HEINRICH-HEINE-UNIVERSITÄT DÜSSELDORF
INSTITUT FÜR INFORMATIK
LEHRSTUHL FÜR RECHNERNETZE
PROFESSOR DR. M. MAUVE

Bachelor-Leitfaden

1 Grundlegende Informationen

Dieses Dokument enthält generelle Empfehlungen für Bachelorarbeiten am Lehrstuhl für Rechnernetze, Heinrich-Heine-Universität Düsseldorf.

Dauer

- 2 Wochen Einarbeitung vor der endgültigen Anmeldung
- 3 Monate Bearbeitungszeit

Anspruch an eine Bachelorarbeit

- Prinzipielle Lösungsidee wird vorgegeben
- Selbstständiges Erarbeiten der Lösung

Voraussetzungen

- Wissenschaftliche Neugier
- Programmierkenntnisse UND
- Freude am Programmieren
- Schwerpunkt: Modul Rechnernetze oder Mobilkommunikation

Achtung

- Wir vergeben keine leichten Bachelorarbeiten. Man kann durchfallen, wenn man keine ausreichende Leistung erbringt!

2 Format

2.1 Schriftliche Ausarbeitung

Die Arbeit ist als LaTeX-Dokument zu Lehrstuhl stellt eine entsprechende Formatvorlage zur Verfügung. Das Literaturverzeichnis ist mittels BibTeX zu erstellen. Der Lehrstuhl stellt eine BibTeX-Datei mit grundlegenden Referenzen zur Verfügung. Zur Arbeit mit LaTeX gibt es viele gute Nachschlagewerke und Online-Quelle. Beispielhaft ist hier

`https://latex.tugraz.at/latex/tutorial`

genannt.

Umfang der Arbeit: 25–30 Seiten reiner Text; der Anhang (Literaturverzeichnis), das Deckblatt und große Grafiken zählen nicht dazu.

Einige Grundregeln für (LaTeX-) Dokumente

- Zahlen von eins bis einschließlich zwölf werden ausgeschrieben.
- Die Wörter *Section*, *Table* und *Figure* werden groß geschrieben, wenn es sich um Referenzen auf Objekte im eigenen Text handelt (auf deutsch "Abschnitt", "Tabelle" und "Abbildung").
- Überschriften werden groß geschrieben.
- Fußnoten fangen mit einem großen Buchstaben an und enden mit einem Punkt.
- Für Grafiken ist es grundsätzlich immer das Sinnvollste, Vektorgrafikformate (svg, eps, pdf) zu verwenden. Falls der Unterschied zu Rastergrafiken nicht klar ist, kann dies unter

  `https://de.wikipedia.org/wiki/Vektorgrafik`

  nachgelesen werden. LaTeX selbst kann nur `eps` und `pdf` Grafiken einbinden, das vom Lehrstuhl zur Verfügung gestellten Rakefile kann eine Vielzahl von Grafikformate bei korrekter Benutzung automatisch in diese beiden Formate konvertieren (im Zweifelsfall: Betreuer fragen).
- Jede Tabelle und jede Grafik ist im Text zu referenzieren und ausreichend zu erläutern. "Ausreichend" bedeutet in der Regel, dass der Text die Grafik erklärt und die Aussagen der Grafik darstellt. Eine solche Referenz wird im LaTeX Code als

  ```
  Abbildung~\ref{Tabelle:zahlen_eins}
  ```

  angelegt. Wichtig ist hier u.A. das "~", es sorgt dafür das der Bezeichner für die Tabelle am Zeilenende nicht umgebrochen wird.
- Alle Probleme mit dem Textsatz, Layout und dem Literaturverzeichnis sollten **frühzeitig** behoben werden. Wird LaTeX richtig benutzt, treten die sonst typischen Formatierungsprobleme anderer Texteditoren gar nicht erst auf, nutzen Sie das! Die Nacht der Abgabe ist für das Beheben dieser Probleme **absolut** ungeeignet!

2.2 Quellcodes

Es sind sämtliche für die Bachelorarbeit geschriebenen Quellcodes abzugeben. Diese sollen ausreichend und durchgehend entweder in Deutsch oder Englisch kommentiert sein.

Am Lehrstuhl werden in der Regeln nur Abschlussarbeiten vergeben die einen gewissen bis großen Programmieranteil haben. Naturgemäß ist deswegen neben der reinen Implementierungsqualität (etwa Bugfreiheit) auch die ausreichende Kommentierung des produzierten Quellcodes wichtig und wird bei der Notengebung berücksichtigt. Als kleine Randbemerkung: auch übermäßiges Kommentieren kann das Verstehen des Sourcecodes erschweren, es ist also auf die richtige Menge an Kommentierung zu achten.
Bei der Abgabe der Arbeit auf CD/DVD bietet es sich an die Sourcen entsprechend ihrer logischen Struktur innerhalb der Arbeit z.B. in verschiedenen Verzeichnissen unterzubringen. Außerdem ist hier unter Umständen auch eine kurze README.txt pro Verzeichnis sinnvoll die kurz die der gelieferten Software zugrundeliegenden Prinzipien oder das Zusammenspiel der einzelnen Komponenten erläutert.

3 Zeitlicher Ablauf

Für den Ablauf der Bachelorarbeit ist die Prüfungsordnung, zu finden unter
`https://cs.hhu.de/studium-lehre-informatik/studierende/`
maßgeblich. Die im Folgenden aufgelisteten Punkte sind nur Empfehlungen des Lehrstuhls.

3.1 Beginn (t=0)

Der Student kommt beim Lehrstuhl vorbei und erhält nach gemeinsamer Absprache einen groben Themenvorschlag und die nötigen Referenzen, um sich einzuarbeiten.

3.2 t=2 Wochen

1. Der Student meldet das Thema formal an oder lehnt die Bearbeitung ab.
2. Interaktion mit dem Betreuer (Mitarbeiter des Lehrstuhls).
3. Beginn der Bearbeitung (Implementieren, Experimentieren, Entwurf der Ausarbeitung).

3.3 Zwischenabgaben / Fortschrittskontrolle (t=6, t=10)

Etwa 4–6 Wochen nach der Anmeldung des Themas muss eine erste Rohversion der Arbeit dem Betreuer vorgelegt werden. Es sollten hier mindestens vorhanden sein

- Die Gliederung
- Die Einleitung und
- Vom Betreuer vorgegebene Kapitel.

3.4 Entwurf der Bachelorarbeit (t=12)

Die gesamte Bachelorarbeit sollte zu diesem Zeitpunkte inhaltlich fertiggestellt sein und dem Betreuer vorgelegt werden.

3.5 Eine Woche vor Schluss

. . . sollten keine inhaltlichen Änderungen mehr nötig sein, d.h. die letzte Woche der Arbeit sollte für den sprachlichen und optischen Feinschliff genutzt werden. Lassen Sie die Arbeit auch von einem “Unbeteiligten” lesen und korrigieren.

3.6 t=3 Monate und zwei Wochen: Finale Abgabe

Abzugeben sind neben der gebundenen Arbeit auch eine CD, die die Arbeit als PDF sowie die gesamten zugrundeliegenden LaTeX-Quellen und alle weiteren Quellen beinhaltet, sodass die Arbeit aus diesen Quellen neu erzeugt werden kann. Des Weiteren muss die CD alle Quellcodes beinhalten, die im Laufe der Arbeit erstellt worden sind. Die CD muss weiterhin eine Textdatei enthalten, in der anzugeben ist, welche Dateien sich wo auf der CD befinden.

4 Ressourcen

Folgende Ressourcen werden von Seiten des Lehrstuhl zur Verfügung gestellt:

- Aufgabenstellung
- Formatvorlage für die Arbeit (LaTeX)
- BibTeX-Dateien mit einem Teil der relevanten Literatur

5 Abschlussvortrag

- Der Abschlussvortrag (für die Bachelorarbeit) hat eine maximale Dauer von 20 Minuten, hierin enthalten ist auch eine evtl. Demo-Vorführung
- Es gilt die Empfehlung: für jede Folie (mit Inhalt) benötigt man in etwa zwei Minuten zum erklären
- Wenn Sie deutlich mehr als 10 Folien in ihrem Vortrag haben: halten Sie Rücksprache mit ihrem Betreuer!
- Üben Sie den Vortrag mehrmals, stoppen Sie die Zeit; wenn der Vortrag schon beim Üben deutlich über 20 Minuten dauert: überdenken Sie das Konzept, kürzen Sie den Vortrag.

6 Sonstiges

6.1 Termine

Es empfiehlt sich, Termine mit dem Betreuer vorher per E-Mail zu vereinbaren. Der Betreuer

- ist dann physikalisch anwesend und
- hat Zeit sich vorzubereiten.

6.2 Zufallszahlen

Es gilt der Grundsatz: **Random numbers are evil!**
Einige Beispiele aus den Erfahrungen des Lehrstuhls mit Abschlussarbeiten und Zufallszahlen:

- Für die Implementierung einer ns-2 Simulation wurden bestimmte Werte mit echten Zufallszahlen initialisiert. Im Nachhinein war es dadurch nicht mehr möglich die Läufe dieser Simulation zu reproduzieren.
- Die Paket-IDs in einem Experiment wurden von jedem Knoten zufällig gewählt. Die Wahrscheinlichkeit das zwei Knoten dieselbe ID wählen ist je nach Zahl der Pakete sehr hoch (das sogenannte **Geburtstagsproblem/paradoxon**: bei 23 Personen ist die Wahrscheinlichkeit das zwei dieser Personen am selben Tag Geburtstag haben > 50%). Da die Pakete nicht unterschieden werden konnten waren die Messungen unbrauchbar.

