FF

```
$graph->Add($format);
$graph->Stroke();
```

Wie schon oben genannt wird hier die Datei *jpgraph_pie.php* eingebunden, die Tortendiagramm-Datei. Bei 1) findet man nun einen neuen Befehl, der dazu dient das Tortendiagramm zunächst zu erstellen. Der Unterschied hierbei ist die automatische Umwandlung der Werte in Prozentzahlen. Dabei werden alle Werte zusammengezählt, und die Gesamtzahl wird durch jeden einzelnen Wert dividiert und mit 100 multipliziert. Diese Werte werden dann an die ausgegebene Grafik zu seinem jeweiligen Stück zugewiesen.



Bei 2) wird eine Legende zur Grafik hinzugefügt. Dabei wird ein Konstruktor verwendet (`$gDateLocale`) . In der aktuellen Version wird dieser Konstruktor übrigens automatisch erstellt, man muss ihn nicht mehr initialisieren. Dieser Konstruktor sorgt dafür, dass die Werte der Reihenfolge nach die jeweilige Bezeichnung zugewiesen bekommt. Dies geschieht dynamisch, also könnte man anstelle jeden Wert zusätzlich neu zuzuweisen diesen Preset verwenden. Auffällig ist allerdings der `SetLegends()`-Befehl. Während dieser beim Tortendiagramm auf diese Weise geschrieben wird, wird beim Balken bzw. Liniendiagramm der Befehl `SetLegend()` verwendet. Dies zeigt, dass noch nicht alle Befehle einheitlich gehalten sind, und man die Befehle für jede Diagrammart nochmal nachschlagen sollte. `GetShortMonth()`; ist ein Array-Preset, welche alle 12 Monate in Kurzform als Strings abgespeichert hat (beginnend von 0 à Jan bis 11 à Dec) . Also besitzt JPGraph Kontruktor und Array-Presets, die viel umständliche Programmierung abnehmen können. Diese sind ebenfalls alle in der Dokumentation nachschlagbar.

Vorteile eines 3D-Diagramms wären eine bessere Übersicht, wie groß ein Anteil verhältnismäßig sich auf die Gesamtzahl auswirkt und direkte Vergleiche zu ziehen.

Es gibt allerdings Visualisierungsmöglichkeiten, die auf anderen basieren. Beispiel hierfür wäre das 3D-Tortendiagramm.

4.5 3D-Tortendiagramm

Die 3D-Tortendiagrammdatei ist eine Erweiterung seiner 2D-Variante. Dies bedeutet, wenn ein 3D-Tortendiagramm erstellt werden soll, muss auch die 2D-Datei mit eingebunden werden. Im Beispiel Code sind die Werte des vorherigen Graphen übernommen worden. Es

sollen hier lediglich ein paar neue Möglichkeiten der Darstellung zur Anschauung gebracht werden.

```
include ("src/jpgraph.php");
include ("src/jpgraph_pie.php");
include ("src/jpgraph_pie3d.php");

$data = array(41,30,7,33,55,22,11);

$graph = new PieGraph(330,200,"auto");
$graph->SetShadow();

$graph->title->Set("Beispiel 3D-Tortendiagramm");
$graph->title->SetFont(FF_FONT1,FS_BOLD);

$format = new PiePlot3D($data);
$format->SetCenter(0.45);
$format->SetLegends($gDateLocale->GetShortMonth());
$format->SetAngle(30);                                    //1
$format->ExplodeSlice(1);                                 //2

$graph->Add($format);
$graph->Stroke();
```

Als zusätzliche Funktion ist die Neigung 1) hinzugekommen. Da jetzt eine 3D Perspektive vorhanden ist muss diese nun definiert werden. Die Neigung ist auch gleichzeitig der Vorteil an einem 3D-Diagramm. Mit dieser „Technik“ könnte man kleinere Anteile größer darstellen lassen als sie eigentlich sind. Ebenfalls hinzugefügt wurde der `ExplodeSlice();`-Befehl, welcher eine leichte Abkapselung einzelner Tortenfragmente dient. Der Effekt kommt in der 3D Perspektive besser, er ist jedoch auch in der 2D Variante möglich. Da jetzt ein grober Überblick über JPGraph gegeben wurde, kommen wir aber zum eigentlichen Thema: der Visualisierung von Datenbanken

