

# **WISSENSCHAFTLICHES ARBEITEN UND SCHREIBEN**

Stefanie Scherzinger, Uni Passau  
Wolfgang Mauerer, OTH Regensburg  
WS 2020/21

Oktober 2016, Anzeige in der Rundschau



**WAS BEDEUTET "WISSENSCHAFTLICH"?**

# Schüler und Schülerinnen testen selbstgebaute Flugobjekte

22.07.2016

Von: Dr. Dagmar Stark, Dipl. Päd. Univ. Armin Gardeia

**Im Rahmen des Natur- und Technik-Unterrichts besuchen Schüler und Schülerinnen das Labor Windkanal und Strömungsmesstechnik der OTH Regensburg. Ein Testflug mit selbstgebauten Flugzeugen ist ein Highlight für die Kinder der sechsten Klasse des Privatgymnasiums Pindl.**

Beim ersten Termin im Labor Windkanal und Strömungsmesstechnik informierten Prof. Dr. Stephan Lämmlein und Laboringenieur Siegfried Schrammel über die Theorie des Fliegens und welche Prinzipien dahinter stecken, damit ein Flugobjekt vom Boden abhebt. Einige Experimente im Windkanal verdeutlichen den Besucherinnen und Besuchern die theoretischen Vorstellungen. Mit der Hausaufgabe, Flugobjekte selbst zu bauen um sie dann an der OTH Regensburg zu testen, verließen die Kinder die Hochschule.



Im Labor Windkanal und Strömungsmesstechnik besprechen die Kinder mit Laboringenieur Siegfried Schrammel die Theorie des Fliegens.



Schüler stoppen mit Lehrerin Dr. Dagmar Stark die Zeit ihrer Flugzeuge.

Beim zweiten Termin ihres Natur und Technik-Themas „Flugtechniken“ hatten die Schülerinnen und Schüler ihre selbstgebastelten Modellflugzeuge dabei, die im Foyer getestet wurden. In einem ersten Durchgang ging es darum, die Zeit zwischen zwei Messpunkten zu stoppen, um damit die Datengrundlage für spätere Geschwindigkeitsberechnungen zu erhalten. Im zweiten Durchgang wurde aus lichter Höhe die Datengrundlage zur Ermittlung der Sinkgeschwindigkeit festgehalten.



Schüler messen die Sinkgeschwindigkeit ihrer Flugobjekte von der höchsten Brücke im Foyer des Hörsaalgebäudes am Forum.  
Fotos: Junge Hochschule

**Beschreibung unpräzise**  
→ Experimente nicht reproduzierbar!

**journalistischer Stil,  
aber nicht wissenschaftlich**



# Experimente

Welche Hardware?  
Welche Software?

The test environment consists of a cluster with 6 nodes. Each node consists of 4 CPUs, in total 150 GB RAM and 3.8 TB storage. YARN as the resource manager was configured to allocate maximum 12 GB RAM for each user because the development cluster was used by other test developments simultaneously.

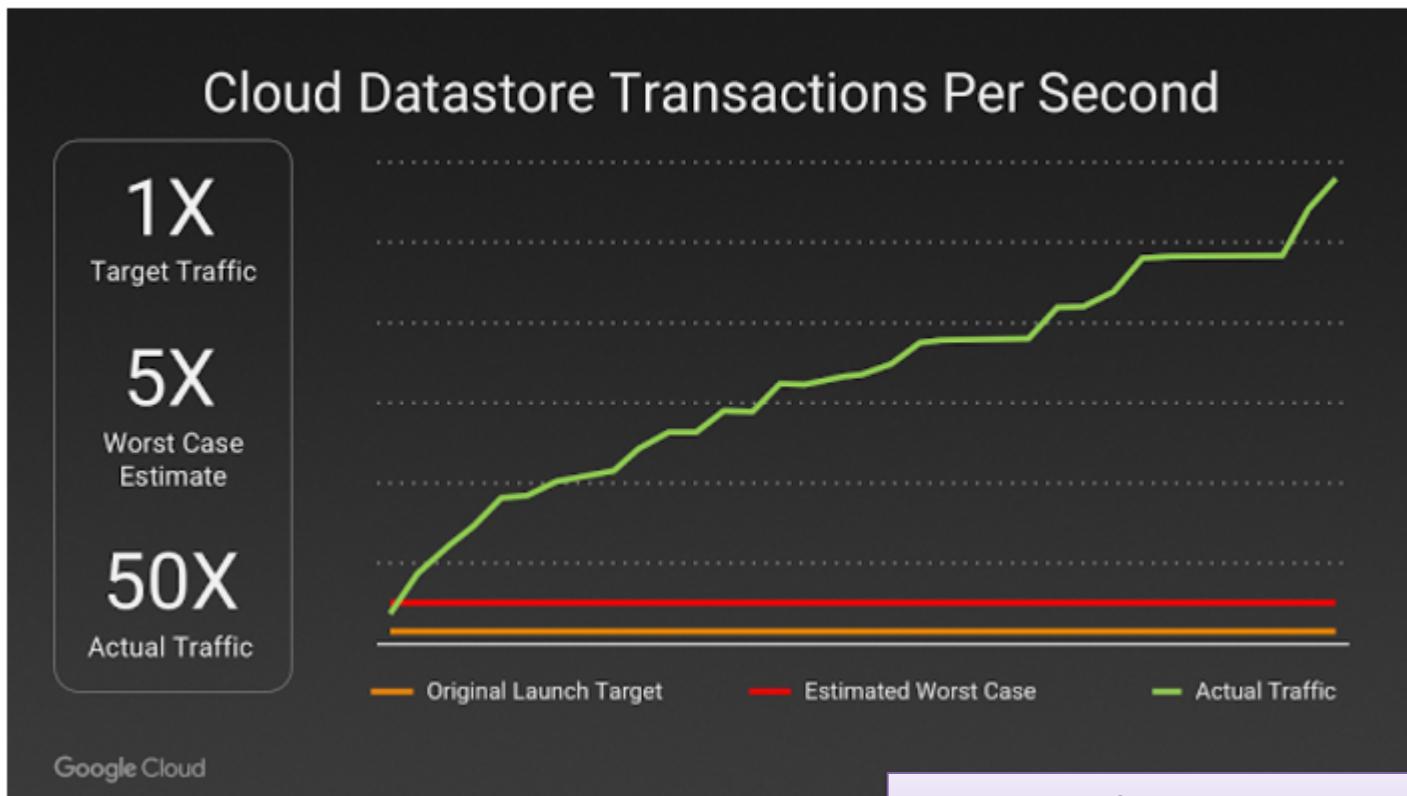
The execution times in the next section are the mean value of five measures. Only this mean value is provided because the deviation of measures was about five percent which is insignificant for the comparison of the examined technologies.

Was wurde gemessen?

Unklar: Wie wurde die Ausführungszeit gemessen?

# Pokemon Go! auf Google Cloud Datastore

As a teaser, I'll start with a picture worth a thousand words:



To create your first Web Application, select **File > New > Web Application Project** from the Eclipse menu.

In the **New Web Application Project** wizard, enter a name for your project and a java package name, e.g., `com.example.mywebapp`. You can specify a project ID or choose not to (see [Setting Project ID in New Web Application Project wizard](#) for more details). Click **Finish**.



**Note:** Using Java keywords as a project name (e.g. "New" and "Class") will result in various build errors. Avoid these where possible.

Congratulations, you now have an App Engine and GWT-enabled web application!

Eine Anleitung  
aber keine wiss. Arbeit

## Running your Web Application locally

Right-click on your web application project and select **Debug As > Web Application** from the popup menu. You are testing your application using the [Java Development Server](#).

This action creates an Eclipse **Web Application** launch configuration for you and launches it. The web application launch configuration will start a server, whose URL will appear in the Development Mode view at the bottom of the Eclipse window. Select the URL and click **Copy** in the context menu, and then open it in your web browser to load the application.

At this point, you can set breakpoints, inspect variables and modify code as you would normally expect from a Java Eclipse debugging session.

## Deploying your Web Application

To deploy your web application, you need to create an App Engine project (not to be confused with the Eclipse project) in the [Google Developers Console](#). If you already have an App Engine project, then you can skip this step.

Once you have an App Engine project ID, right-click on your Eclipse project, and select **Google > App Engine Settings...** from the context menu. Enter your project ID into the **Project ID** text box. Click **OK**.

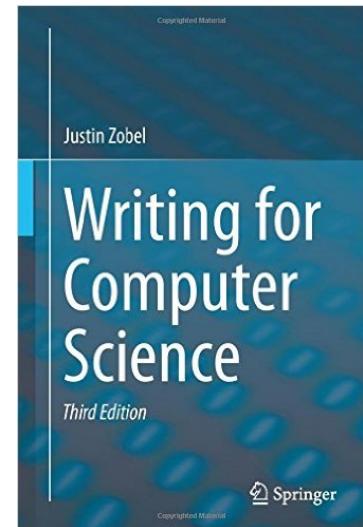
Right-click on your project and select **Google > Deploy to App Engine**. If you haven't already signed in using your Google account, you will be prompted to do so. See [Sign In](#) for more details.

Click **Deploy**.

[https://developers.google.com/eclipse/docs/getting\\_started](https://developers.google.com/eclipse/docs/getting_started)  
Go to <http://project-id.appspot.com/> to see your application.

# Wissenschaftliches Arbeiten und Schreiben

- Kann man lernen! Es gibt gute Bücher, auch in unserer Bib!
- Gehört zu den Qualifikationen, die man von Ihnen erwartet, wenn Sie einen Hochschulabschluss haben.
- Wenn Sie Karriere machen wollen, müssen Sie sich schriftlich präzise ausdrücken können.



# Die Abschlussarbeit als Prüfungsleistung

# Wussten Sie's?

Name des Studiengangs: Bachelor Informatik (PO: 20122)	Modulname: Bachelor-Arbeit		
Modulbezeichnung (ggf. englische Bezeichnung) <b>Bachelor-Arbeit</b>	Modul-KzBez. oder Nr. <b>31</b>		
Modulverantwortliche/r <b>Vorsitzender der Prüfungskommission</b>	Fakultät <b>Informatik und Mathematik</b>		
Studiensemester gemäß Studienplan <b>7.</b>	Studienabschnitt <b>3.</b>	Modultyp <b>Pflicht</b>	Arbeitsaufwand [ECTS-Credits] <b>12</b>

**Aha!**

- Thema gut wählen!
- Betreuer gut wählen!
- Mögl. keine Nebenkriegsschauplätze,  
damit Sie sich fokussieren können



30 Arbeitsstunden (netto!)

60 Arbeitsstunden

15 Tage (a 8h)

→ 9 Wochen (a 5 Tage)

→ 2,25 Monate (a 4 Wochen)

# SPO BSc Informatik



## § 10 Bachelorarbeit

- (1) In der Bachelorarbeit sollen die Studierenden ihre Fähigkeit nachweisen, die im Studium erworbenen Kenntnisse und Fertigkeiten zur Bearbeitung einer komplexen fachwissenschaftlichen Aufgabenstellung selbstständig anzuwenden.
- (2) Zur Bachelorarbeit kann sich anmelden, wer sich im dritten Studienabschnitt befindet (§ 7 Abs. 4) und den Praxisteil des zweiten Studienabschnitts erfolgreich absolviert hat.

### Aha!

- Look&Feel einer **wissenschaftlichen Arbeit**  
z.B. Literaturverzeichnis
- **Learning by Doing:**  
Ihre Betreuer coachen Sie,  
zunächst mehr, dann weniger

Bachelorarbeit soll so beschaffen sein, dass die Arbeit bei zusammenhängen in der Regel in drei Monaten fertig gestellt sein kann. Die Frist von der Fristgabe darf fünf Monate nicht überschreiten. Die Möglichkeiten einer Frist: die APO, § 19 Abs. 3 c.