Fazit: Sobald an einer Stelle Zufallszahlen zum Einsatz kommen sollen, muss dies gut bedacht werden! Es empfiehlt sich eine kurze Rücksprache mit dem Betreuer.

7 Versionsverwaltung

Zur Verwaltung von Programmcode und den Quelldateien der schriftlichen Ausarbeitung empfiehlt es sich ein Versionsverwaltungssystem einzusetzen. Am Lehrstuhl wird `git` verwendet. Zur Verwaltung der einzelnen Projekte nutzen wir unser GitLab unter

`https://gitlab.cs.uni-duesseldorf.de`

Zugang dazu gibt es über den betreuenden Mitarbeiter bei unserem Administrator.

7.1 git

Um sich mit der Versionsverwaltung `git` vertraut zu machen gibt es unter

`https://try.github.io/`

ein kurzes Tutorial. Best Practices für `git` sind unter

`https://indico.cern.ch/event/288437/contributions/1642636/attachments/538025/741708/git-best-practices.pdf`

zu finden.

7.2 GitLab

Zum Umgang mit `git` und GitLab gibt es unter

`https://docs.gitlab.com/ce/gitlab-basics/README.html`

die *GitLab Basics*.

Anhang

Abgeleitet von `https://juffalow.com/other/write-good-git-commit-message`.

A Write good git commit message

For a long time I didn't know that even writing commit message can have its own "best practices". When I got in touch with git for the first time, this part was described with words like "…and here you can write something short about what's going on in the commit".

Bad commit messages Look at the commit messages below. If you want to merge them, you really don't know, what you are adding / changing, what they do or why you need them. The same applies if you want to search something in the history. You scroll down the log, but it is a mess and a waste of time.

```
cd3e27a contact page
aee9d0d comments
eac95e5 list of online users, some other changes because of server
fae5636 little edit
```

Good commit messages And now take a look at these messages. Better? I think so.

```
43ec6aa Fix error when the URL is not reachable
4fe84ab Add error message if something went wrong
753aa05 Add server fingerprint check
df3a662 Fix shadow box closing problem
```

A.1 How to write a good commit message

The whole message should have its format — subject, body and optionally conclusion consisting of resolved / closed issues.

Subject The git commit help page described it very good as *a single short (less than 50 character) line summarizing the change, followed by a blank line*. The subject should start with a capital letter and should not end with dot. And the important thing here is, it has to be in imperative form. Chris Beams wrote a simple rule to get it right every time:

> A properly formed Git commit subject line should always be able to complete the following sentence: if applied, this commit will *your subject line here*.

For example:

- if applied, this commit will *Delete unnecessary files*
- if applied, this commit will *Add grep option*
- if applied, this commit will *Fix error when protocol is missing*

It will not work for bad commit messages:

- if applied, this commit will *contact page*
- if applied, this commit will *list of online users, some other changes because of server*

The git itself is using this approach. When you merge something it generates a commit message like “Merge branch...” or when reverting “Revert...”.

Body Here you write what and why is changed. The body should not exceed 72 characters for a line. Of course not every commit has to have body.

Bottom line On the end, you can add which issue does the commit fix or is related to. This can be a link, number or if you use GitHub you can write it as *Resolves #N* / *Closes #N*, where *N* is the issue ID.

Example This is an example commit from one of my repositories:

```
Fix error when protocol is missing

First, it checks if the protocol is set.
If not, it changes the url and add the
basic http protocol on the beginning.
Second, it does a "preflight" request
and follows all redirects and returns
the last URL.
The process then continues with this URL.

Resolves #17
```

A.2 Conclusion

Thank you for reading, I hope you learned something new. If you have another tip(s) how to write better commit messages, or how to better use this tool, please leave a comment.

By Maria Luisa Carrion D. on Medium

General Coding Guidelines: Clean Code From Day 1

Help your teammates understand, reuse and extend your code easily, and help them not to introduce bugs while they are at it.

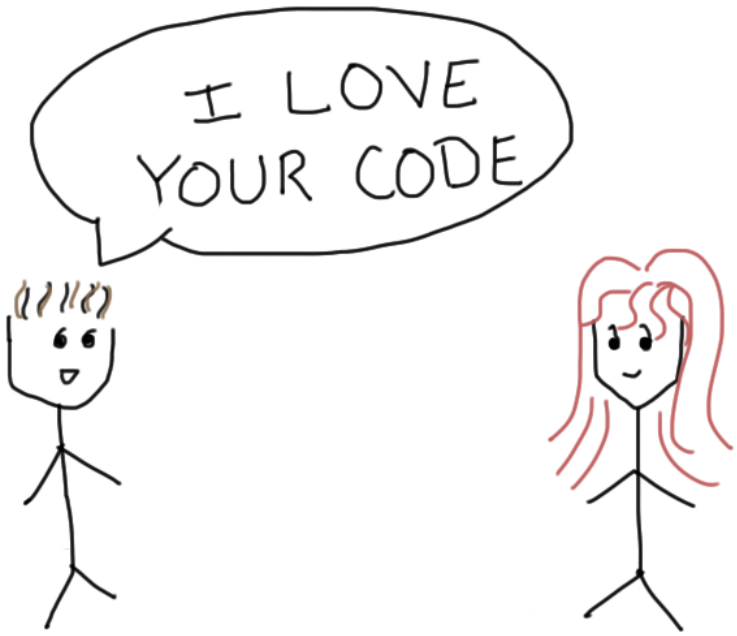

Abbildung 1: Make people happy with your code

Good coding habits make working with your team a piece of cake

The life of a programmer is beautiful when the code they work with is beautiful. Through your code, you have the power to affect the life of your fellow programmers:

- They will **rejoice/panic** when they know they have to work with the code you wrote.
- They will spend **10 minutes/1 hour** understanding the code you wrote.
- They will introduce **0 bugs/2 bugs** when they edit the code you wrote.

I have been there, in all of those scenarios. I know that every programmer focuses on building the functionality they want, but not all of them pay attention to how they build it. They tend to forget that, eventually, someone else will have to read that code.
It seems silly that I took the time to write this article when there are tons of articles and books on the subject, but the sad reality is that I have seen bad coding habits in all kinds of environments:

Abbildung 2: Every time you code like this you make a programmer cry

as a student in a master's degree, as a programmer in a big company, as a programmer in a startup and as a freelancer working with other freelancers.
If you want people to rejoice with the idea of working with your code, just focus on two things: make your code readable and make your code robust.

Make Your Code Readable

Your teammates and your future self should be able to understand, reuse and extend your code seamlessly whenever they need to.

1. Use Meaningful Names and Avoid Made-Up Acronyms

```
// Avoid this
label1.text = dog.name
label2.text = dog.breed
label3.text = dog.int_temp
label4.text = dog.breed.int_fact
```

How will your teammates know if the values are being assigned correctly if the labels don't give any hint about the information they are supposed to show? and how will your teammates know what do `int_temp` and `int_fact` mean exactly?

```
// Do this
name_label.text = dog.name
breed_label.text = dog.breed
internal_temperature_label.text = dog.internal_temperature
interesting_fact_label.text = dog.breed.interesting_fact
```

Your teammates will appreciate it if you save them the time of looking through the code to know what something means or what something is used for. To make sure that the name of a variable, class or function is meaningful enough, you can ask yourself:

> If I see this name alone in the source code, will I know what it is for without looking further into the code?

True story: I have seen JavaScript code where references to UI controls were called **x**, **y** and **z**, so I had to look extensively through the code to determine what controls **x**, **y** and **z** were referring to.

2. Follow the Conventions

I cannot tell you exactly how to name your variables, functions and other stuff because I don't know what programming language you are using. Each language, and sometimes each project, has a specific coding convention.

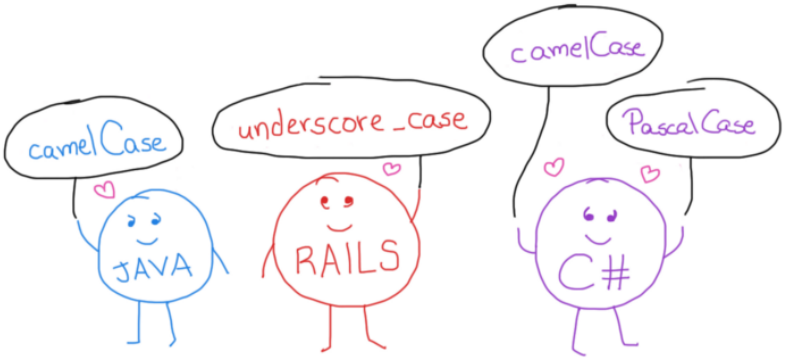

Abbildung 3: Each language has a specific coding style

Every time you learn a new language, one of the first things you should do is read its corresponding coding or style guidelines, so that you can do things the right way from day 1. If you wait too long to learn the right way to code in your specific language, you'll start developing some bad habits.
Note that if you are working on a project and the project's team has defined a coding convention that is different from what you expect, you should follow that convention. The important thing here is follow the convention of the project or platform you are working on and make your fellow programmers happy.