5 Bundestagswahl-Generator

Direktbeispiel von Datenbankvisualisierung

Ziel bei diesem Generator war es, eine dynamische Grafikdatei zu erstellen, welche sich immer an den Datenbankwerten orientiert, die auf Knopfdruck zufälligen Werten zugewiesen wurden. Oder anders gesagt: Es soll eine realistische Bundestagswahl generiert werden, und deren Ergebnisse sollen auf diverse Grafiken abgebildet werden. Hier beschränken wir uns lediglich auf die Übertragung der Werte auf ein Balkendiagramm. Zunächst haben wir unsere alte PHP-Homepage, die wir im Unterricht erarbeitet haben, kopiert, um

Verbindung.php

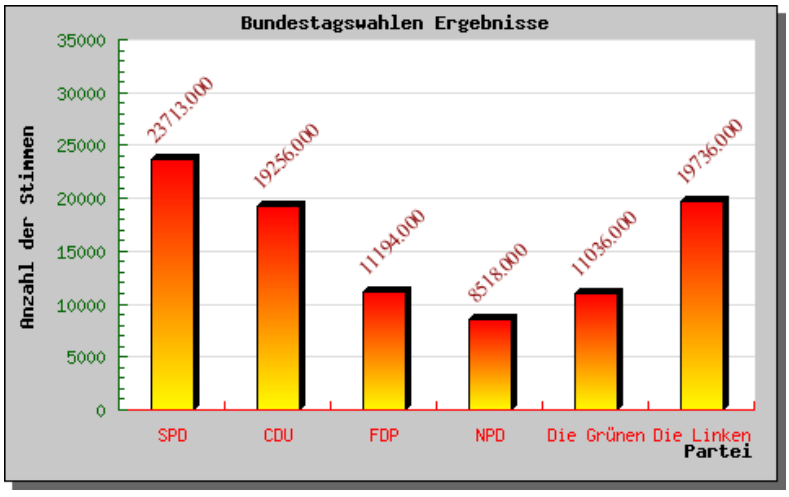
```
$host           = "localhost";
$user           = "root";
$password       = "";
$dbname         = "facharbeit";

$verbindung =
mysql_connect($host, $user,
$password)   or die ("DATABASE:
You failed!");

$select =
mysql_select_db($dbname,
$verbindung)  or die("DATABASE:
You Fail!");
```

ein grobes Layout vorliegen zu haben. Diese wurde darauf auf unsere benötigten Felder angepasst. Darauf folgte die Erstellung einer neuen Datenbank mit den Namen „*Facharbeit*“. Da bei der Erstellung der Grafiken kein HTML-Header erlaubt war, wurde eine separate PHP-Datei erstellt, um eine Verbindung mit dem MySQL-Server herzustellen. Nun folgte der Wahlgenerator.

Auszug aus wahlrandom.php

```php
$sql = "DROP TABLE wahlergebnis";
$ergebnis = mysql_query($sql,$verbindung)
  or die("ERROR: Fail.");

$sql = "CREATE TABLE wahlergebnis (ID INTEGER,
partei VARCHAR(255), wahl INTEGER)";
$ergebnis = mysql_query($sql,$verbindung)
  or die("ERROR: Fail.");

//SPD Wahl
$wahl = mt_rand(1000,30000);
$sql = "INSERT INTO wahlergebnis VALUES (null,
'SPD','$wahl')";
$ergebnis = mysql_query($sql,$verbindung)
  or die("ERROR: Fail.");

//CDU Wahl

...
```