# SPO BSc Informatik

1	2	3	4	5	6	7	8	9	10
Modul Nr.	Modulbezeichnung (englische Bezeichnung)	SWS*	Credits*	Art der Lehrveranstaltung	Prüfungen <sup>2</sup>			Ergänzende Regelungen	Notengewicht*
					mündlich schriftlich Dauer in Min.	Studienbegleitende Leistungsnachweise	Zulassungsvoraussetzungen		
1	Datenverarbeitungssysteme (Computer Systems)	6	8	SU, Ü, Pr	schrP, 90-120		LN		1
2	Grundlagen der Informatik (Fundamentals of Computer Science)	6	8	SU, Ü	schrP, 90-120		LN		1
3	Physik (Physics)	4	5	SU, Ü, Pr	schrP, 90-120		LN		1
4	Fachspezifisches Englisch (English)	4	4	SU, VU, Ü	schrP, 90-120		TN		1
5	Betriebswirtschaftslehre (Business Administration)	4	5	SU, Ü	schrP, 90-120		LN		1
6	Programmieren 1 (Programming 1)	6	8	SU, Ü, Pr	schrP, 90-120		LN		1

29	Bachelorarbeit (Bachelor's Th)	Eine 1,0 in der Bachelorarbeit ist so, als hätten Sie eine 1,0 in <ul style="list-style-type: none"> <li>• Datenbanken und in</li> <li>• Betriebssysteme und in</li> <li>• Software Engineering</li> </ul> (jeweils Gewicht 2)	Ausarbeitung		6
30	Bachelorseminar (Bachelor Sem)	R	Aha! "4 gewinnt" ist hier die falsche Strategie	der ior- notw.	-

# Effektive Selbstsabotage

# Rechtschreibung & Grammatik

## 3 Azure

Microsoft Azure [7] ist eine Cloud-Computing-Plattform von Microsoft für Softwareentwickler, welche verschiedene Dienste wie SQL Azure, AppFabric oder Webservices unter einem Cloud-Betriebssystem namens Winows Azure zur Verfügung stellt. Beim Cloud-Computing werden die netzbasierten Anwendungen für den Benutzer bereitgestellt, welche von überall der Welt erreichbar sind. Des weiteren sind alle Dateien und Konfigurationen nicht mehr auf dem System des Entwicklers abgelegt, sondern auf den Servern von Azure.

Die wichtigsten Funktionen von Microsoft Azure unterteilen sich in Compute, Netzwerk, Speicher, Web- und mobile Anwendungen und Datenbanken. In diesem Kapitel bezieht sich das Abschnitt 3.1 auf die Benutzung der Speicher und Warteschlagen Funktionen.

### 3.1 Speicher und Warteschlange

Bei der Kategorie Speicher handelt es sich **Hauptsächlich** um das sogenannte Azure Storage, einem dauerhaften, hoch verfügbaren und in hohem Maße skalierbaren in der Cloud abgelegten Speicher.

Das Azure Storage besitzt wiederum mehrere Arten wie dem Blob Storage für unstrukturierte Daten oder dem Table Storage für NoSQL-Daten, jedoch wird nur auf das Warteschlagen- oder *Queue*-Storage weiter eingegangen.

Um einen Cloudspeicher anzulegen muss zuerst ein Azure Storage Account erstellt werden. Diesem Account muss ein eindeutiger Namespace Name zugewiesen werden, welcher es ermöglicht, **Datenobjekte in das Storage zu schreiben oder aus diesem zu Lesen.**

Nachdem ein Storage Account existiert, kann auf diesem im nächsten Schritt der benötigte Typ installiert werden.

# Denglisch

## 2.1 Datenstruktur

### 2.1.1 Inverted Index

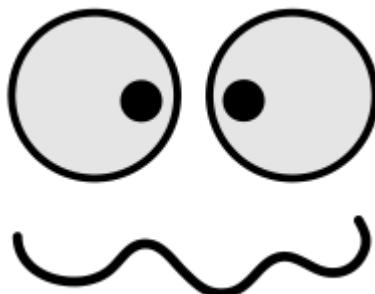
Ein invertierter Index ist eine Datenstruktur bei welcher ein Mapping erstellt wird, in dem einzelne Wörter zu den jeweiligen Dokumenten, in denen diese auftreten und die einem Index hinzugefügt sind, gemappt werden. Ein so genanntes Posting, welches den indizierten Wert des Dokuments wieder spiegelt und aus verschiedenen Datentypen wie einem String, Integer oder ähnlichem besteht wird beispielsweise als einfacher binärer Wert abgelegt, der angibt ob sich ein bestimmter Suchbegriff in einem Dokument befindet oder nicht. Zusätzlich können noch andere Werte abgespeichert werden, welche einen tieferen Einblick in die indizierten Daten gewähren können, unter anderem die Häufigkeit des auftretenden Wertes im Dokument und ein Offset für diese, der angibt wo es sich befindet. Im Gegensatz zu einem normalen Forward Index kann man hierbei beispielsweise alle Dokumente in einem Index finden die einen bestimmten Term beinhalten. Da der Zugriff bei einem Inverted Index auf einem einzelnen zu suchenden Schlüssel beruht wird die Ablage des Indexes meist entweder sortiert oder als Hashtabelle abgelegt. Hierbei wird der Fokus auf unterschiedliche Kriterien festgelegt:

#### Update des erstellten Mappings

Da die abgespeicherten Werte zu groß für den Arbeitsspeicher sind, werden diese auf einem weiteren Speichermedium, wie einer Festplatte, abgelegt. Da dort alles sortiert gespeichert wird ist die schlechteste und durchschnittliche Dauer hierfür  $O(\log n)$ , wobei  $n$  die Anzahl der vorherigen, bereits zum Index hinzugefügten Dokumente ist.

Der **Webjob** wird nur ausgeführt, wenn eine **Message** in der **Queue** für diesen **Webjob** hinterlegt wird. Sobald er diese prozessiert hat, wird der **Webjob** wieder heruntergefahren.

Bei einem kontinuierlichen **Webjob** wird dieser einmal gestartet und läuft dauerhaft, sodass keine Verzögerung zwischen Start und Abarbeiten der **Queue** besteht. Falls keine **Nachrichten** in der **Warteschlange** vorhanden sind wird der



Bitte nicht zwischen Synonymen  
(Warteschlange <-> Queue,  
Message <-> Nachricht)  
hin- und herwechseln

Alles nur unter Androhung von  
Zwang erklären...



## Geschwindigkeit beim Zugriff

Hiermit ist die Dauer bei der Suche nach einem bestimmten Term und dem Zugriff auf das Dokument, in dem dieser auftritt, gemeint. Der Zugriffszeit ist wiederum  $O(\log n)$ , wenn man jedoch nur bis zum ersten auftreten dieses Terms sucht kann diese in weniger als  $O(\log n)$  Zugriffen auf die abgelegten Daten resultieren.

## Index Größe

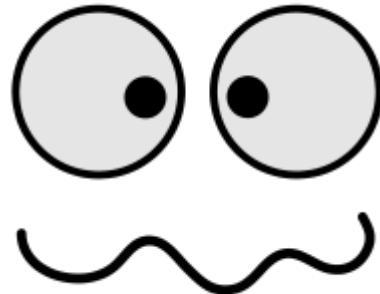
Der benötigte Speicherplatz für den Inverted Index. Da für jedes Posting ein Eintrag gemacht werden **muss**, ist die Größe des Indexes proportional zur Menge der Postings.

## Skalierbarkeit

Der Algorithmus für das Indizieren **muss** mit der zunehmenden Größe gut skalieren. Genauer muss die Verwendung des Hauptspeichers bzw. In-Memory Speichers uneingeschränkt von der kompletten benötigten Speichergröße verwendet werden.

Um die oben genannten Punkte allesamt befriedigend zu erfüllen, **muss** der richtige Algorithmus gewählt sein, wie der Algorithmus des B-Baums auf den im weiteren Näher eingegangen wird.

Es den Lesern unnötig  
schwer machen.



Testdata	Graph1	Graph5	Graph10	Graph30	Graph60
Lot	100	500	500	1,000	2,000
Wafer	10	10	25	25	25
Chips	1,000	1,000	1,000	1,000	1,000
Substrate	12	12	12	12	12
Baseplate	4	4	4	4	4
Vertices	1,100,000	5,500,000	13,000,000	28,000,000	56,000,000
Edges	2,100,000	10,420,000	26,000,000	52,000,000	112,000,000

TABLE 7.1: Specification of different test data sets (part 1)

# Tabellen

**Tabelle 74: Bestand an Arbeitslosen 2004 bis 2009 in Deutschland (Version A)**  
im Jahresdurchschnitt

Arbeitslose	2004	2005	2006	2007	2008	2009
Gesamt .....	4381281	4860880	4487225	3776509	3267907	3423283
Frauen .....	1932563	2254773	2149194	1873453	1599918	1555507
Männer .....	2448719	2605656	2337914	1900322	1667592	1867774

Quelle: *Bundesagentur für Arbeit* (2010): Arbeitsmarkt in Zahlen – Jahressahlen – Arbeitslosigkeit im Zeitablauf, Februar 2010. Nürnberg, ohne Seitenangabe



**Tabelle 75: Bestand an Arbeitslosen 2004 bis 2009 in Deutschland (Version B)**  
im Jahresdurchschnitt

Arbeitslose	2004	2005	2006	2007	2008	2009
Gesamt .....	4.381.281	4.860.880	4.487.225	3.776.509	3.267.907	3.423.283
Frauen .....	1.932.563	2.254.773	2.149.194	1.873.453	1.599.918	1.555.507
Männer .....	2.448.719	2.605.656	2.337.914	1.900.322	1.667.592	1.867.774

Quelle: *Bundesagentur für Arbeit* (2010): Arbeitsmarkt in Zahlen – Jahressahlen – Arbeitslosigkeit im Zeitablauf, Februar 2010. Nürnberg, ohne Seitenangabe



# Tabellen

**Tabelle 74: Bestand an Arbeitslosen 2004 bis 2009 in Deutschland (Version A)**  
im Jahresdurchschnitt

Arbeitslose	2004	2005	2006	2007	2008	2009
Gesamt .....	4381281	4860880	4487225	3776509	3267907	3423283
Frauen .....	1932563	2254773	2149194	1873453	1599918	1555507
Männer .....	2448719	2605656	2337914	1900322	1667592	1867774

Quelle: *Bundesagentur für Arbeit* (2010): Arbeitsmarkt in Zahlen – Jahreszahlen – Arbeitslosigkeit im Zeitablauf, Februar 2010. Nürnberg, ohne Seitenangabe



**Tabelle 76: Bestand an Arbeitslosen 2004 bis 2009 in Deutschland (Version C)**  
im Jahresdurchschnitt

Arbeitslose	2004	2005	2006	2007	2008	2009
Gesamt .....	4 381 281	4 860 880	4 487 225	3 776 509	3 267 907	3 423 283
Frauen .....	1 932 563	2 254 773	2 149 194	1 873 453	1 599 918	1 555 507
Männer .....	2 448 719	2 605 656	2 337 914	1 900 322	1 667 592	1 867 774

Quelle: *Bundesagentur für Arbeit* (2010): Arbeitsmarkt in Zahlen – Jahreszahlen – Arbeitslosigkeit im Zeitablauf, Februar 2010. Nürnberg, ohne Seitenangabe



Das Deutsche Institut für Normung (DIN) und das Statistische Bundesamt empfehlen, die Zifferngruppen nicht durch Punkte, sondern durch Leerzeichen aufzuteilen.<sup>68)</sup>

# Tabellen



**Tabelle 75: Bestand an Arbeitslosen 2004 bis 2009 in Deutschland (Version B)**

im Jahresdurchschnitt

Arbeitslose	2004	2005	2006	2007	2008	2009
Gesamt .....	4.381.281	4.860.880	4.487.225	3.776.509	3.267.907	3.423.283
Frauen .....	1.932.563	2.254.773	2.149.194	1.873.453	1.599.918	1.555.507
Männer .....	2.448.719	2.605.656	2.337.914	1.900.322	1.667.592	1.867.774

Quelle: *Bundesagentur für Arbeit* (2010): Arbeitsmarkt in Zahlen – Jahresszahlen – Arbeitslosigkeit im Zeitablauf, Februar 2010. Nürnberg, ohne Seitenangabe



**Tabelle 77: Unemployment 2004 to 2009 in Germany**

average

Unemployed	2004	2005	2006	2007	2008	2009
Total .....	4,381,281	4,860,880	4,487,225	3,776,509	3,267,907	3,423,283
Women .....	1,932,563	2,254,773	2,149,194	1,873,453	1,599,918	1,555,507
Men .....	2,448,719	2,605,656	2,337,914	1,900,322	1,667,592	1,867,774

Source: *Bundesagentur für Arbeit* (2010): Arbeitsmarkt in Zahlen – Jahresszahlen – Arbeitslosigkeit im Zeitablauf, Februar 2010. Nürnberg, n. p.