3. Keep Functions Short and to the Point

How short? around 20 lines of code or less. And yes, nobody likes to count lines of code while coding. Thankfully, you don't have to do that, you can use the old recommendation: *the function should fit entirely on your screen so that you don't have to scroll to read it completely*.
Both of us know that nowadays some screens are HUGE. So you can adjust that recommendation and make the functions fit half of your screen, 1/3 of your screen, or whatever portion of your screen works for you.
Keeping the function short isn't enough. You have to keep it doing what the name PROMISED it would do. If the function says Annie will eat an apple, then when the function is executed, Annie should be eating an apple and not oranges, grapes or strawberries. Keep your functions short and to the point, so that your teammates don't have to keep a mental map of everything the function does because its name doesn't match its body.
Update: As the reddit user lykwydchykyn pointed out in this comment, it's dangerous to blindly stick to a given amount of lines when writing a function. Beginners might be tempted to write "hideously dense, long lines of code to keep functions under a line count limit", or "bucket brigade functions, where a single procedure is just broken up arbitrarily into functions that call one another (i.e. `step1()` gets to line 19 and then calls `step2()`, which gets to line 19 and calls `step3()`. No individual step makes any sense in isolation.)". So, beginners should keep in mind that it is more important to keep your functions meaningful than sticking to a given line

Abbildung 4: If your function occupies a huge screen entirely, it doesn't count as a short function

count. Also, even when programming languages allow you to place two or more statements in the same line, it is better to write each statement in its own line. And finally, IDEs display a right margin that indicates the recommended line length. You can use that right margin to know if your line is growing too much.

4. Update Member Variables Explicitly

Other guides explain this more broadly by saying:

> Functions should not have side effects.

Updating a member variable of a class IS a side effect of a function if the function doesn't indicate that it will be updating that variable.
This part is tricky because we usually update the private members of a class inside its methods when the class represents a business domain entity. For example, `car.move` will update the car's position variable and that's OK because `Car` represents a business domain entity. That's not the kind of classes I'm talking about here.
Think about the classes used for an Android `Activity` or an iOS `UIViewController`. In those classes we have lots of members variables that hold all kinds of information. We may initialize some of those variables inside functions that do some other stuff. That is BAD.
What I prefer to do in this situations, is make a function that returns a value and assign that value explicitly to the member variable. When my teammates see the function call being assigned to the variable, they automatically know that the function is being used to set that variable, and they don't have to look into the function to know if it is changing the value of the variable or not.

5. Use Comments Wisely

Your comments should not explain step by step what the code is doing. Your excellent variable/method/class naming skills will take care of that.

Your comments should be cleverly placed to say WHY something has been coded in a certain way. Jeff Atwood explains this wonderfully in his article Code Tells You How, Comments Tell You Why.
If you find yourself thinking:

> I implemented this in this way because of that important technical decision.

Then that's some code that needs to be commented. If you don't add the comment, your teammates might introduce a bug when they edit the code because they are not aware of the important technical decision.
Finally, this is what your teammates actually think when they see commented code:

> Oh, some comments, but I understand what the code does, I don't need to read them. I'll just change this little piece of code.

And BOOM, the comments are outdated. So this is another reason to avoid not needed comments and only adding comments that you actually want people to read.
True story: I have seen a comment next to a magic number indicating what the number means. A comment next to a magic number doesn't make the comment a good comment.

Make Your Code Robust

Your teammates and your future self should be able to edit your code without introducing bugs in the process.

1. Avoid Duplication

The famous DRY (Don't Repeat Yourself).
This is harder to achieve than you might think. When you are under pressure, you start working on your feature right away without checking the existing code to see what functions already exists. By the time you finish writing your code, you realize that you inadvertently wrote a function very similar to one that already existed. But now you are finished, you are tired and want to go to sleep. It seems like a waste to invest your time in extracting the common behavior into a separate function. After all, the similarity is so small... Why bother?

```
// Written by Tommy two weeks ago
setHourlyAlarm() {
    /**
     * Very cool code for setting an alarm    <- DUPLICATED
     */
    /**
     * Very cool code to do it hourly
     */
}

// Written by you just now because you didn't see Tommy's code
setDailyAlarm() {
    /**
     * Very cool code for setting an alarm    <- DUPLICATED
     */
```

```
    /**
     * Very cool code to do it daily
     */
}
```

If you leave that seemingly innocuous piece of code duplicated, doom will reach you. Because, one day, the code for setting an alarm will change and someone will change it in just one of the functions if the other function is too far away from the first one to see it. And no one will notice that there is one function still using that old, deprecated, and maybe buggy piece of code.
Do yourself a favor and give a quick view to the existing code BEFORE you start working. You can see if there is something you can reuse to make your work more efficient.
True story: I have seen two functions that do similar things one below the other, with some common behavior that could have been refactored into a third function.

2. Avoid Hard-Coded Strings and Magic Numbers

Every string should be a reference to a string resource, an enum or a constant. Why? because you think that you'll only use that string this one time. What you don't know is that there will come a fateful day when that string is used in another place too. And after that, will come an even worse day when someone edits that string, and they only edit it in one place because they have no idea it exists somewhere else. Only very careful people do a "Find in project" for every hard-coded string they update.

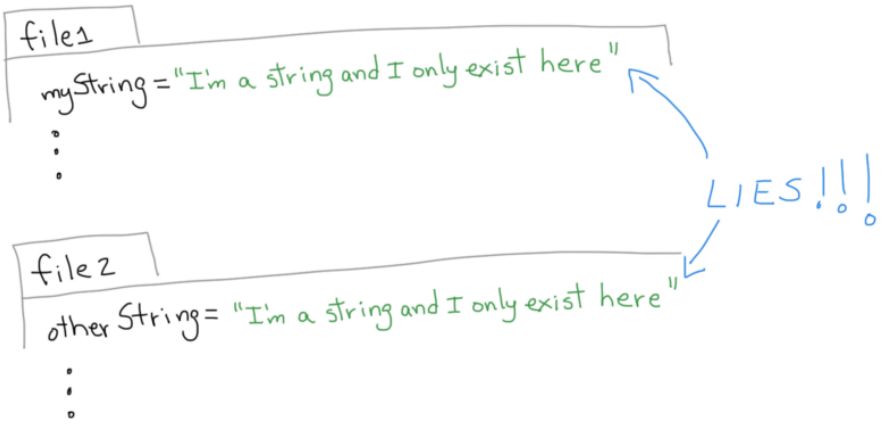

Abbildung 5: Hard-coded strings are death bringers

Avoid your teammates a headache and give them a single place to update a string that is used throughout the code: a good old constant (or enum, or string resource).
The same thing goes with magic numbers. They may be even worse than hard-coded strings, because you have no idea of what that number represents. Take a look at the following example.

```
// Avoid this
switch(status) {
    case 1:
        do_this();
        break;
    case 2:
        do_that();
```

```
            break;
        default:
            do_something_else();
}
```

How do you know what the system is supposed to do in status 1 or status 2 if you have no idea of what those numbers represent? Keep your code auto-documented by placing numbers in a constant or an enum.

```
// Do this
switch(status) {
    case STATUS_ENABLED:
        do_this();
        break;
    case STATUS_DISABLED:
        do_that();
        break;
    default:
        do_something_else();
}
```

True story: I have seen Android preferences keys hard-coded to read and write values to the shared preferences.
True story: I have seen code where a plain number is assigned to a variable called 'status' and next to the number, a comment indicating what the number means.

3. Check Values From Unreliable Sources

Or, in the words of my friend Gabriel Aguirre:

> Don't trust the server.

You have a server, and it has promised you to return a JSON objects with all the values you want: the name and breed of the dog. You trust the server and you use those values directly in your code:

```
// Avoid this
// This code throws a null exception
// if the JSON doesn't have the 'name' or 'breed' properties
dogName = dogDataJSON.get("name").getAsString();
dogBreed = dogDataJSON.get("breed").getAsString();
```

If the server doesn't keep its promise, your app irreparably crashes. But it doesn't end there, it's not just the server whom you can't trust. You can't trust data coming from ANY unreliable resource. Something could have gone wrong with the generation of the information you expect and you might find an unwanted null value.

```
// Do this
// Do a null check before working with data obtained from the server
dogName = dogDataJSON.get("name") != null?
            dogDataJSON.get("name").getAsString() : "";
dogBreed = dogDataJSON.get("breed") != null?
            dogDataJSON.get("breed").getAsString() : "";
```

True story: I've had this error twice with SOAP objects and with JSON objects. I've learned my lesson.

4. Always Write the Braces Around Single Line Blocks

We all know that when you have an if statement with just one line, you can omit the braces. Please don't do this. This is extremely error prone, and I'll tell you why:
Your code will work perfectly fine until a teammate is working on a feature that needs to add just one more statement to that `if` block you wrote. Your teammate comes to your `if`, adds the statement and then keeps working on the feature. Guess what, your teammate forgot to add the braces because no one pays attention to the braces when they have other things in their minds. The `if` statement still has one statement and now there is a random orphan statement executing without anyone noticing. When the software breaks, your teammate will spend precious time trying to find a bug that apparently came out from nowhere.

Abbildung 6: Single-line blocks without braces are the true death bringers

Make life easier for those who will come after you and add the braces in your single-line `if`, `else`, `while`, `for` and any other block that allows optional braces.
True story: I have seen this kind of bugs happen in the technology departments of big companies. Even experienced programmers forget to add the braces.

Write Everything Code-Related in English

This may seem obvious for native English speakers, but I have seen code from Spanish speaking programmers with comments, variable names and function names written in Spanish. This is not a good practice.
Why is it better to write code-related stuff in English?