Hierbei wird jedesmal die alte Tabelle durch eine neue ersetzt, wenn diese Datei ausgeführt wird. Die Werte werden über den `mt_rand`-Befehl immer einem zufälligen Wert zwischen 1000 und 30000 zugewiesen.
Die Min und Max-Werte wurden abhängig der Bundestagswahl 2005 gemacht. So wird verhindert, das zum Beispiel die NPD 80% der Stimmen bekommt.
Da wir nun über dynamische Werte verfügen und die Datenbank fertiggestellt wurde, kann nun die Grafik angepasst werden. Hierbei bedienen wir uns einfacher MySQL-Anweisungen, die über PHP erteilt wurden. Ein Array wird dabei erstellt, welche die Daten aus der Datenbank abruft. Dieser Array wird dann als Wertetabelle für das Diagramm verwendet. Zusätzlich haben wir noch die x-Achse so programmiert, das sie ihre Größe automatisch sich der Anzahl der Parteien anpasst. So können immer neue Parteien hinzugefügt werden, ohne dass die Übersicht verloren geht. Zu guter letzt haben wir uns eine weitere Bedingung gesetzt, wobei sich die Skala ebenfalls anpassen soll, und den Parteinamen unter ihren jeweiligen Balken anzeigen lassen soll. Für eine bessere Übersicht haben wir den Quellcode formatiert und sämtliche Anweisungen kommentiert. So soll nochmal alle Funktionsweisen im Überblick bleiben. Zusätzlich wurden ein paar weitere Funktionen verwendet.



```php
include ("src/jpgraph.php");
include ("src/jpgraph_bar.php");      //Balkendiagramm
include ("verbindung.php");           //Verbindung zum Server

//Dynamische x-Achse vorbereiten; jede Partei verlängert die x-Achse um 80
$sql = "SELECT * FROM wahlergebnis";
$ergebnis = mysql_query($sql);
$anzahl = mysql_num_rows($ergebnis);
$xachse = $anzahl * 80;

// Neuen Graph bilden
$graph = new Graph($xachse,300,"auto");

//Dynamische Skala festlegen
$sql = "SELECT * FROM wahlergebnis";
$ergebnis = mysql_query($sql);
$i = 0;
while ($zeile = mysql_fetch_array($ergebnis) )
{
    $partei[$i]=$zeile["partei"];
    $i++;
}
$graph->SetScale("textlin"); //x = Text; y = Linear
$graph->xaxis->SetTickLabels($partei); //Skala wird mit Array beschriftet

       //Formatierung
//Grafikname
$graph->title->Set("Bundestagswahlen Ergebnisse");

//Festlegung der Titel-Schriftart
$graph->title->SetFont(FF_FONT1,FS_BOLD);

//Achsenbeschriftung
$graph->xaxis->title->Set("Partei");
$graph->yaxis->title->Set("Anzahl der Stimmen");

//Schriftart der einzelnen Achsen
$graph->yaxis->title->SetFont(FF_FONT1,FS_BOLD);
$graph->xaxis->title->SetFont(FF_FONT1,FS_BOLD);

//Definition des Randes, damit Schriften nicht überlappen  (Links, Rechts,
Oben, Unten)
$graph->img->SetMargin(40,20,20,40);

//Verschieben der y-Achsenbeschriftung
$graph->yaxis->SetTitleMargin(50);

//Werte festlegen
$sql = "SELECT * FROM wahlergebnis";
$ergebnis = mysql_query($sql);
$i = 0;
while ($zeile = mysql_fetch_array($ergebnis) )
{
    $wahl[$i]=$zeile["wahl"];
    $i++;
}
//Umwandlung in Graph
$bplot = new BarPlot($wahl);

//Balkenfarbefarbe (JP Graph Presets oder Hex-Code) erfolgt nach der
Erstellung des Graphen
$bplot->SetFillColor('red');
```

```
//ODER es ist auch ein Farbüberlauf möglich (1. Farbe, 2. Farbe,
Ausrichtung)
$bplot->SetFillGradient("red","yellow",GRAD_HOR);

//Anzeige der Werte des jeweiligen Balkens
$bplot->value->Show();

//y-Fläche erhöhen um %
$graph->yaxis->scale->SetGrace(35);

//Um die angezeigten Werte über den Graphen zu drehen, muss zunächst eine
Schriftart festgelegt werden
$bplot->value->SetFont(FF_TIMES);

// Nun folgt eine Rotation um 45°
$bplot->value->SetAngle(45);

//Die angezeigten Werte können ebenfalls angepasst werden. Man könnte so
zum Beispiel die Nachkommatastelle verändern. Dies benutzen wir allerdings
als Darstellung von einer Rundung, weil man bei den Wahlen nicht die
exakten Werte übernimmt
$bplot->value->SetFormat('%01.03f');

//oder die Farbe:
$bplot->value->SetColor("darkred");

//Schatten ist ebenfalls möglich
$bplot->SetShadow();

//x und y-Achse Farben festlegen
$graph->xaxis->SetColor("red");
$graph->yaxis->SetColor("darkgreen");

//Gegeben falls Schatten einrichten
$graph->SetShadow();

//Graph in die Grafik einfügen
$graph->Add($bplot);
// Graph ausgeben
$graph->Stroke();
```