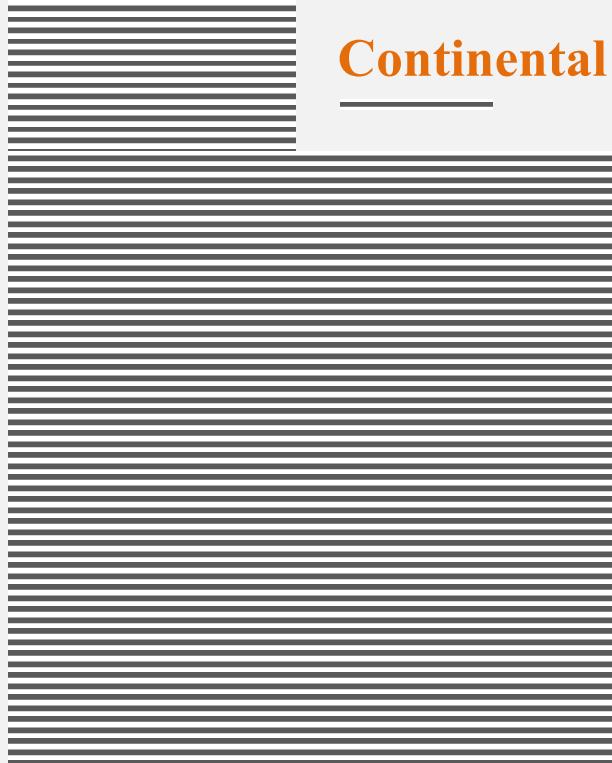
Unbedingt zu beachten ist, dass in englischsprachigen Ländern die Dezimaleinteilung durch Punkte und die Gruppierungen der Zahlen in Drei-Ziffern-Gruppen durch Kommata erfolgt (z. B. USA oder Großbritannien, vgl. Tabelle 77).

Abbildungen/Quellcode  
lieblos "hinklatschen"



# Abbildungen

Kapitel 1: Klassische  
Data Mining Algorithmen



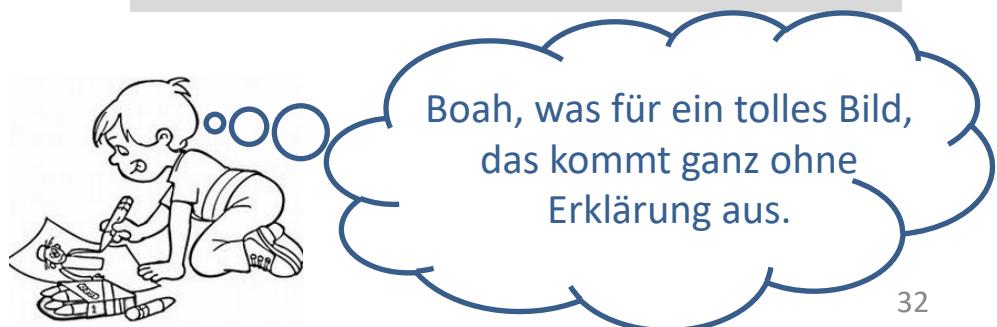
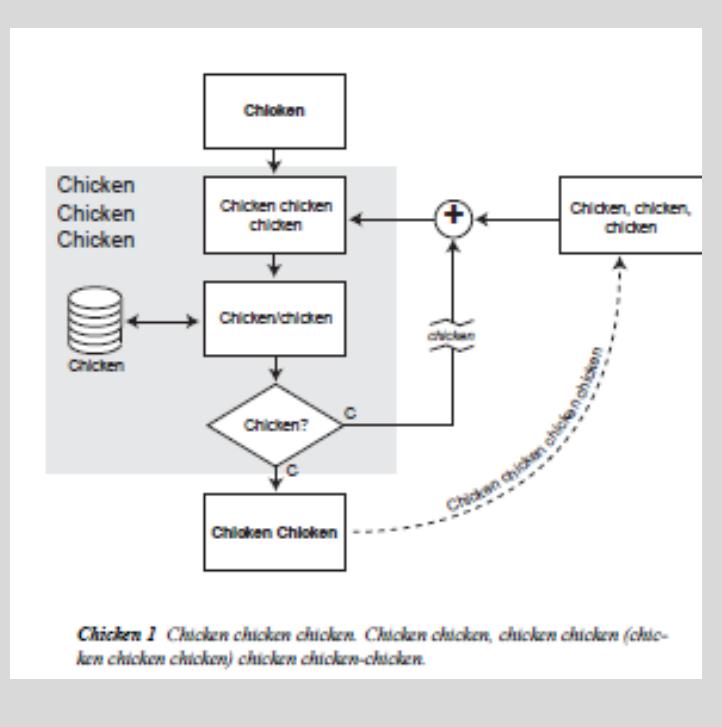
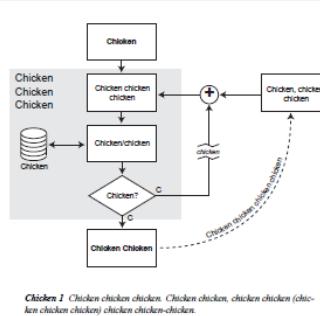
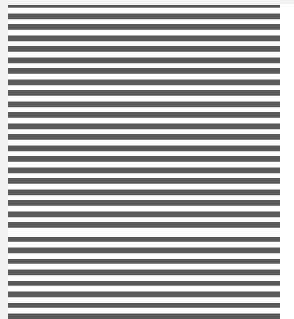
## Continental

Abbildung 1: Das Logo von Continental



# Abbildungen

## Kapitel 1: Einleitung





# Checkliste für Abbildungen

- Die Abbildung liefert einen **Mehrwert**.
  - Es gibt eine passende **Bildunterschrift**.
  - Es wird ggf. die **Quelle** genannt.
- 
- Die Abbildung wird im Text **referenziert**.
  - Der Text erklärt, **was zu sehen** ist.
- 
- Der Text fasst zusammen,  
  **was der Mehrwert der Abbildung** ist.

# Bildunterschrift passt nicht!

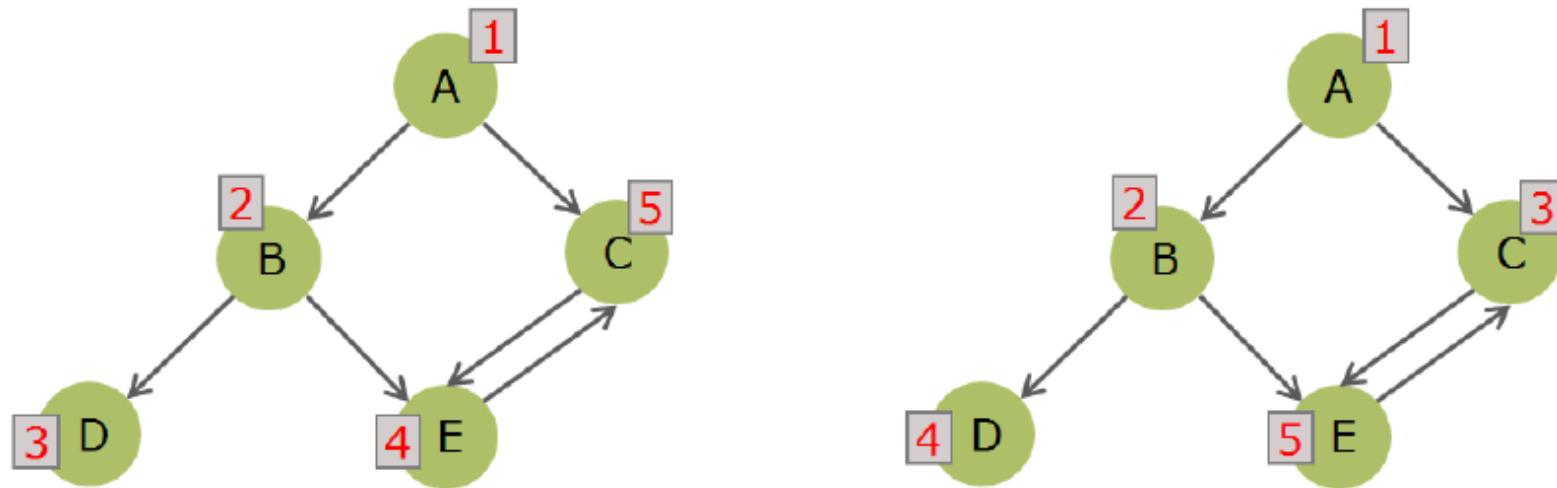


FIGURE 3.9: Depth-first and breadth-first search algorithm



# Abbildungen

## Kapitel 3: Implementierung

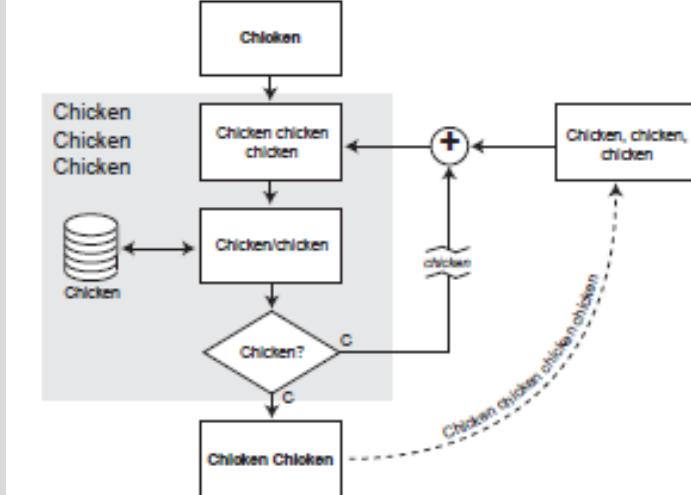
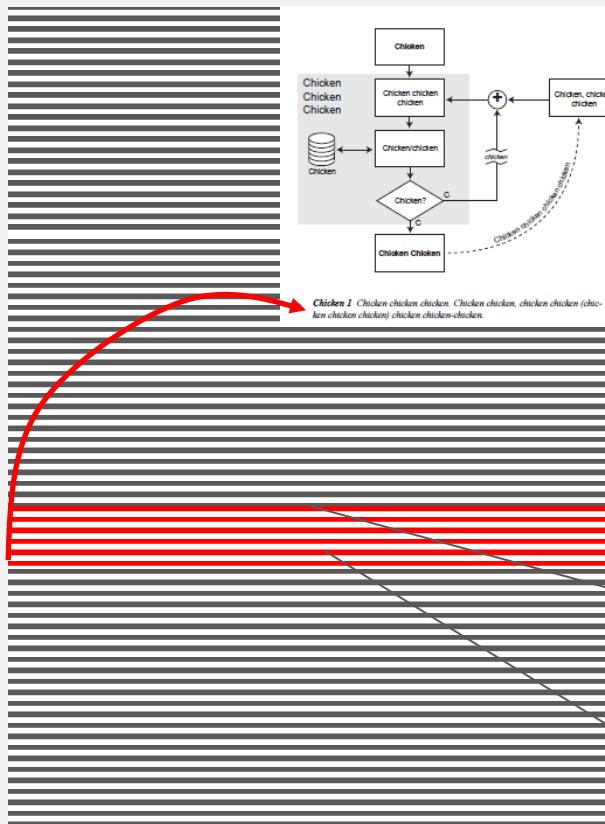
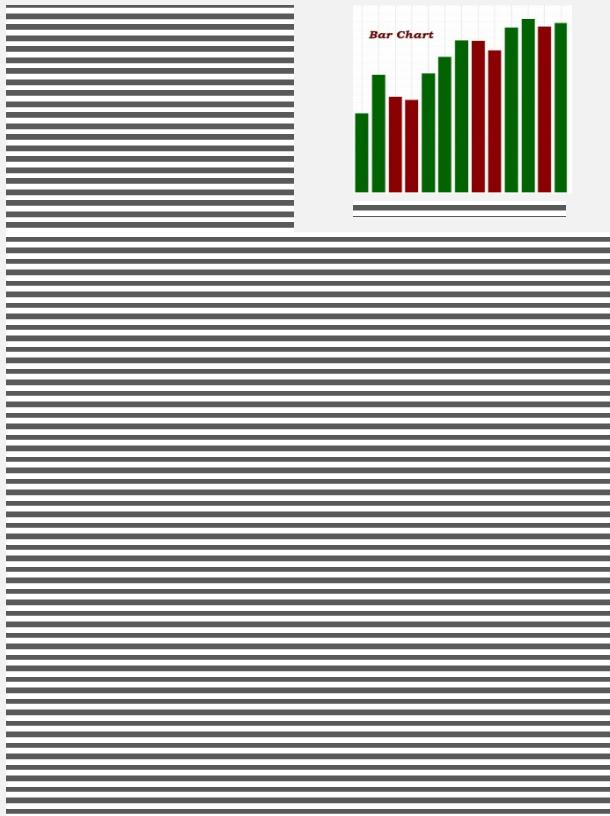


Abbildung 1 zeigt den *Chicken* Ablauf als Flowchart.  
Die *Chicken* Datenbank, links in der Abbildung, verwaltet die *Chicken* Daten<sup>5..</sup>