- **Names in English are shorter**, therefore your variable names and function names are shorter and easier to read if you name them in English.
- It's **easier to ask a question in Stack Overflow** and copy/paste your code without translating the comments or variable/function names for people around the world to answer your question.
- It allows you to seamlessly **add a new team member from another country** to your project, because every programmer knows English.
- **Getters and setters**: it's weird to have getters and setters that don't start with **get** and **set**. So some people mix the **get/set** word with the Spanish word for the object they are **getting/setting**. That's kind of weird.

So what you should write in English? Everything: classes, variables, functions, comments, log messages and even commit messages so that team members from other countries can review the repository without problems.
One problem I have faced with this practice is:

> I only know the business domain language in my native language because that's how my customer explained things to me. Naming things in my native language is so much easier. What should I do?

FEAR NOT! Google Translate is your best friend. Just translate the terms and use the English equivalent in your app. I have done this several times and I have never regretted it, but I have regretted not using English as the single language in my code base.

Final Words

I know it's difficult to apply all these guidelines while you are coding because you are focused on the functionality you are writing and you probably are short on time and that's OK. But once you finish writing a piece of functionality or you are ready to push some code... **STOP!** and read the code. Read it and and you'll realize if your teammates will be able to understand your code easily or if they'll have a hard time modifying it.
Every IDE nowadays allows you to refactor easily, so it will only take a couple of minutes to extract that piece of functionality into another function, rename some variables and add some comments. Your teammates and your future self will appreciate it.
Many books and articles have been written about this subject. One I particularly like is Clean Code: A Handbook of Agile Software Craftsmanship.
Thank you for reading and enjoy your programmer life.

Kommen nach den “science wars” die “reference wars”?

Stefan Weber 29.09.2005
Wandel der Wissenskultur durch Netzplagiate und das Google-Wikipedia-Monopol
Wer eben mal schnell wissen will, wann sich Habermas habilitierte oder was nun Gotthard Günther mit “Polykontexturallogik” genau meinte, der befragt Google oder gleich die Wikipedia. Die Vorteile sind unübersehbar: Es müssen keine dicken Bände durchforstet werden, der Gang in die Bibliothek und zum vergilbten Zettelkasten ist nicht mehr notwendig. Mittlerweile sind Google-Ergebnisse und Wikipedia-Beiträge zu Wissensautoritäten, zu Wissensmonopolen neuer Art geworden: Veröffentlicht und öffentlich zugänglich erscheint oft nur noch, was von Google gefunden wird und/oder in die Wikipedia aufgenommen wurde.
Wir ertappen uns dabei, einem Beitrag in der Wikipedia oftmals bereits mehr Vertrauen zu schenken als einem Eintrag in einem gedruckten Handbuch oder Lexikon. Zumal wenn letzterer auch noch namentlich gekennzeichnet ist, lesen wir den Print-Eintrag oft stärker unter subjektiven Prämissen als die ungemein objektiv und depersonalisiert wirkenden Wikipedia-Einträge. Bei diesen, so glauben wir, können wir uns in der Regel auf das selbstreinigende Potenzial eines kollaborativen Hypertextes verlassen: Falsche Angaben, ja selbst kleinere Unschärfen werden früher oder später erkannt und korrigiert werden. Bei Lexika oder Handbüchern kann das hingegen oft Jahre dauern.

Kollaboratives Umschreiben – von welchem Ausgangstext eigentlich?

Der bequeme und enorm zeitsparende Weg zu Google und Wikipedia verstellt uns oft die Sicht auf ein Problem, das bislang noch kaum diskutiert wurde: In Frage steht nicht, wie einmal in die Wikipedia eingespeiste Texte durch kollaboratives Arbeiten am Text korrigiert werden, problematisiert werden sollte vielmehr, wie der Ursprungstext eines Wikipedia-Beitrags eigentlich zustande gekommen ist.
Sehen wir uns dazu zwei Beispiele an: Der Beitrag Journalismustheorien (1) etwa strahlt auf den ersten Blick hohe wissenschaftliche Seriosität aus: Alles scheint korrekt zitiert zu sein, und der Bogen spannt sich recht umfassend von den frühen personalistisch-idealisierenden Ansätzen bis zu heute vertretenen Theorien wie Strukturierung, Cultural Studies, Konstruktivismus und Systemtheorie.
Doch die Einteilung der Journalismustheorien in “normativen Individualismus”, “legitimistischen Empirismus” oder “integrative Sozialtheorien” macht den Kenner stutzig: sie ist keine systematisierende Eigenleistung des Wikipedia-Erstautors. Die Systematik stammt vielmehr vom an der TU Ilmenau lehrenden Medienwissenschaftler Martin Löffelholz. Bei näherem Hinsehen zeigt sich, dass ein Großteil des Wikipedia-Textes eine Paraphrase (wenn man es gut meint) oder ein Plagiat (wenn man es nicht beschönigend aussprechen will) einer Arbeit von Martin Löffelholz darstellt, die 2003 im Handbuch “Öffentliche Kommunikation” (Westdeutscher Verlag) publiziert wurde. So weit, so gut, werden Sie sagen, und wo liegt das Problem?

Verstoßen Wikipedia-Texte permanent gegen das Urheberrecht?

Es geht hier gar nicht primär um den Schutz des Copyrights des Originalautors bzw. des Verlags, obwohl jeder Wikipedia-Autor die folgenden beiden Hinweise vor dem Verfassen eines Textes eigentlich gelesen haben müsste:
–Du versicherst hiermit, dass du den Text selbst verfasst hast [...].
Das Kopieren urheberrechtlich geschützter Werke ohne Erlaubnis des Autors ist verboten. [Fett im Original]–

Der Wikipedia-Texter könnte nun antworten: Er habe die Paraphrase (oder eben das Plagiat!) der Arbeit von Martin Löffelholz ja selbst verfasst. Deshalb handle es sich um keine Kopie, weshalb der Autor auch nicht um Erlaubnis gefragt werden müsse (genau derartige Missverständnisse bilden den Kern eines meines Erachtens neuartigen Problems mit der "Referenz-Kultur"). Angemerkt werden sollte an dieser Stelle: Der Autor Martin Löffelholz wurde nicht gefragt - vielmehr war er erstaunt, eines Tages große Teile einer seiner Arbeiten in der Wikipedia zu finden: "Zufällig bin ich auf den Wikipedia-Artikel über Journalismustheorien gestoßen. Ganze Passagen des Artikels ähneln - teilweise sogar wörtlich - Beiträgen, die ich an anderer Stelle veröffentlicht habe", berichtet Löffelholz gegenüber Telepolis. Sein Urteil ist klar: "Dem Zitationsgebot kommt der Wikipedia-Artikel nicht hinreichend nach."
Das komplexere Problem liegt jedoch nicht in der Copyright-Frage, sondern ganz wo anders: Studierende, die nun den Eintrag "Journalismustheorien" lesen, werden - nicht zuletzt auf Grund der Wissensautorität, die der Wikipedia zugeschrieben wird - davon ausgehen, dass hier korrekt vorgegangen wurde: Sie bemerken ja in der Regel nicht, dass ganze Absätze nahezu 1:1 übernommen wurden, und der lapidare Vermerk am Ende eines jeden Absatzes - wie etwa "(nach Löffelholz 2003, S. 36-37)" - könnte ihnen als korrekte Zitierweise erscheinen. In jeder guten Einführung in das wissenschaftliche Arbeiten lernt man jedoch: Das ist ganz und gar nicht korrekt, weil dann für den Leser nicht mehr klar ist, was nun von wem stammt.
Ein weiteres Problem: Plagiate oder auch nur Paraphrasen auf der Wikipedia nötigen den Plagiierten, selbst Änderungen am Text vorzunehmen. Nun kann aber nicht davon ausgegangen werden, dass ein Wissenschaftler große Lust (oder auch genug Zeit) hat, an einer Paraphrase seines eigenen Textes herumzuarbeiten. Martin Löffelholz dazu: "Wenn ich mich als wissenschaftlicher Autor ständig mit Plagiaten auseinandersetzen müsste, wäre dies schon ein erheblicher Nachteil." Das heißt: Wer einmal ein Plagiat in die Wikipedia gestellt hat, der kann damit rechnen, dass dieses entweder ganz gelöscht wird oder aber das 'Rumpf-Plagiat' im Wesentlichen als Basis für weitere Korrekturen erhalten bleibt.

Netzplagiate: Zwischen 15 und 50 Prozent der Arbeiten?