6 Schluss

„Zusammenfassung, Reflexion des Themas und viel mehr“

6.1 Zusammenfassung und abschließende Überlegungen

„Zusammenfassung

Das Thema setzte sich von vorherein mit einem großen Aufwand an praktischer Arbeit mit auseinander in der wir auf der einen Seite durch unser Wahlbeispiel begegnet sind. Techniken wurden somit einzeln erklärt und anhand von Quellcodekommentaren ersichtlich gemacht. Somit wurde es klar inwiefern eine fertige Bibliothek dem Nutzer einen gewissen Vor- bzw. Nachteil in der Benutzung verschafft um auch vielleicht in Zukunft dynamische Diagramme durch Datenbanken erstellen zu können.

6.2 Schlussfolgerungen über das Thema hinaus

„In wie weit spielen heute Darstellungen von Datenbankinhalten eine Rolle?“

In der heutigen Informationsgesellschaft spielen visuelle Darstellungen zunehmend eine große Rolle, da anhand von ihnen wichtige Entscheidungen getroffen werden können oder wenn sie einfach den Betrachter über eine bestimmte variable Sachlage informieren wollen (wie z.B. in unserem Fall die Wahlergebnisse). Statische und adynamische erscheinende Zahlen werden somit zu einer schnell erkennbaren Aussage. Dadurch ist es sehr wichtig geworden große Datenbanken die über weitaus viele Statistiken enthalten mit einem äußeren Erscheinungsbild zu versehen, wie es zum Beispiel täglich an der der Börse der Fall ist. Ohne solche Ausschlaggebenden Grafiken würden sich somit Entscheidungen verlangsamen und damit das tägliche Geschäft an der Börse zum Beispiel hemmen. Dadurch ist es von ein großen Bedeutung einen Blick hinter die „Kulissen“ zu werfen um zusehen und zu verstehen wie Diagramme basierend auf Datenbankinhalten erschlossen werden können.

6.3 Reflexion über das eigene Vorgehen und die angewandten Verfahren

„In wie weit hätten man unser Vorgehen noch verbessern können?“

Durch unsere pragmatische Trennung von vornherein haben wir nicht das maximale an Informationsgehalt aus uns beider Vorgehen schöpfen können somit war nie so richtig klar in wie weit die praktische beziehungsweise theoretische Arbeit vorangeschritten war um aufgrund von diesen vielleicht neuen Erkenntnissen weiter zuarbeiten. Aber im Großen und Ganzen gehen wir davon aus dem wir den Hauptbestandteil eben durch unseren praktischen Teil erfasst haben um somit den Leser dieser Arbeit die grafische Darstellung von Datenbankinhalten näher zu bringen.

6.4 Quellen und Aufgabenteilung

Formaler/Theoretischer Teil: Nikolas Kittler (Punkte 1. bis 3.)

Praktischer Teil: Timm Schütte (Punkte 4. bis 5.)

http://www.binnendijk.net/jpgraph/

http://www.easy-coding.de/

http://code.google.com/intl/de-DE/apis/visualization/documentation/using_overview.html

JPGraph Dokumentation

http://ezcomponents.org/docs/api/latest/introduction_Graph.html