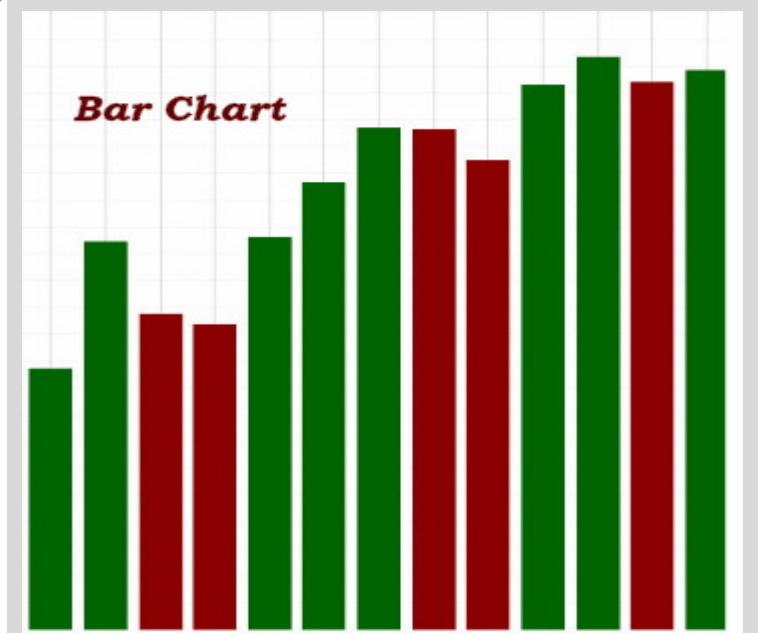


# Abbildungen

Kapitel 7: Evaluierung



**Bar Chart**

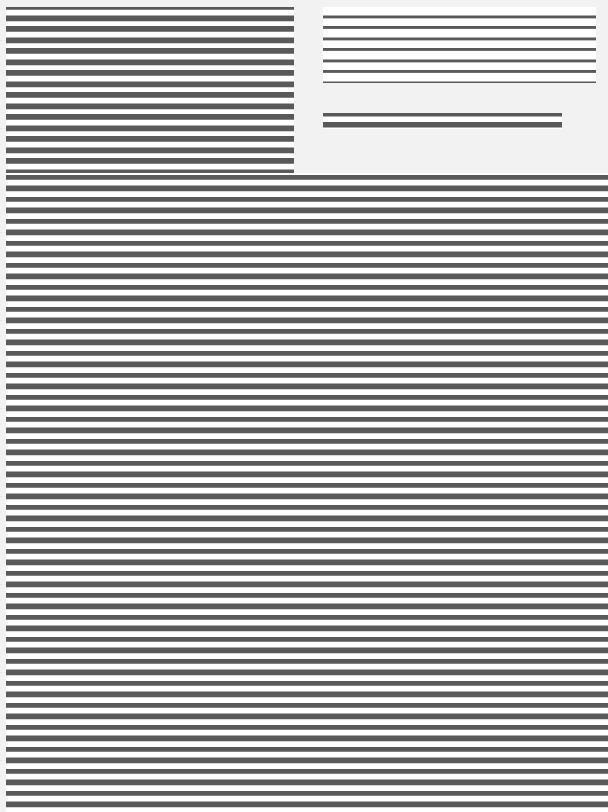


**Abbildung 8:** Rote Balken zeigen Äpfel,  
grüne Balken die Anzahl der Birnen an.



# Quellcode

Kapitel 4: Implementierung



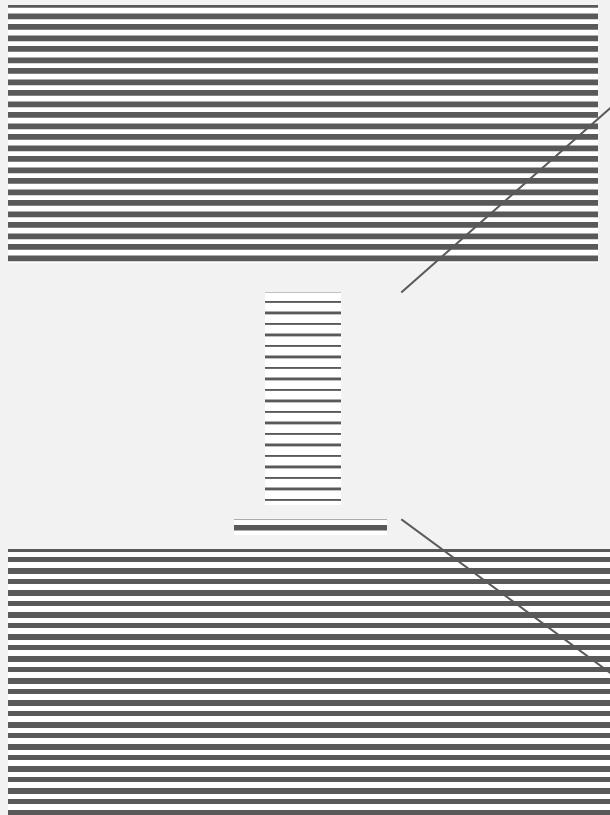
```
#include <stdio.h>
#include <ctype.h>
```

Abbildung 7: Die Include Dateien.



# Quellcode

## Kapitel 2: Hauptteil



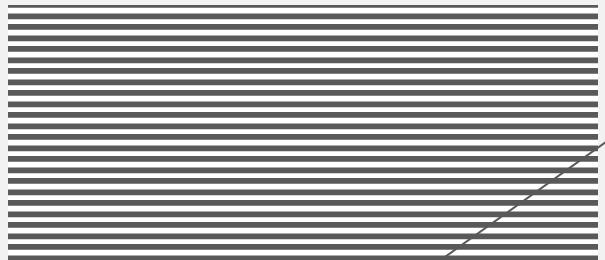
```
int t9(char c) {  
    c = tolower(c);  
  
    switch(c) {  
        case 'a':  
        case 'b':  
        case 'c':  
            return 2;  
        case 'd':  
        case 'e':  
        case 'f':  
            return 3;  
        case 'g':  
        case 'h':  
        case 'i':  
            return 4;  
        case 'j':  
        case 'k':  
        case 'l':  
            return 5;  
  
        default:  
            return 1;  
    }  
}
```

**Abb. 5:** Ein unwesentlicher Programmausschnitt.

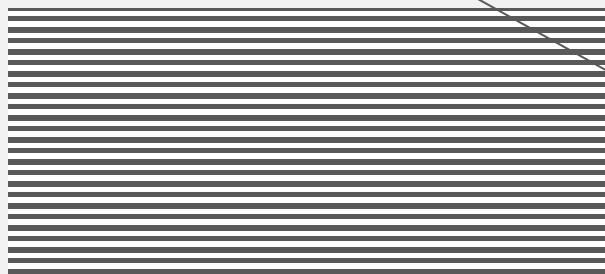


# Quellcode

## Kapitel 2: Hauptteil



```
1 #!/usr/bin/perl -w
2
3 # this
4 # is
5 # a
6 # loop
7
8 for
9 {
10     my
11     $i
12     =
13     0;
14     $i
15     <
16     99;
17     $i++
18 }
19
20 print
21 $i;
22
23
24
```



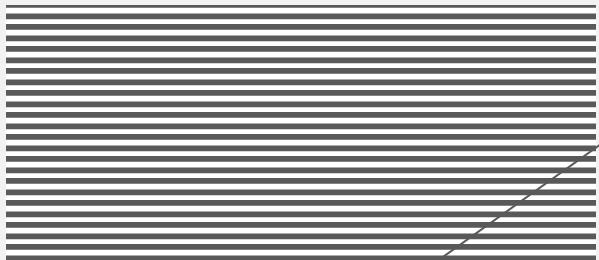
```
1 #!/usr/bin/perl -w
2
3 # this
4 # is
5 # a
6 # loop
7
8 for
9 (
10 my
11 $i
12 =
13 0;
14 $i
15 <
16 99;
17 $i++
18 )
19 {
20 print
21 $i;
22 }
```

**Abbildung 5:** Hier sieht man etwas Code  
(und dass ich nicht einrücken kann).

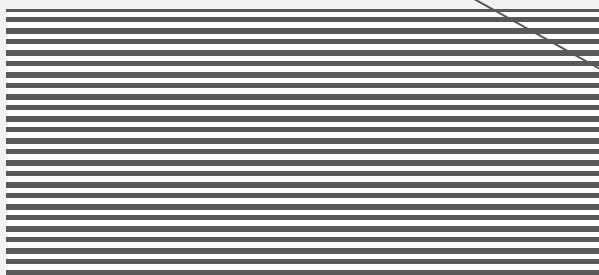


# Quellcode

## Kapitel 2: Hauptteil



```
1 #!/usr/bin/perl -w
2
3 # this
4 # is
5 # a
6 # loop
7
8 for
9 {
10 my
11 $i
12 =
13 0;
14 $i
15 <
16 99;
17 $i++
18 }
19 {
20 print
21 $i;
22 }
23
24
```



```
1 #!/usr/bin/perl -w
2
3 # this
4 # is
5 # a
6 # loop
7
8 for
9 {
10 my
11 $i
12 =
13 0;
14 $i
15 <
16 99;
17 $i++
18 }
19 {
20 print
21 $i;
22 }
23
24
```

Abbildung 5: Hier sieht man etwas Code  
(und dass ich JPEG nicht verstanden habe).

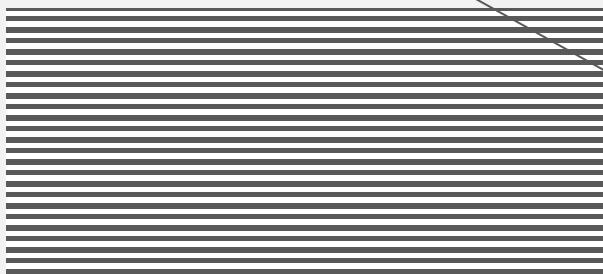


# Quellcode

## Kapitel 2: Hauptteil



```
1 #!/usr/bin/perl -w
2 # this
3 # is
4 # a
5 # loop
6
7 for
8 {
9     my
10    $i
11    =
12    0
13    $i
14    <
15    99;
16    $i
17    print
18    $i
19    }
20
21 }
```



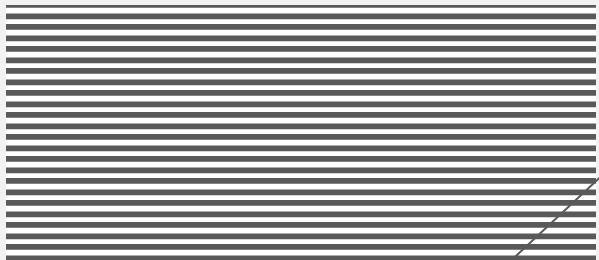
```
1 #!/usr/bin/perl -w
2
3 # this
4 # is
5 # a
6 # loop
7
8 for
9 (
10 my
11 $i
12 =
13 0;
14 $i
15 <
16 99;
17 $i++
18 )
19 {
20 print
21 $i;
22 }
23
24
```

Abbildung 5: Hier sieht man etwas Code  
(und dass meine Rechtschreibkorrektur an ist). 41



# Pseudocode

## Kapitel 2: Hauptteil



```
search(int k) {  
    int li = 0;  
    int re = n-1;  
  
    while (re >= li) {  
        int m = (li + re)/2;  
        if (k == a[m].key)  
            breche ab, da k gefunden;  
        else if (k < a[m].key)  
            re = m - 1;  
        else  
            li = m + 1;  
    }  
    k wurde nicht gefunden;  
}
```

```
search(int k) {  
    int li = 0;  
    int re = n-1;  
  
    while (re >= li) {  
        int m = (li + re)/2;  
        if (k == a[m].key)  
            breche ab, da k gefunden;  
        else if (k < a[m].key)  
            re = m - 1;  
        else  
            li = m + 1;  
    }  
    k wurde nicht gefunden;  
}
```