Ein anderes Beispiel: Auch der Eintrag Theorie der Fotografie (2) stammt nicht nur von jenen Autorinnen und Autoren, deren Nicknames unter "Versionen/Autoren" aufgelistet sind. Zwei Absätze wurden nahezu wortwörtlich abgeschrieben von einem Text des Autors dieses Telepolis-Beitrags. Ich hatte ihn 1999 in dem Suhrkamp-Band "Konstruktivismus in der Medien- und Kommunikationswissenschaft" publiziert. Zitate von Arthur Kroker oder Siegfried J. Schmidt, die ich - man muss es betonen: als genuine Eigenleistung - herausgesucht habe, finden sich sinnentstellend wieder in einem Eintrag zur "Theorie der Fotografie" - zu einem Thema, mit dem mein Originalaufsatz eigentlich gar nichts zu tun hatte. Wenn an einer Stelle so gearbeitet wurde, wie ist dann der Rest des Eintrags zustande gekommen?
Bloße Haarspalterei, Erbsenzählerei eines überakribischen Medienwissenschaftlers oder doch Indizien für einen sich abzeichnenden fundamentalen Wandel der Autorenreferenz? Nach einer Häufung weiterer Hinweise optiere ich für Letzteres:
Bereits im November 2004 behauptete die Hamburger Medienkulturwissenschaftlerin Joan Bleicher in einer dpa-Meldung: "Bis zu 15 Prozent der angehenden Akademiker in Deutschland laden sich [...] wissenschaftliche Artikel aus dem Internet herunter, um damit ungekennzeichnet ihre eigenen Arbeiten aufzuwerten." Gegenüber Telepolis berichtet Bleicher heute, sie habe in Reaktion auf diese Meldung zahlreiche E-Mails erhalten, deren einhelliger Tenor überraschte: Bleichers Schätzwert sei zu niedrig, einige Wissenschaftler-Kollegen gingen von bis zu 50 Prozent zumindest teilweise plagiierten Arbeiten aus. Besonders beliebte Tricks wurden erwähnt: Die Einleitung und das Schlusskapitel werden selbst geschrieben, der Rest verdankt

sich dann der Copy, Shake & Paste-Methode. Und längst wird nicht mehr nur aus online verfügbaren Texten gesampelt: Immer öfter kommt es auch vor, dass Kapitel von bereits existierenden Seminar- oder Diplomarbeiten eingescannt und diese dann mittels Optical-Character-Recognition-(OCR-)Software in einen weiterbearbeitbaren Text umgewandelt werden: Der vermeintliche wissenschaftliche Texter wird somit vollends zum versierten Computertechniker, zum Software-Trickser.
Seit Mitte Mai 2005 werden am Geschichte-Institut der Universität Wien alle Diplomarbeiten und Dissertationen mittels Plagiatserkennungssoftware überprüft. Fünf von 30 Arbeiten waren trotz eines zuvor bekannt gewordenen Präzedenzfalls zumindest zum Teil plagiiert, berichtet das österreichische Nachrichtenmagazin "profil" ("profil extra uni" vom 12. September 2005).
Im Juli 2005 wurde einem Tübinger Theologie-Absolventen der Doktorgrad aberkannt, weil 110 Seiten seiner 215 Seiten umfassenden Doktorarbeit so gut wie wortidentisch waren mit der Dissertation des Telepolis-Autors, der seine Arbeit 1996 in Buchform publiziert hat. Der Absolvent hatte magna cum laude promoviert, und ihm standen Tür und Tor offen zur Habilitation. Gegenwärtig wird geprüft, wie stark die Übereinstimmungen der übrigen Seiten der Dissertation mit einer ebenfalls bereits existierenden Arbeit (München 1992) sind. - Der Fall wurde durch einen kuriosen Zufall aufgedeckt. Doch wie viele Plagiatsfälle bei Diplom- und Doktorarbeiten bleiben für immer unentdeckt?
Alles Einzelfälle, bloße Koinzidenzen, oder doch die Spitze des Eisbergs in einem kaum noch zu entwirrenden Sumpf aus Copy/Paste, aus Einscannen und OCR-Software, aus Umschreiben, Zitat-Zitaten und 'Originalen', die nie gelesen wurden? Fest zu stehen scheint: Geistiges Eigentum flottiert frei herum - und im Netzzeitalter mehr denn je. Zu diskutieren wird in diesem Zusammenhang auch die Frage sein, wie oft Internetprojekte, die vordergründig der Vernetzung von Wissen und dem Informationsaustausch dienen, eigentlich für schamloses Copy/Paste verwendet werden (wie etwa die österreichische Arbeiten-Börse Mnemopol (3)).
Offen diskutiert werden müssten auch die negativen Aspekte des Google-Wikipedia-Wissensmonopols. Martin Löffelholz diagnostiziert in diesem Zusammenhang einen Qualitätsverlust der wissenschaftlichen Arbeiten: "Schon seit längerer Zeit machen wir an unserer Universität die Erfahrung, dass Studierende primär Instrumente wie Google oder Wikipedia nutzen, um sich einen Überblick zu einer bestimmten wissenschaftlichen Frage zu verschaffen. Damit erhalten diese Studierenden aber keineswegs den aktuellsten und relevantesten Forschungsstand. Das führt zu einem deutlichen Qualitätsverlust in der wissenschaftlichen Arbeit."
Die vielen Gesichter des Netzplagiarismus sind derzeit ein Fass ohne Boden. Was könnte oder muss daher kurzfristig getan werden?
Joan Bleicher regte an, "Netzplagiate" als neu zu definierende Delikte in die Prüfungsordnungen der Universitäten aufzunehmen. Es kann wohl in Zukunft nicht mehr genügen, wenn aufgedeckte Plagiatoren einfach um ein Um- oder Neuschreiben ihrer Arbeit gebeten werden. Sie haben ja offenbar bereits gezeigt, dass sie an genuinem Texten nicht interessiert sind oder dieses nicht beherrschen.
In einem Weblog wurde vorgeschlagen, eine Lehrveranstaltung "Einführung in alte Kulturtechniken" in die Lehrpläne aufzunehmen. Darunter fielen dann Dinge wie: mit der Hand schreiben, Texte genau lesen, in realen Bibliotheken recherchieren. Das hört sich zunächst etwas plakativ an, die erwähnten Techniken scheinen aber genau jene Kernkompetenzen zu sein, die den Studierenden in der Studieneingangsphase derzeit offenbar nicht (mehr) vermittelt werden. Und solange noch nicht alle Texte aus realen Bibliotheken eingescannt und in digitalen Archiven verfügbar sind, wird eine solche Lehrveranstaltung Sinn machen. Gegenwärtige Studierende bewegen sich oft hochselektiv in jener Menge von Texten, die online verfügbar sind. Historische Tiefe wird vorgetäuscht, indem einfach historische Zitate aus Online-Texten übernommen werden. Ein derartiges Lehrmodul könnte da wohl Abhilfe schaffen.

Im härtesten Fall könnten besonders dreiste Plagiatoren sogar von der Universität verwiesen, also exmatrikuliert werden. Medienwissenschaftler Martin Löffelholz will das für die Zukunft nicht ausschließen: “Bisher haben wir von der drakonischsten Strafe, nämlich der Exmatrikulation, abgesehen. Sollten solche Fälle jedoch zunehmen, müssen die Universitäten deutlich schärfer reagieren, um das Niveau wissenschaftlicher Arbeit und den Urheberschutz zu gewährleisten.”

Lehrmodul “Einführung in alte Kulturtechniken” gefordert

Die Wissenskultur und vor allem die Kulturtechnik der Textproduktion wandeln sich rapide. Umso erstaunlicher ist, dass es gegenwärtig zu diesem Thema kaum Forschungen gibt und auch in den Medien von einer vermeintlichen Zunahme von Textbetrug und Heuchelei in den (Kultur-)Wissenschaften - bis auf wenige Einzelfälle - nichts zu hören ist. Doch der Wandel der Konstruktion von Referenz würde eine Diskussion auf einer breiteren Ebene erfordern.
Wo bleiben die großen Forschungsprojekte, die den Anteil der aufgedeckten Plagiatsfälle an den insgesamt abgegebenen Arbeiten mit Hilfe einer repräsentativen Befragung von Lehrenden erheben? Wo bleiben die Lehrstühle oder Forschergruppen für Netzplagiate und wissenschaftliche Text-Fakes? Zumindest die Diskussion darüber, ob das Problem tatsächlich ein Problem ist, sollte nun endlich einsetzen. Der Hut brennt, denn: Ist es am Ende bereits möglich, ein Studium erfolgreich abzuschließen, ohne jemals eine Bibliothek von innen gesehen zu haben? Die Antwort von Joan Bleicher kann man sich ausmalen.
LINKS
(1) http://de.wikipedia.org/wiki/Journalismustheorien
(2) http://de.wikipedia.org/wiki/Theorie_der_Fotografie
(3) http://www.mnemopol.net/

Telepolis Artikel-URL:
http://www.telepolis.de/r4/artikel/20/20982/1.html

KAPITEL 4

Zitieren und Belegen in wissenschaftlichen Arbeiten

download http://www.bui.haw-hamburg.de/pers/klaus.lorenzen/ASP/zitierenbelegen.pdf

von Klaus F. Lorenzen, 10. Juni 2003

In jeder wissenschaftlichen Arbeit werden mehr oder minder ausgiebig die Gedanken, Ergebnisse, Daten, Materialien usw. anderer Veröffentlichungen herangezogen, verglichen, kritisch bewertet und als Grundlage eigener weiterführender Studien benutzt. Die Auseinandersetzung mit den Erkenntnissen anderer bildet die Voraussetzung, mit der eigenen Arbeit einen sinnvollen Beitrag zum Erkenntnisfortschritt leisten zu können. Eventuell entsteht dann eine neue Veröffentlichung. Die Auswertung fremden Wissens ist also nicht nur fruchtbar, sondern sogar notwendig, um Irrtümer und Doppelarbeit zu vermeiden und noch offene Fragestellungen zu erkennen. Allerdings gehört es zu den wesentlichen Grundprinzipien der Wissenschaftlichkeit, alle benutzten Quellen zu belegen – ihre Herkunft anzugeben – , damit sie in ihrem primären Kontext eingesehen und überprüft werden können. Bei Diplomarbeiten wird ihre Unterschlagung wegen der eidesstattlichen Versicherung des Autors als Betrugsversuch gewertet.

Zitierfähigkeit und Zitierwürdigkeit von Quellen

Zitieren bedeutet, sich auf Aussagen, Erkenntnisse, Ergebnisse, Daten Dritter zu berufen. Zitier*fähig* sind alle veröffentlichten, allge-

elektronische Publikationen

Graue Literatur

Diplomarbeiten

Mündliche Äußerungen, Vorträge, Ansprachen

Interviews, Gesprächsauskünfte

mein und dauerhaft zugänglichen Quellen. Problematisch können in diesem Sinne z.B. mündliche Äußerungen, Materialien in Privat- oder Geheimarchiven, aber auch elektronische Publikationen im Internet sein, deren fortdauernde Speicherung bei ihren Hosts aus vielerlei Gründen unsicher und ungewiß ist. Zudem werden sie häufig verändert (aktualisiert), so daß ihre ursprüngliche Fassung, auf die sich das Zitat bezieht, nicht mehr vorliegt! Auch die sogenannte *Graue Literatur,* eine zumeist spezielle wissenschaftliche Literatur, die in Eigenverlagen entsteht und nicht über den Buchhandel vertrieben wird, kann wegen ihrer schwierigen Beschaffbarkeit zum Grenzfall werden. Wird bislang *unveröffentlichtes* Material, z.B. fremde Datenerhebungen, verarbeitet, so wird das sinnvollerweise *im Anhang* der eigenen Arbeit bekannt gemacht. Dazu muß die Genehmigung des eigentlichen Urhebers der Daten eingeholt werden. Ebenso verfährt man bei der Benutzung noch unveröffentlichten Archivmaterials. Diplomarbeiten sind *unveröffentlichte Prüfungsarbeiten* und daher nur bedingt zitierfähig, nach strenger Auffassung sogar nicht zitierfähig (vgl. THEISEN 1990, S.133)[5]. Auf jeden Fall muß auf diesen Status hingewiesen und eine genaue Bezugsquelle der Diplomarbeit genannt werden. Mündliche Äußerungen, beispielsweise öffentliche Vorträge oder Reden, die nicht aufgezeichnet worden sind, sind u.U. nur schwierig überprüfbar und deswegen nur eingeschränkt zitierfähig. Handelt es sich um einen Rundfunk- oder Fernsehvortrag, existieren wahrscheinlich Ton- oder Filmaufzeichnungen des Senders, auf die man sich gegebenenfalls berufen kann. Besitzt man eigene Aufzeichnungen, so können diese eventuell als Anhangsmaterial veröffentlicht werden wie schon weiter oben erwähnt wurde. Bei mündlichen Auskünften, Interviews u.a. ist mindestens der genaue Personenname des Interviewten und des Interviewers, die Institution, die Adresse, Telefonnummer, Ort und Zeit des Gesprächs anzugeben.

Ob eine Quelle zitier*würdig* ist, hängt von der Thematik einer wissenschaftlichen Arbeit ab. So dürften in einer Dissertation im Fach Physik populär-wissenschaftliche Darstellungen aus der Wochenendbeilage einer Tageszeitung nur schwerlich als Primärquelle anerkannt werden. Handelt es sich hingegen bei dem Thema um eine publikati-

5. Dissertationen sind hingegen reguläre Veröffentlichungen.

onswissenschaftliche Untersuchung von naturwissenschaftlichen Inhalten in verschiedenen Zeitschriften und Zeitungen sieht es schon ganz anders aus.

Quellenkritik

Die kritische Bewertung von Quellen, also ihre Prüfung auf Richtigkeit, Vollständigkeit, Objektivität usw. ist eine der wichtigsten Anforderungen an einen Autor. Dies gilt besonders für Informationen, die im Internet gefunden werden. Da es sich fast immer um selbstorganisierte Veröffentlichungen handelt, die nicht dem Auswahlprozeß anerkannter Fachverlage unterworfen sind, ist ihre Glaubwürdigkeit und Zitierfähigkeit mit besonderer Aufmerksamkeit zu prüfen. Um diese Quellen besser einschätzen zu können, sollte man die im Internet ausgesprochen leichte und unkonventionelle Kontaktaufnahme zu den Verfassern rege nutzen. Persönliche Elektronische Post oder die entsprechenden Beiträge in Elektronischen Diskussionsforen sind wie unveröffentlichte private Korrespondenz zu behandeln: man gibt sie im Anhang der Arbeit wieder. Wird das Diskussionsforum jedoch offiziell archiviert, erfolgt der Beleg wie bei einer regulären Publikation.

Internet

Elektronische Post, Usenet-Listen

Urheberrecht im Internet

Es sei an dieser Stelle die Anmerkung erlaubt, daß selbstverständlich auch für die im Internet *frei* zur Verfügung gestellten Informationen[6] der volle Urheberrechtsschutz besteht und der Umgang mit ihnen den allgemeinen Kriterien der Wissenschaftlichkeit (vgl. Kap. 1) unterliegt.

Zitieren nach der Kurzbelegmethode

Zitat

Die wörtliche (=direkte) oder sinngemäße (=indirekte) Wiedergabe von Textstellen, Aussagen, Meßergebnissen, Materialien usw. einer Quelle bezeichnet man als zitieren. Die Wiedergabe nennt man ein *Zitat*. Ein Zitat wird durch eine bibliographische Beschreibung der Quelle, der es entnommen wurde, *belegt*. Diese bibliographische Beschreibung selbst wird häufig als *Literaturzitat* bezeichnet. Um

6. Hierzu zählen auch Bilder, graphische Darstellungen aller Art, Filme, Computerprogramme, Tonwerke usw.

Mißverständnisse zu vermeiden, wird nur die inhaltliche Wiedergabe einer Quelle als *Zitat* und die bibliographische Beschreibung als *Quellenbeleg*, *Literaturbeleg* oder kurz *Beleg* genannt.

Literaturbeleg

Literaturbelege werden in umfangreicheren Arbeiten üblicherweise in einem eigenen Literaturverzeichnis aufgeführt. Bei Diplomarbeiten ist dies Pflicht. Nur in kurzen Referaten mit wenigen Zitaten ist es akzeptabel, die Belege unmittelbar im Text als Vollbeleg in der Fußnote anzuführen, unter Verzicht auf ein eigenes Literaturverzeichnis. *Existiert ein solches, muß auch eine Methode gewählt werden, das Zitat im Text mit dem Vollbeleg im Literaturverzeichnis zu verknüpfen*.

Literaturverzeichnis

Jedes Zitat ist unmittelbar durch einen Hinweis auf die Quelle und die Entnahmestelle (Zitatstelle) zu belegen. Seit langem sind verschiedene Methoden hierfür in Übung, ohne daß der einen oder der anderen eine absolute Überlegenheit zugesprochen werden kann. Einige Vor- und Nachteile beschreibt Theisen, auch Seidenspinner (vgl. Theisen 1990, S. 134--139; Seidenspinner 1994, S. 82--85), um nur ganz wenige zu nennen. In unserem Fachbereich[7] ist in den *Richtlinien für Diplomarbeiten* die sogenannte „Kurzbelegmethode“ vorgeschrieben. Die Kurzbelegmethode kennzeichnet ein Zitat im Text durch eine bestimmte in runde Klammern gesetzte *Marke* des Typs „ Autor–Jahr “. Genau unter dieser Marke ist im Literaturverzeichnis am Ende der Arbeit der vollständige bibliographische Beleg zu finden. Die Marken dienen der Ordnung — der Sortierung — des Literaturverzeichnisses. Dessen Erstellung und Gestaltung erfolgt nach der Norm DIN 1505 Teil 2 und 3. Eine Anleitung zur Anfertigung DIN gerechter Literaturverzeichnisse mit zahlreichen Beispielen gibt LORENZEN (1997)[8]. In der Ausdrucksweise der Norm heißt eine Autor–Jahr–Marke *Einordnungsformel*, da sie bestimmt, an welche Stelle im Literaturverzeichnis der Quellenbeleg einzuordnen ist. Sie setzt sich normalerweise aus dem Nachnamen des Verfassers und dem Erscheinungsjahr der Veröffentlichung zusammen, z.B. „KRÄMER 1995“. Als Kurzbeleg eines Zitats wird diese Formel im Text in

Kurzbelegmethode

Autor--Jahr-System

Einordnungsformel

7. FH Hamburg, FB Bibliothek und Information
8. Als Postscriptdatei per Internet zu beziehen: http://www.fh-hamburg.de/pers/Lorenzen/tum/litverz.ps

runde Klammern gesetzt: „(KRÄMER 1995)“. Stammt das zitierte Werk von zwei Verfassern, können auch beide in die Einordnungsformel eingehen, z.B. „(VON ALEMANN und FORNDRAN 1990)“. Gibt es mehr als zwei Verfasser, wird mit „et al.“, bzw. „u. a.“ abgekürzt: „(DEININGER et al. 1996)“. Die Seitenzählung der Entnahmestelle des Zitats wird an den *Autor-Jahr* - Ausdruck nach Komma angehängt. Solche Zitatmarken gelten als lesefreundlich und funktionell,

Beispiel (KRÄMER 1995, S. 96)

direktes Zitat

da sie dem fachkundigen Leser die Bewertung der benutzten *Zitate* erleichtern. Denn leicht ist zu erkennen, ob sie von bekannten — und anerkannten — Autoren oder eben noch unbekannten stammen. Durch die Angabe des Erscheinungsjahres wird auf die Aktualität bzw. den historischen Bezug einer Quelle aufmerksam gemacht. Gibt es keinen Verfasser, so wird stattdessen der Herausgeber, die herausgebende Körperschaft oder gegebenenfalls ein sinntragendes Titelstichwort u. ä. gewählt.