Suche weiter in linker Hälfte

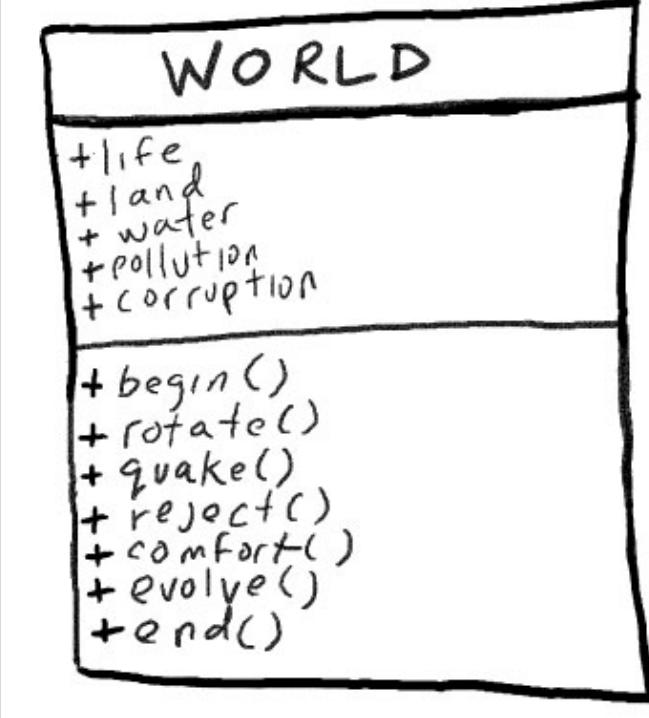
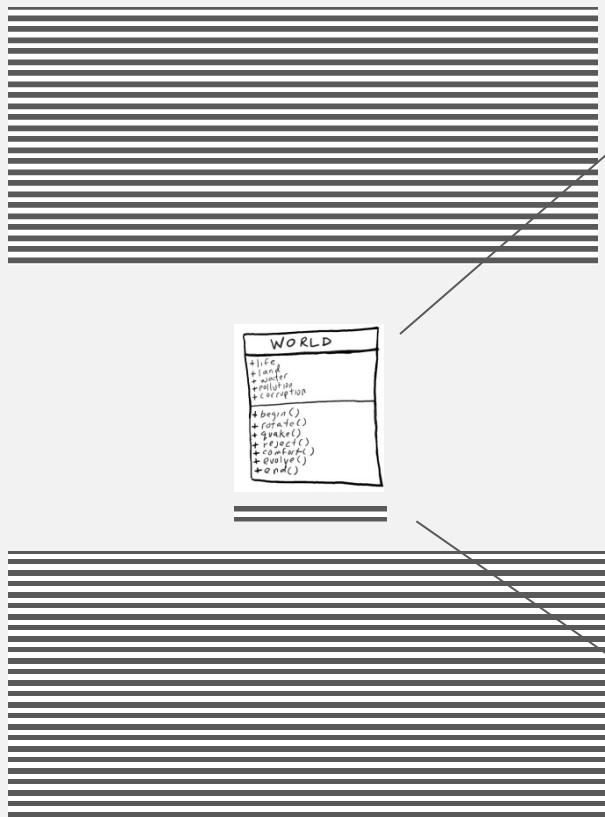
Suche weiter in rechter Hälfte

**Abbildung 5:** Der Algorithmus zur binären Suche.



# UML

## Kapitel 2: Hauptteil

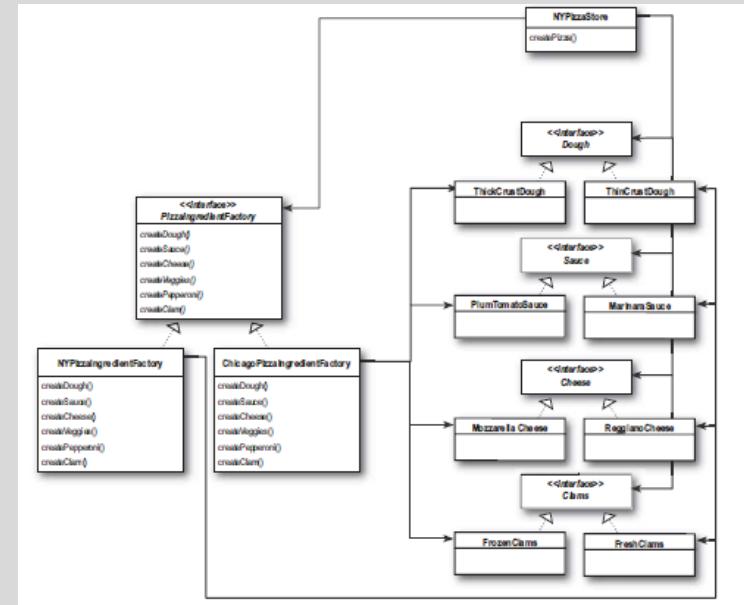
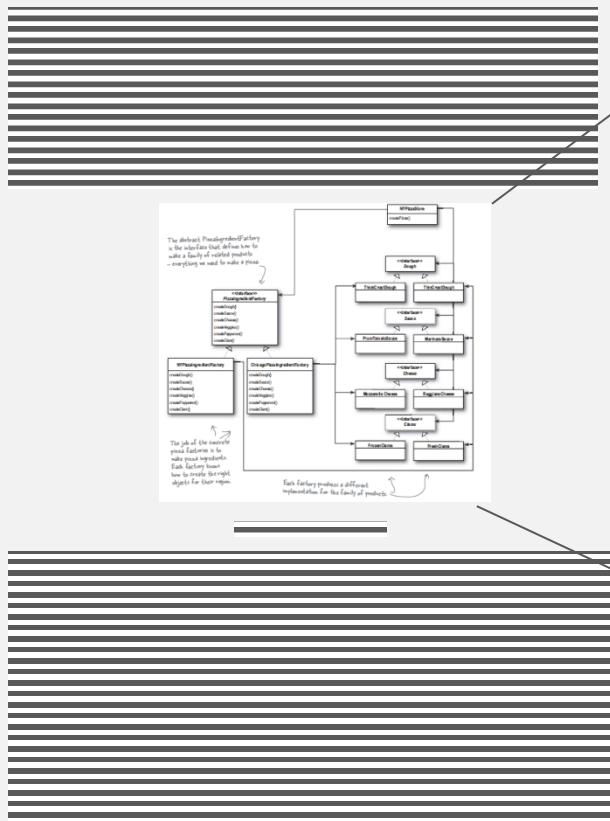


**Abbildung 5:** Diese völlig uninteressante Klasse dient vor allem als Daten-Container.

# UML



## Kapitel 2: Hauptteil



**Abbildung 8:** Das *Abstract Factory* Design Pattern im Pizza-Bestellsystem.

# Charts & Co



# Rug Plot

**Figure 1.** Rug plot of Web-request rates.

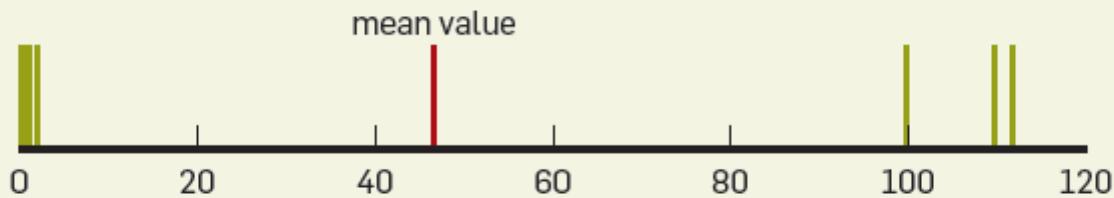


Präziser als nur der Durchschnittswert

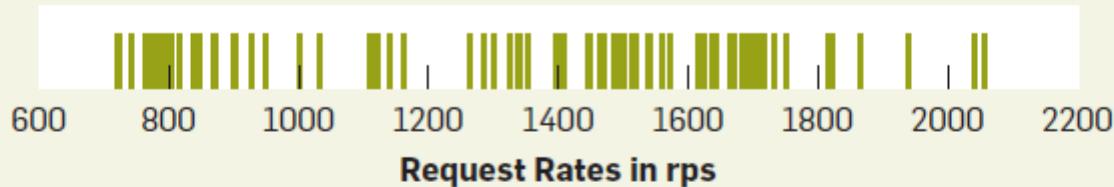
**Tipp:** Mit "Jitter" werden Überlappungen aufgelöst

# Rug Plot

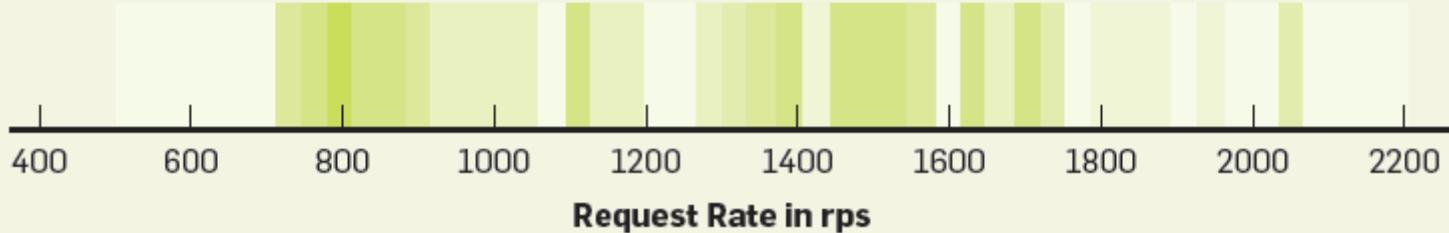
**Figure 9.** Rug plot of a two-modal dataset.



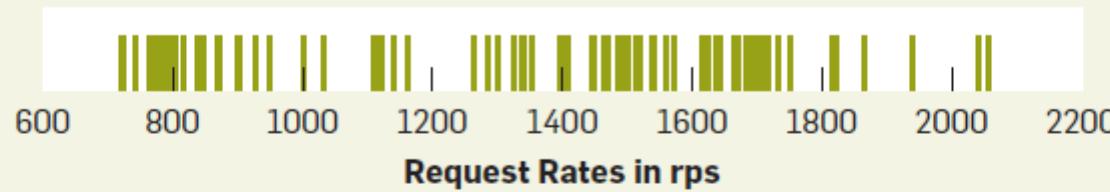
**Figure 1. Rug plot of Web-request rates.**



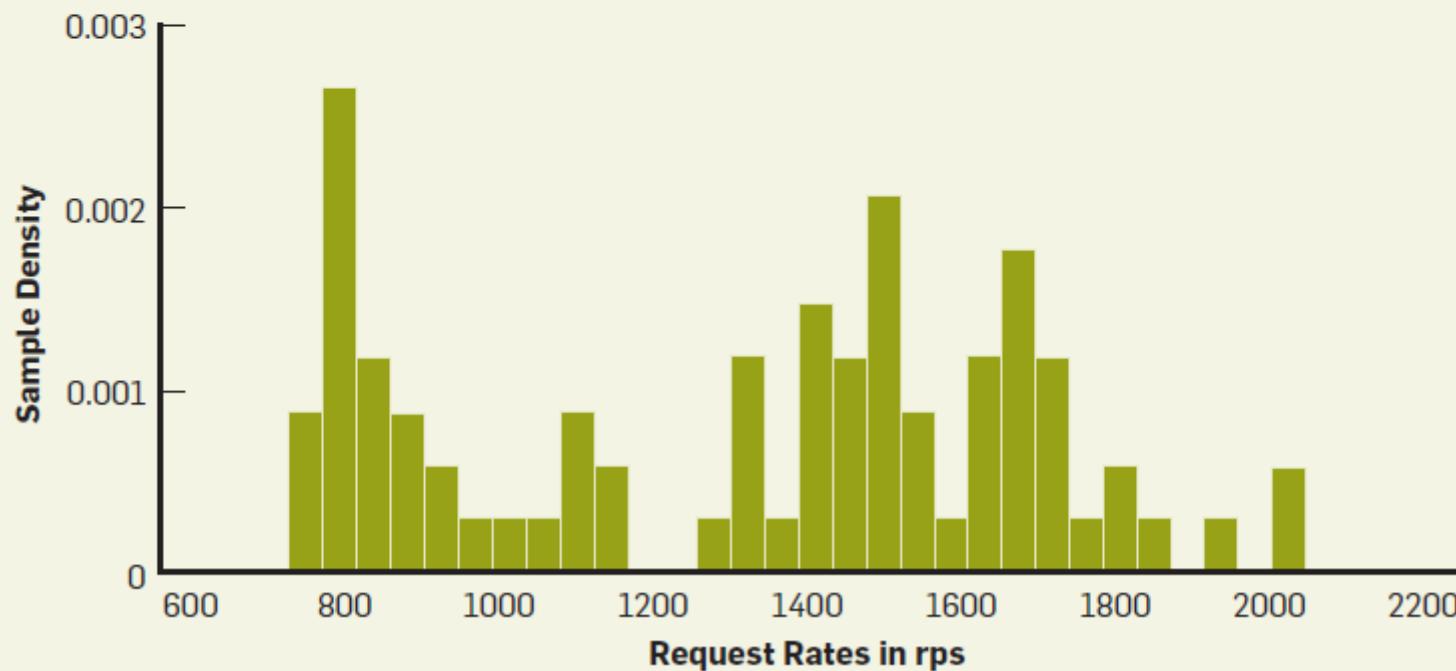
**Figure 7. Request rate histogram (50 bins) presented as a heat map.**