Zur leichteren formalen Unterscheidung von direkten und indirekten (*sinngemäßen*) Zitaten (näheres im nächsten Abschnitt) wird bei indirekten Zitaten *immer das Kürzel „vgl.“* vor die Autor–Jahr–Marke gesetzt.Mitunter wird in einem Textabschnitt aus *mehreren*

Beispiel (vgl. THEISEN 1993, S. 143-144)

indirektes Zitat

Werken sinngemäß zitiert, z.B. werden unterschiedliche Auffassungen mehrerer Autoren miteinander verglichen; dann können die Belege in *einem* Kurzbeleg zusammengefaßt und durch Semikolon voneinander getrennt werden. Im Literaturverzeichnis werden diese

Beispiel (vgl. BÄNSCH 1993, S. 98; THEISEN 1993, S. 144)

Quellen jedoch separat jeweils unter ihren eigenen Einordnungsformeln aufgeführt.

Technik des Zitierens

Die Benutzung von Zitaten wird allgemein als wichtiges Indiz der Wissenschaftlichkeit angesehen. Mittels Zitaten kann der Leser in das thematische Umfeld einer wissenschaftlichen Arbeit eingeführt

werden. Andererseits werden anhand der benutzten Zitate Schlußfolgerungen über die Sachkompetenz des Verfassers gezogen (vgl. die Begriffe *zitierwürdig* und *zitierfähig* weiter oben*)*. Die Auswahl und die Anzahl der Zitate muß sachlich gerechtfertigt sein und nicht bloß Autoritätsgläubigkeit oder Belesenheit demonstrieren. Ein schwerwiegender Fehler liegt vor, wenn sinnentstellend und einseitig zitiert wird, um die eigene Argumentation durch Berufung auf anerkannte Autoritäten abzusichern. Einige wichtige Gründe für das Zitieren sind

sachliche Begründung

keine Sinnentstellung!

- Erläuterung des derzeitigen wissenschaftlichen Erkenntnisstandes im Hinblick auf das zu bearbeitende Thema
- Mitbenutzung von primärem Quellenmaterial anderer Bearbeiter
- Nachweis der geistigen Urheberschaft (ideengeschichtlicher Hintergrund)
- Nachweis der Urheberschaft von Meßmethoden oder Erhebungstechniken, Meßdaten, Versuchsanordnungen usw.

Wörtliche oder „direkte“ Zitate

Wörtliche Zitate müssen *wortwörtlich und zeichengetreu* der Vorlage entsprechen, d.h. eine authentische Kopie der Vorlage darstellen. Zeichengetreu bedeutet, daß die Zeichen in Schriftschnitt, --auszeichnung und Anordnung, also auch fehlerhafter, wiederzugeben sind und nicht verändert werden dürfen. Wörtliche Zitate werden in ein- und ausleitende Zitierzeichen gesetzt. Im Deutschen sind die sog.

wortwörtlich, zeichengetreu

Zitat

„Das hätte ich nicht gedacht!“

„Nicht verwunderlich, keiner von uns war darauf gekommen.“

„Gänsefüßchen“ sehr verbreitet, die aus doppelten oder auch einfachen Anführungs- und Abführungszeichen bestehen („ , “). In Druck-

Beispiel 1

Zitat

›Das hätte ich nicht gedacht!‹

›Nicht verwunderlich, keiner von uns war darauf gekommen.‹

werken wird die wörtliche Rede häufig durch *guillemets*, einfache (oder zweifache) spitzwinklige Klammern, kenntlich gemacht.

Ein direktes Zitat darf auch gekürzt werden, wenn dadurch nicht der Sinn entstellt wird. Solche Auslassungen — auch Ellipsen genannt — werden durch drei eng gesetzte Punkte gekennzeichnet („…“). Auslassungen *vor* dem wörtlichen Zitatbeginn brauchen nicht gekennzeichnet zu werden. Das eigentliche Zitat beginnt mit den „Gänsefüßchen“ unten.

Beispiel 2

Vorlage

Wahrscheinlichkeitstheoretische Überlegungen spielen in der modernen physikalischen Forschung eine entscheidende Rolle; dennoch fehlt bisher eine befriedigende, als widerspruchsfrei erweisbare Definition des Wahrscheinlichkeitsbegriffs …

Ellipse

.

Zitat

„Wahrscheinlichkeitstheoretische Überlegungen spielen …eine entscheidende Rolle; dennoch fehlt bisher eine befriedigende, als widerspruchsfrei erweisbare Definition des Wahrscheinlichkeitsbegriffs “(POPPER 1994, S. 106).

Längere wörtliche Zitate (mehr als 3 Textzeilen) werden innerhalb des normalen Textflusses am besten als beidseitig eingerückter Textblock in kleinerer Schriftgröße bei kleinerem Zeilenabstand wiedergegeben (vgl. Beisp. 4 und 7). Bei einem solchermaßen als *wörtlich* gekennzeichneten Zitat entfallen die Anführungs- und Abführungszeichen!

Beispiel 3

Auslassung im Zitat

Zitat

Wahrscheinlichkeitstheoretische Überlegungen spielen in der modernen physikalischen Forschung eine entscheidende Rolle; dennoch fehlt bisher eine befriedigende, als widerspruchsfrei erweisbare Definition des Wahrscheinlichkeitsbegriffs …(POPPER 1994, S. 106).

Manchmal kommt es vor, daß eine wörtlich zitierte Passage ihrerseits ein Zitat enthält. Letzteres wird als *Zitat im Zitat* durch einfache An– und Abführungszeichen „ ‚“ , „ ‘ “ von diesem abgesetzt. Den Quellenbeleg für dieses Zitat im Zitat übernimmt man von dem zitierten Autor so wie dort vorgefunden. In dem folgenden Beispiel wird der Autor KARL POPPER wörtlich zitiert mit einem Text, in dem ein Aus-

Beispiel 4

längeres wörtliches Zitat

druck des Autors WAISMANN wörtlich zitiert und natürlich auch belegt worden ist..

Zitat im Zitat

Zitat

> Popper erkennt darin einen neuen Ansatz, der „die Wahrscheinlichkeitsaussagen nicht psychologisch, sondern *logisch* [im Orig. hervorgeh.] deutet, sozusagen als Aussagen über die ‚logische Nähe' von Sätzen" (POPPER 1994, S. 108 mit e. Zitat v. WAISMANN 1930, S. 237).

Man beachte, daß der grammatikalisch abschließende Punkt des zitierten Satzes *nach* der Zitatformel gesetzt wird!

Beispiel 5

Wegen des schon oben erwähnten Gebots der Zeichentreue müssen im Original vorgefundene Textauszeichnungen genau so im Zitat wiedergegeben werden. Meistens verfügt man aber nicht über die satztechnischen Hilfsmittel des Originals und muß darum im eigenen Zitat zu Ersatzdarstellungen greifen. Auf diese wird dann durch eine in eckige Klammern gesetzte Bemerkung hingewiesen, um beim Leser kein Mißverständnis bezüglich der Zuordnung aufkommen zu lassen: *[Hervorhebung i. Original]* u.a. Wenn der Zitierende selbst ein bestimmtes Wort als wichtigen Schlüsselbegriff in einem Zitat hervorhebt, muß er einen entsprechenden Hinweis (am besten mit seinem Namen) in eckigen Klammern geben.

Veränderungen der Vorlage bei wörtlichen Zitaten

Zitat

> Schlechte Arbeiten haben ihre häufigste Ursache in einer *fehlenden Planung* [meine Hervorhebung, K.F.L.]: Es wird ziellos herumgewurschtelt, mal ein bißchen dort gespritzt oder dort poliert (KRUMM 1995, S. 69).

Genauso ist bei etwaigen *Ergänzungen, Zusätzen, grammatikalischen Anpassungen* usw. zu verfahren. Dem Leser muß jede Veränderung am Zitat und ihr Verursacher klargemacht werden. Mit den Zeichen „[!]" oder „[sic]" macht der Zitierende den Leser auf im Original vorgefundene ungewöhnliche Schreibweisen oder Ausdrücke, auch offenkundige Fehler aufmerksam, um die Zuordnung zu sichern und darauf hinzuweisen, daß es sich nicht um einen eventuellen Übertragungsfehler handelt.

Beispiel 6

[!] , [sic]

Beispiel eines längeren wörtlichen Zitats

Im folgenden Beispiel werden verschiedene der oben besprochenen Zitiertechniken, insbesondere die Behandlung von *Zitaten in einem Zitat* — sowohl direkte wie indirekte Zitate — und eine grammatikalische Veränderung der originalen Satzstellung, angewandt.

.