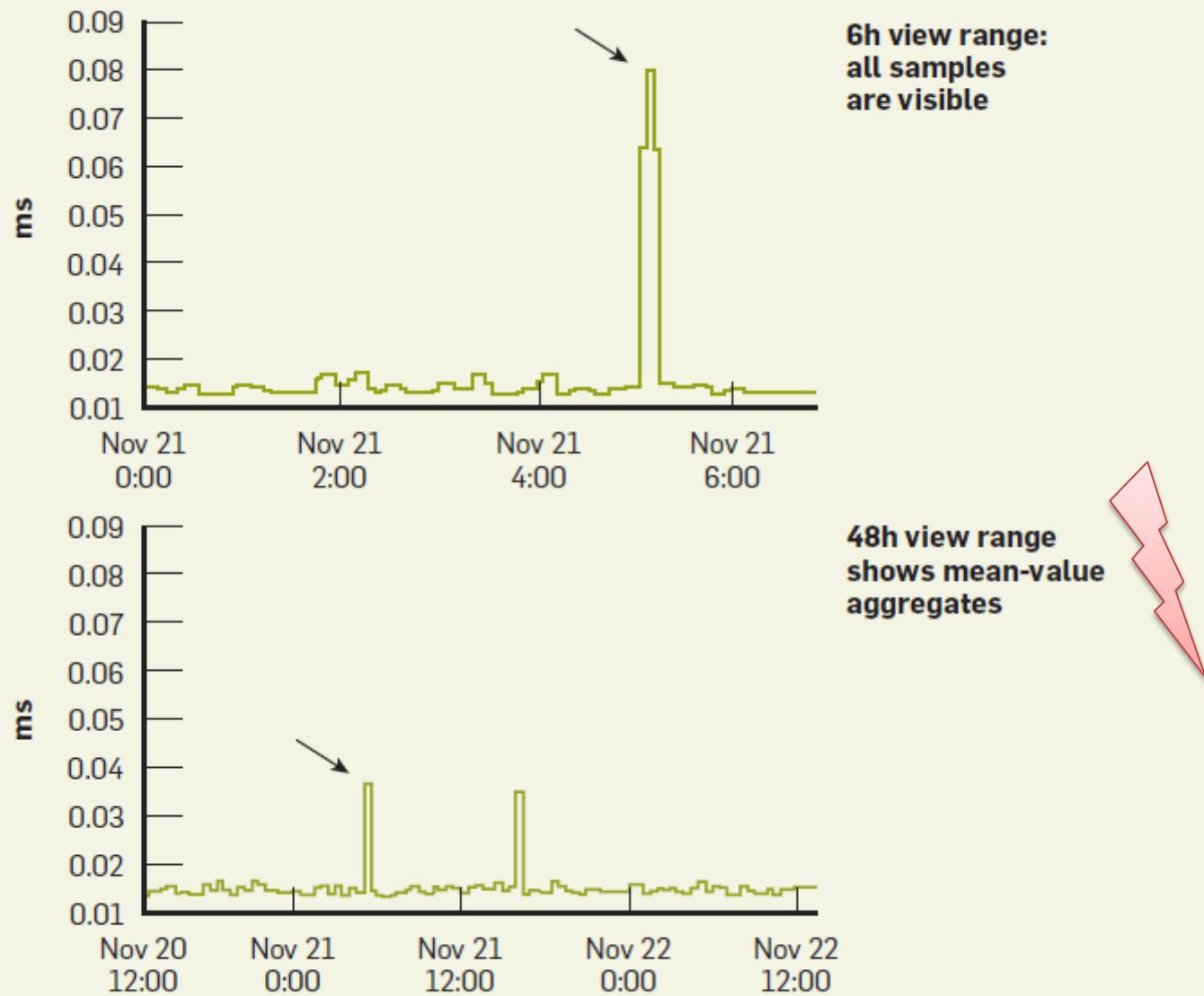
**Figure 1. Rug plot of Web-request rates.**



**Figure 2. Histogram of Web-request rates.**



**Figure 10. Ping latency spike on a view range of 6H vs. 48H.**

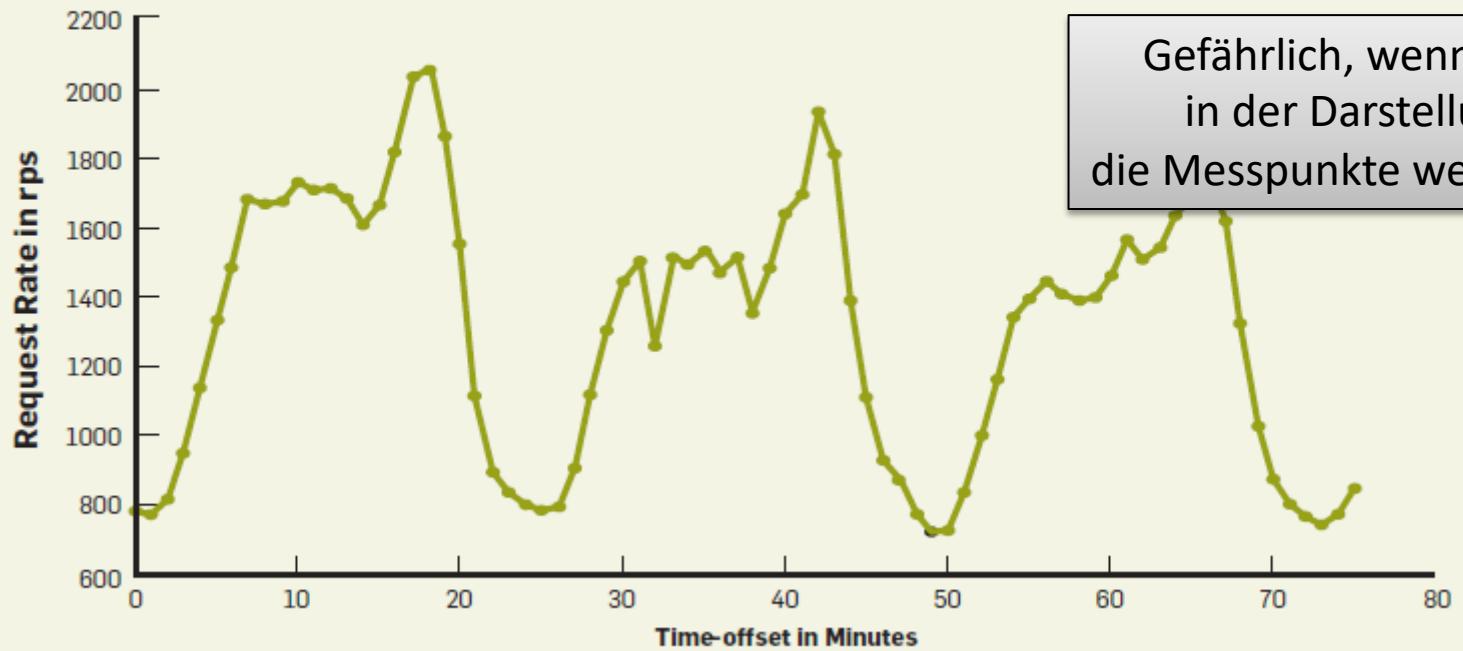


# Line Plot

Figure 3. Line plot of Web-request rates.

Mit Vorsicht zu genießen,  
was passierte zwischen den  
Messpunkten?

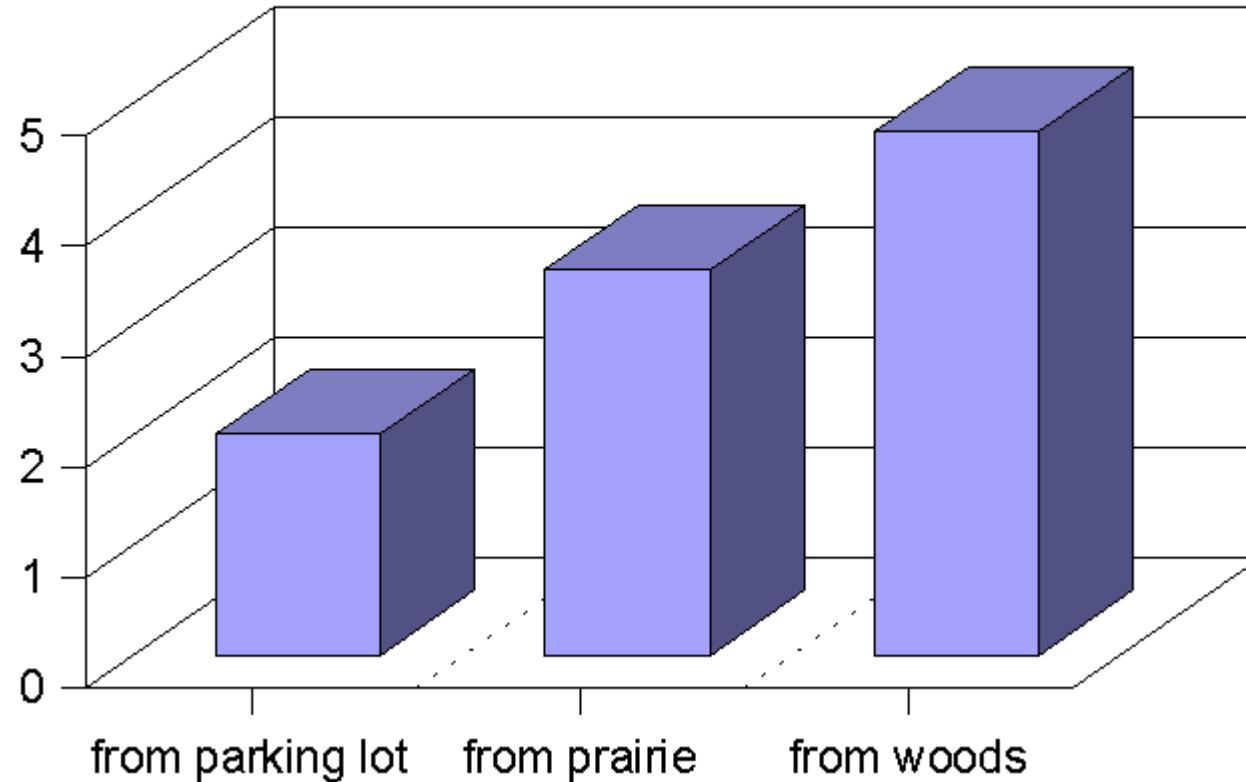
Gefährlich, wenn man  
in der Darstellung  
die Messpunkte weglässt....



# Achtung... was sind die Werte?

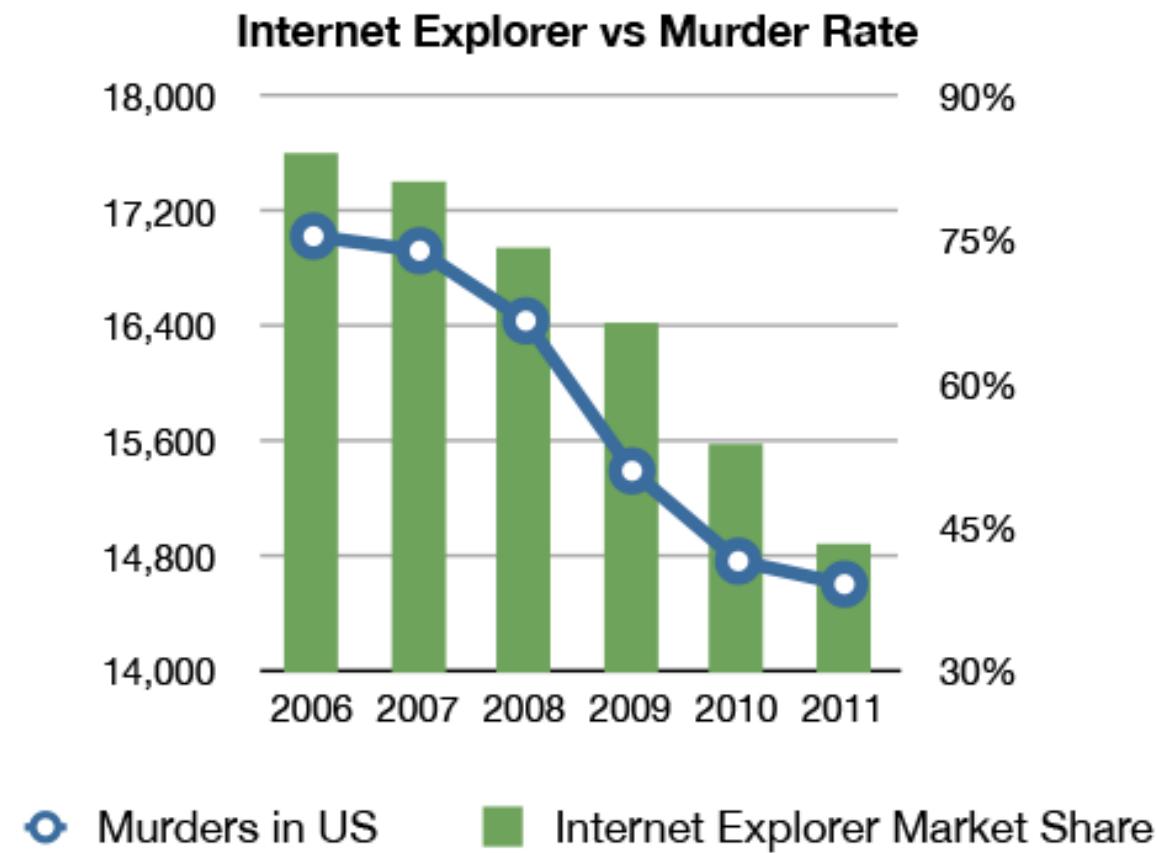


## Leaf Size vs. Location



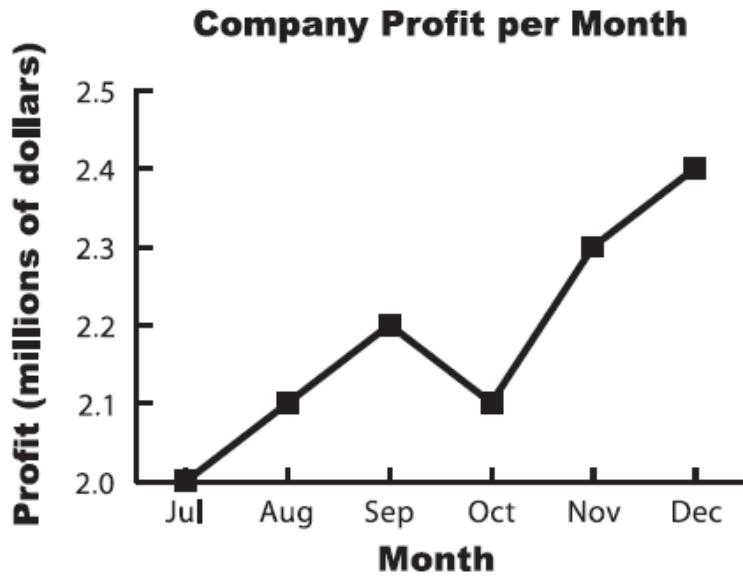
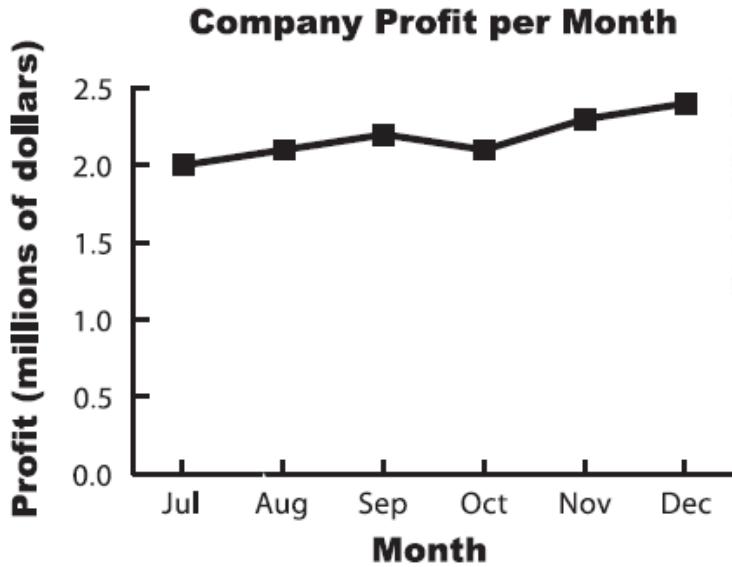


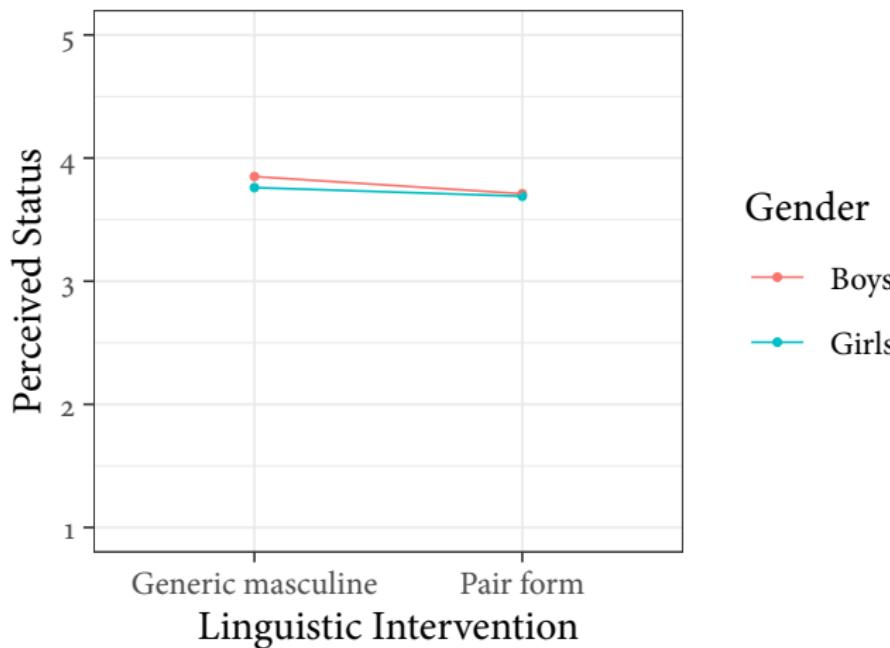
# Korrelation?

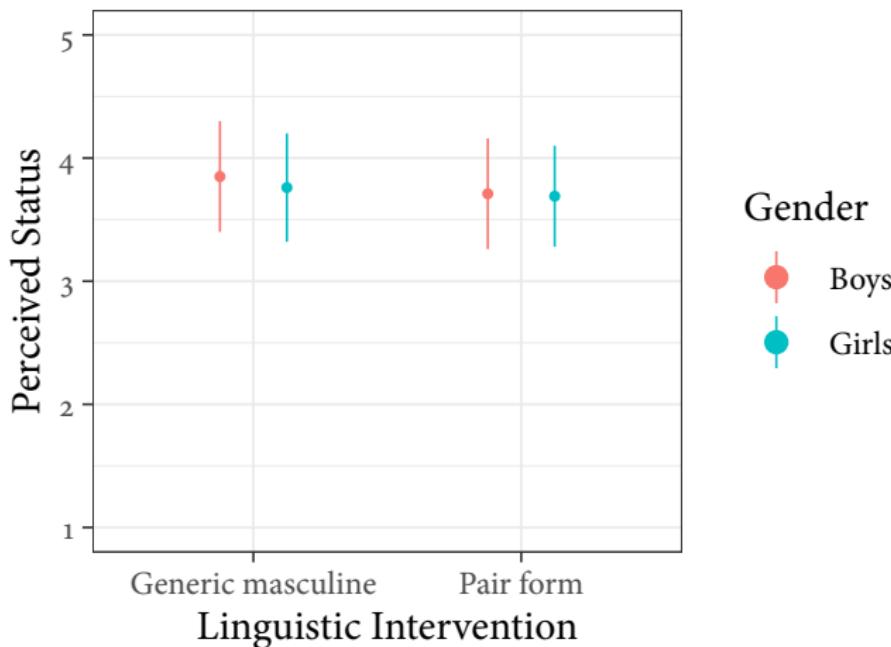


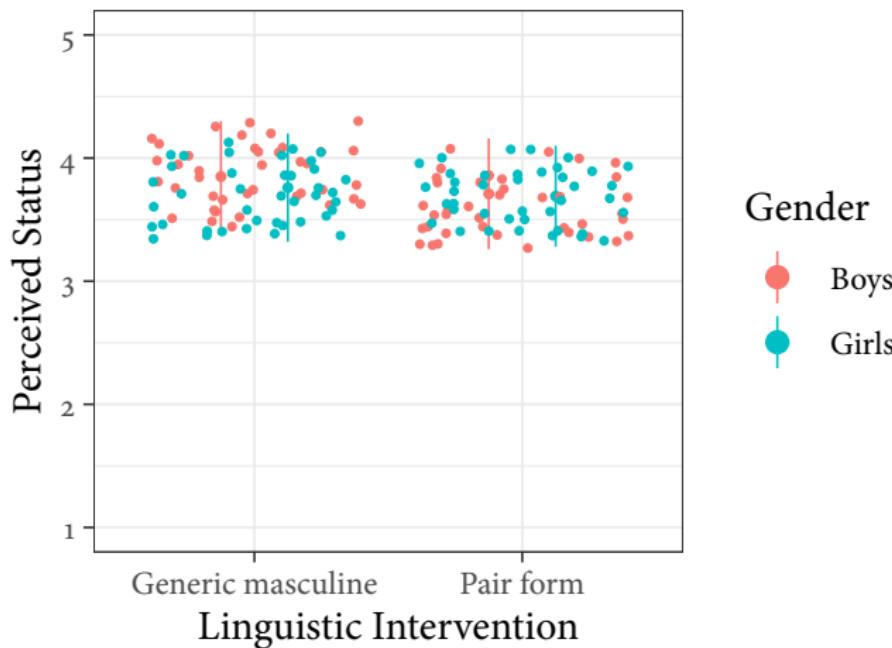


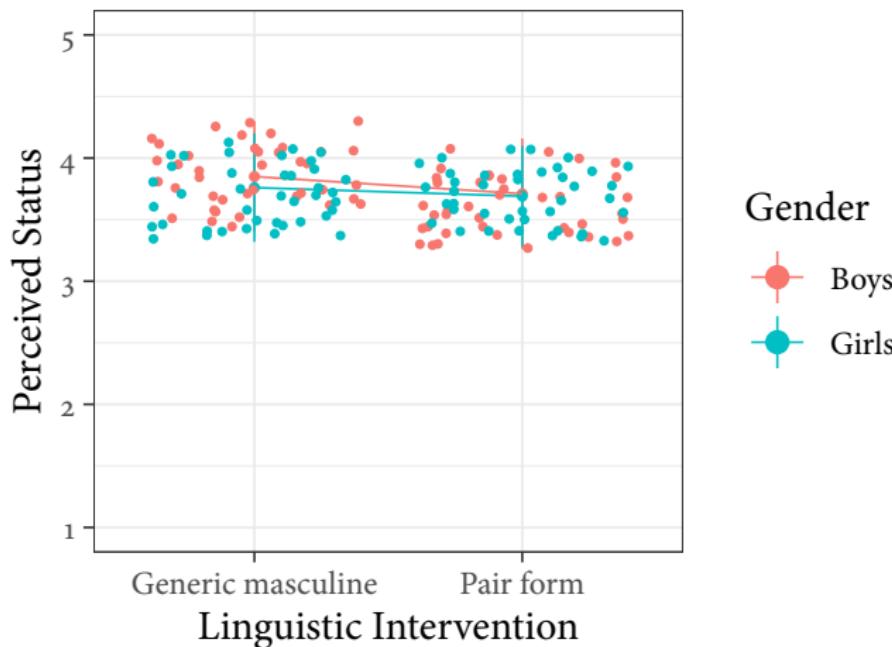
# Manipulation?