Zitat

> Wie soll bei fortdauernden Sparhaushalten ein , benutzerorientierter Leistungsabbau‘ (FLEMING 1996, S. 97) betrieben werden? Fragen dieser Art betreffen die Qualität eines Dienstleistungsangebots. Um *qualitative* [i. Orig. hervorgeh.] Leistungen meßbar und damit kontrollierbar zu machen, müssen Lektorinnen und Lektoren für das Bibliothekssystem in Jahresberichten und möglichst auch in der Presse Aussagen über Ziele und Zwecke des Bestandsaufbaus, über Bestandsprofile einzelner Einrichtungen, neue Mediensegmente, hauptsächliche Zielgruppen, differenziert nach Bestandsgruppen, über anzustrebende Ausbaustufen einzelner Bestandssegmente machen können (vgl. UMLAUF 1995, S. 243). ... angesichts [der] oben angeführte[n] Fragen [ist es] offensichtlich, daß althergebrachte Aufbau– und Ablauforganisationsstrukturen, die immer nur mittels quantitativer Leistungsmessungen überprüfbar waren, für heutige Anforderungen im Dienstleistungsmanagement nicht mehr angemessen sind (vgl. HAPPEL 1996, S. 7 ff.) (SCHMOLLING 1996, S. 179; mit 3 weiteren Zitaten)

direkte und indirekte Zitate innerhalb eines direkten Zitats

grammatikalische Ergänzungen im Zusammenhang mit einer Auslassung

Beispiel 7

Der Hinweis „mit 3 weiteren Zitaten“ im Kurzbeleg dieses wörtlichen Zitats macht eindeutig klar, daß die zitierten Autoren *innerhalb* des Zitats der Autorin SCHMOLLING zuzuschreiben sind. Diese ist für die Korrektheit der Quellenbelege ihrer Zitate *allein* verantwortlich. Der Zitierende muß diese gewöhnlicherweise nicht auch noch überprüfen. Es sei denn, er benutzt diese gefundenen Zitate für seine eigene weitere Argumentation. Dann handelt es sich um *Sekundärzitate* und deren Behandlung wird noch im nachfolgenden auf S. 11 beschrieben.

Fremdsprachige Zitate

Englisches Zitat

Zitate in englischer Sprache können im Wortlaut übernommen werden. Sie gilt als internationale Wissenschaftssprache schlechthin und ihre Beherrschung wird vorausgesetzt. Andere fremdsprachige Texte werden aus Gründen der Leserorientierung besser ins Deutsche übertragen und dann auch deutsch zitiert. Es kann sinnvoll sein, den Originaltext als Fußnote bzw. bei längeren Passagen als Anhang mitzugeben.

Sinngemäße („indirekte“) Zitate

Kennzeichnung eines indirekten Zitats

Gedankliche Anregungen, Übernahme von Ergebnissen und Erkenntnissen anderer Autoren, die man mit *eigenen* Worten in die eigene Arbeit einfließen läßt, sind sinngemäße Zitate, die genau so zu belegen sind wie wörtliche Zitate (vgl. die Beispiele auf S. 5!). An die Stelle der An– und Abführungszeichen, die wörtlich übernommene Passagen kennzeichnen, treten allerdings sprachliche Hilfsausdrücke. Meistens wird man den *Beginn* eines indirekten Zitats mit einer Erwähnung des zitierten Autors einleiten und mit dem eigentlichen Beleg beenden. Zur Unterscheidung von den direkten Zitaten wird der Beleg stets mit dem Vermerk „‘vgl.’“, d.i. „‘vergleiche!’“, eingeleitet. Das nachfolgende Beispiel bezieht sich auf das weiter oben gegebene längere Zitat. Stilistisch bieten sinngemäße Zitate die Möglichkeit, eine unschöne Häufung von direkten Zitaten des gleichen Autors — womöglich noch auf einer Seite — durch Konzentration auf die wesentliche Aussage zu vermeiden. Auch innerhalb von indirekten Zitaten können weitere direkte oder indirekte Zitate auftreten. Sie werden nicht anders behandelt als die schon weiter oben bespro-

Beispiel (vgl. THEISEN 1993, S. 143-144)

Zitat

... SCHMOLLING untersucht angesichts erheblicher finanzieller Einsparungen in den Bibliotheksetats den Sektor Bestandsaufbau. Um hier zu einem effektiven Mitteleinsatz zu kommen, verlangt sie, daß die bisherige Ausrichtung auf quantitative Leistungswerte ersetzt wird durch neue Ansätze zu qualitativer Leistungsbewertung, um strategische Entscheidungshilfen bei der Wahl zwischen verschiedenen Alternativen des Bestandsaufbaus gewinnen zu können (vgl. SCHMOLLING 1996, S. 179).

Beispiel 8

chenen Fälle. Beispielsweise wird ein direktes Zitat im indirekten Zitat in An– und Abführungszeichen gesetzt und unmittelbar belegt als *indirekte* Fundstelle (s. Beispiel 9). Vergleiche auch das o.g. Beispiel 5 eines direkten Zitats in einem direkten Zitat (POPPER / WAISMANN).

direktes Zitat in einem indirekten Zitat

Zitat

Bei der Untersuchung, was Wissenschaft eigentlich ist, machen VON ALEMANN UND FORNDRAN (1990) besonders auf die Komponente der Kritik, Wissenschaft als Kritik der Wissenschaft und der Wirklichkeit aufmerksam. Für sie ist Kritik mehr als positive Abbildung der Realität, nämlich „eine Methode, die auf Erkenntnis und Aufhebung von Widersprüchen zielt“ (TUDYKA 1973, S. 9). Die angesprochenen grundsätzlichen erkenntnistheoretischen Positionen werden nicht weiter verfolgt und es wird nur festgestellt, daß in den Grundfragen zwar ein gewisser Konsens, aber ebenfalls Konflikt herrscht (vgl. VON ALEMANN UND FORNDRAN 1990, S. 42--45; mit e. Zit. v. TUDYKA 1973).

Beispiel 9

Sekundärzitate

Wissenschaftlich korrekte Zitate sind von ihrem Anwender in der Primärquelle selbst ausgewählt und überprüft worden, sie sind „unmittelbar“ (SEIDENSPINNER 1994, S. 85). Zitate aus anderen Quellen einfach zu übernehmen, kann leicht zu Fehlinterpretationen führen und ist nicht gerade ein Ausdruck überzeugender Selbständigkeit. Es gibt aber auch begründete Fälle für den Einsatz von Sekundärzitaten, dann nämlich, wenn die Primärquelle nicht mehr existiert, z.B. eine verschollene historische Quelle, oder wenn ihre Einsichtnahme nur mit ungerechtfertigtem Aufwand verbunden ist, z.B bei ausländischer Literatur. Das Primärzitat ist dann *nur in der Fassung* einer Sekundärquelle bekannt und ist somit ein Pseudoprimärzitat. Zur Klarstellung wird dem Beleg der Sekundärquelle „zitiert nach“ vorangestellt. Im folgenden Beispiel 10 wird ein Zitat aus einem Zeitschriftenaufsatz aus dem Jahr 1895 gebracht. Aber nicht der alte Zeitschriftenjahrgang hat vorgelegen, sondern sein, vermutlich auszugsweiser, Nachdruck in einer Monographie aus dem Jahr 1970.

Pseudoprimärzitat

SCHMOLLING zitiert (1996, S. 164) wörtlich aus diesem Nachdruck, den THAUER (1970) herausgegeben hat.

Sekundärzitat

Zitat

> „ Wer das nicht versteht und glaubt, jeder Bücherfreund wisse genug und passe zum Bibliothekar, der hat einfach gar keine Vorstellung von der ungeheuren Masse Bücher, die es giebt [sic], und unter denen der Bibliothekar Bescheid wissen muß“ (NÖRRENBERG 1895, S. 2242,43; zit. n. THAUER 1970, S. 173f.).

Beispiel 10

Bei einer Weiterverwendung *dieser* Fassung, hätte man sogar den seltenen Fall eines Tertiärzitats – wofür es aber keine Entschuldigung gibt, da das Buch von THAUER ja noch leicht beschaffbar sein dürfte.

Tertiärzitat

Zitat

> „ Wer das nicht versteht und glaubt, jeder Bücherfreund wisse genug und passe zum Bibliothekar, der hat einfach gar keine Vorstellung von der ungeheuren Masse Bücher, die es giebt [sic], und unter denen der Bibliothekar Bescheid wissen muß“ ((NÖRRENBERG 1895, S. 2242,43; zit. n. THAUER 1970, S. 173f.) zit. n.SCHMOLLING 1996, S.164)..

Beispiel 11

Literaturverzeichnis

KRÄMER, Walter: *Wie schreibe ich eine Seminar-, Examens- und Diplomarbeit : eine Anleitung zum wissenschaftlichen Arbeiten für Studierende aller Fächer an Universitäten, Fachhochschulen und Berufsakademien.* 4., erw. u. aktualis. Aufl. Stuttgart ; Jena : G. Fischer, 1995 (UTB für Wissenschaft : Uni-Taschenbücher 1633). 199 S. - ISBN 3-437-40342-7

SEIDENSPINNER, Gundolf: *Wissenschaftliches Arbeiten : Techniken, Methoden, Hilfsmittel ; Aufbau, Gliederung, Gestaltung ; richtiges Zitieren.* 9. Aufl., Neuausg. München ; Landsberg a. Lech : mvg-verl., 1994 (Schriften der Deutschen Studentenschaft). 127 S. - ISBN 3-478-71310-2

THEISEN, Manuel René: *Wissenschaftliches Arbeiten : Technik - Methodik - Form.* 3., erw. u. aktualisierte Aufl. München : Vahlen, 1989 (WiSt-Taschenbücher). - 236 S. - ISBN 3-8006-1366-2