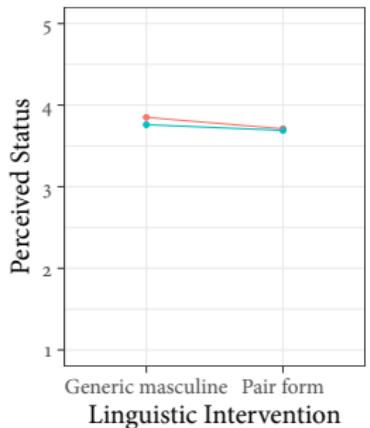




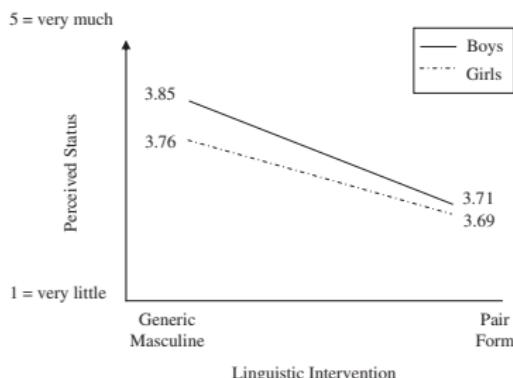




## Unbewusste und bewusste Missinterpretation



Gender  
— Boys  
— Girls



## Ideologie

- ▶ Im Besitz der Wahrheit
- ▶ Selektive Wahrnehmung von Fakten
- ▶ Ignorieren von Unsicherseiten
- ▶ Selbstreferentiell
- ▶ Dichotomisierung
  - ▶ Schwarz/Weiss
  - ▶ “Wir” gegen “Die anderen”
- ▶ Fakten ⇒ alternativlose Handlung

## Ideologie

- ▶ Im Besitz der Wahrheit
- ▶ Selektive Wahrnehmung von Fakten
- ▶ Ignorieren von Unsicherseiten
- ▶ Selbstreferentiell
- ▶ Dichotomisierung
  - ▶ Schwarz/Weiss
  - ▶ "Wir" gegen "Die anderen"
- ▶ Fakten ⇒ alternativlose Handlung

## Wissenschaft

- ▶ Organisierter Zweifel
- ▶ Hinterfragen und revidieren
- ▶ Quantifizieren & prognostizieren
- ▶ Beobachtung, Hypothese, Theorie

## Ideologie

- ▶ Im Besitz der Wahrheit
- ▶ Selektive Wahrnehmung von Fakten
- ▶ Ignorieren von Unsicherseiten
- ▶ Selbstreferentiell
- ▶ Dichotomisierung
  - ▶ Schwarz/Weiss
  - ▶ “Wir” gegen “Die anderen”
- ▶ Fakten ⇒ alternativlose Handlung



## Ideologie

- ▶ Im Besitz der Wahrheit
- ▶ Selektive Wahrnehmung von Fakten
- ▶ Ignorieren von Unsicherseiten
- ▶ Selbstreferentiell
- ▶ Dichotomisierung
  - ▶ Schwarz/Weiss
  - ▶ "Wir" gegen "Die anderen"
- ▶ Fakten ⇒ alternativlose Handlung

### 1 | Was versteht man unter gendergerechter Sprache?

Sprache ist Kommunikationsmittel und Erkenntnisinstrument, wodurch sie unser Denken prägt und wiederum von diesem geprägt wird. Liest man den Satz „Die Professoren beantworten nach den Vorlesungen die Fragen der Studenten“, so werden die meisten wahrscheinlich männliche Professoren und Studenten vor Augen haben. Da andere Geschlechter nicht explizit genannt werden, bleiben sie unsichtbar. Gendergerechte und diskriminierungsfreie Sprache bedeutet, alle Menschen sprachlich sichtbar zu machen und bei Formulierungen gleichermaßen zu nennen.

Eine psychologische Studie zeigt zudem, dass Kinder Berufe, die in einer gendergerechten Sprache dargestellt werden, erreichbar einschätzen als solche, die nur der männlichen Form ausgedrückt werden. Diese beruflichen Stereotype werden sprachlich zementiert und können dazu führen, dass sich gerade Mädchen nicht trauten später einmal solche Berufe – meistens aus dem technischen Bereich – zu ergreifen (vgl. Vervecken/Hannover 2015). Eine gendergerechte Berufsbezeichnung kann somit das Selbstvertrauen für diese Berufe steigern.

Der Begriff „Gender“ kommt aus der Linguistik und bezeichnet im Englischen das grammatische Geschlecht (Genus). In der Wissenschaft entwickelte sich der Begriff zudem zu einer Analysekategorie, um das soziale Geschlecht (*gender*) vom biologischen Geschlecht (*sex*) zu unterscheiden. Da es im Deutschen keine Entsprechung gibt, wurde „Gender“ als Lehnwort in den deutschen Sprachgebrauch übernommen. Damit wird betont, dass Geschlechterrollen keine biologischen Phänomene sind, sondern soziale Zuschreibungen.

Insbesondere durch die Änderung des Personenstandsgesetzes wird eine sprachliche Anpassung an den Gleichberechtigungsgrundgesetz immer dringender. Dies zeigt sich auch durch die zunehmenden Aufforderungen und Anfragen an den Rat für deutsche Rechtschreibung, den staatlichen Stellen Änderungsvorschläge zum Regelwerk vorzulegen. Dieser empfiehlt keine konkrete Schreibweise, definiert jedoch sechs grundlegende Kriterien (vgl. Rat für deutsche Rechtschreibung 2018: 8).

Geschlechtergerechte Texte sollen:

- sachlich korrekt sein
- verständlich und lesbar sein
- vor allem im Hinblick auf die Altersentwicklung der Bevölkerung und die Tendenz in den Medien, Texte in vorlesbarer Form zur Verfügung zu stellen
- Rechtsicherheit und Eindeutigkeit gewährleisten
- übertragbar sein im Hinblick auf deutschsprachige Länder mit mehreren Amts- und MinderheitsSprachen
- für die Lesenden bzw. Hörenden die Möglichkeit zur Konzentration auf die wesentlichen Sachverhalte und Kerninformationen sicherstellen

Auch wenn eine eindeutige und allgemeingültige gendergerechte Schreibweise noch nicht etabliert ist, nehmen sich viele Hochschulen und öffentliche Institutionen des Themas durch einen Leitfaden zur gendergerechten Sprache an. So soll zumindest ein einheitliches Bild innerhalb einer Institution entstehen.

## Wissenschaftliche Grundlagen des Genderleitfadens

## Positivismus

- ▶ Erkenntnisse: Sinnlich wahrnehmbare, überprüfbare Befunde
- ▶ Empirismus

## Kritischer Rationalismus

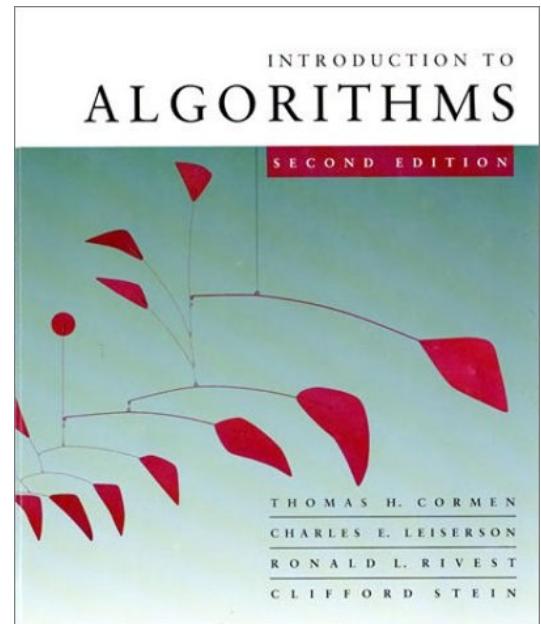
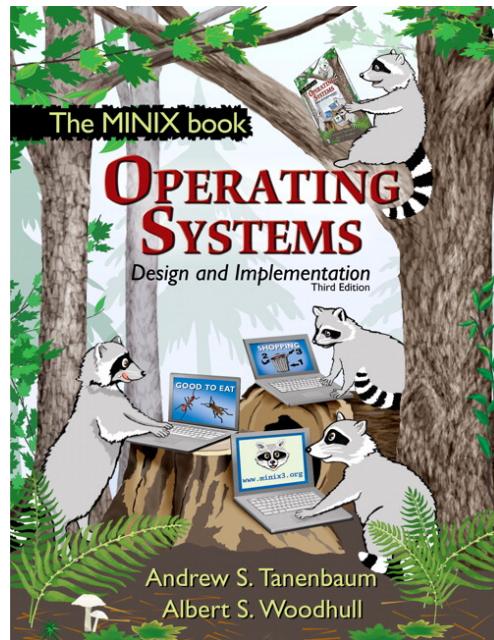
- ▶ Grundsätzliche Unbeweisbarkeit naturwissenschaftlicher Zusammenhänge
- ▶ Erkenntnisse: Falsifizierbar
- ▶ »Ich kann mich irren, du kannst recht haben, und wir zusammen kommen vielleicht der Wahrheit auf die Spur«

# Zum Schluss

# Lerne, indem Du anderen hilfst, sich zu verbessern



# Lerne von den Besten



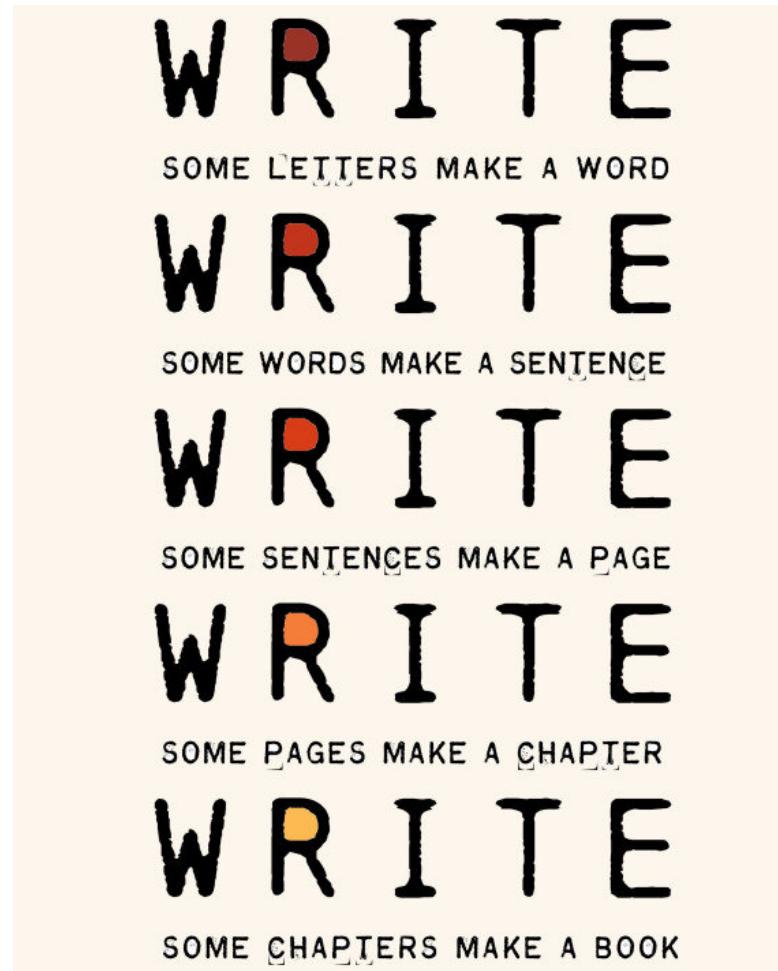
# Lies (und zitiere) lieber Bücher statt Artikel.



A screenshot of a Wikipedia article page for "Datenbank". The page title is "Datenbank". The main content area starts with a definition: "Dieser Artikel beschreibt Datenbanksysteme und Datenbanken aus Sicht der EDV. Im juristischen Sinne gelten auch nicht elektronische Datensammlungen als Datenbank. Mehr zum Thema findet man unter Datenbank (Recht) und Datenbankwerk." Below this, there is a detailed explanation of what a database is, mentioning DBMS, data, and data models like relational databases. The sidebar on the left contains links for "Hauptseite", "Themenportale", "Von A bis Z", "Zufälliger Artikel", "Mitmachen" (with sub-links for improving articles, creating new ones, and contacting the community), "Drucken/exportieren", "Werkzeuge", and "In anderen Sprachen" (with links for Afrikaans, Aragonés, العربية, and مصرى). The footer of the page includes the URL "de.wikipedia.org/wiki/Wikipedia:Hauptseite".



# Schreibe jeden Tag.



# Das eigene Tempo realistisch einschätzen lernen!



Mein Tempo: 1 – 1,5 Seiten mit Inhalt pro Tag (Vollzeit!)

# Die Abschlussarbeit ist ein Vollzeitjob

Können Sie es sich zeitlich *wirklich* leisten...

- kurz vor Abgabe zwei Wochen nach Mallorca zu fahren
- parallel Bewerbungsgespräche abzuhalten
- nebenher zu kellnern
- das neue Haus der Eltern zu renovieren
- sich die Weisheitszähne entfernen zu lassen
- eine türkische Hochzeit zu organisieren
- ...

→ Es wird noch genügend von alleine dazwischen kommen (!)

...based on true stories...

# Erfolgreiche Studis

- Lesen sich breit in das Thema ein
- Lesen über die Aufgabenstellung hinaus
- Sind enthusiastisch
- Sind hartnäckig und geben nicht zu schnell auf
- Übernehmen die Initiative
- Übernehmen die Verantwortung für Ihr Projekt
- Sind gut organisiert
- Sind bestrebt, sich zu verbessern
- **Das Ergebnis sieht nach einer wiss. Arbeit aus**



Aha!

Hier steht nix von "Genie"...