



FOM Hochschule für Oekonomie & Management

Hochschulzentrum Düsseldorf

Bachelor Thesis

im Studiengang Wirtschaftsinformatik

zur Erlangung des Grades eines

Bachelor of Science (B.Sc.)

über das Thema

LATEX-Vorlage - mit Biblatex

von

Max Mustermann

Betreuer : Prof. Dr. Peter Lustig

Matrikelnummer : 123456

Abgabedatum : 1. Dezember 2025

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VI
Symbolverzeichnis	VII
Glossar	VIII
1 Einleitung	1
1.1 Aufbau der Arbeit	1
2 Informationen vom Thesis-Day	2
2.1 Pre-Anmeldephase	2
2.1.1 Vorüberlegungen	2
2.1.2 Anregungen finden	2
2.2 Anfertigungsphase	3
2.3 Post-Abgabephase	3
3 Latex-Details	5
3.1 Verwendete Software, Editor und Zusatzpakete	5
3.1.1 Windows 8+	5
3.1.2 Mac OSX und iOS	5
3.1.3 Online	5
3.2 Dokumentenklasse	5
3.3 Grafiken	6
3.4 Quellcode	6
3.5 Tabellen	8
3.6 Biblatex	8
3.6.1 Erklärung	8
3.6.2 Beispielfußnoten	9
3.6.3 KI-generierte Quellen	9
3.6.3.1 Struktur für KI-Einträge:	10
3.6.3.2 Wichtige Hinweise:	10
3.6.3.3 Beispiel für die Zitierung:	10
3.7 Abkürzungen	10

3.8	Formeln	11
3.9	Symbole	11
3.10	Glossar	12
3.11	Listen und Aufzählungen	12
3.11.1	Listen	12
3.11.2	Aufzählungen	12
3.11.2.1	Tiefste Ebene 1	12
3.11.2.2	Tiefste Ebene 2	13
3.12	Skript zum Kompilieren	13
3.13	PlantUML	13
4	Fazit	13
	Anhang	14
	Anhang A: Dokumentation der KI-Prompts	16
	Anhang B: Code-Listing - Beispiel für Datenverarbeitungsskript	17

Abbildungsverzeichnis

Abbildung 1: Verzeichnisstruktur der \LaTeX -Dateien	1
Abbildung 2: FOM-Vorgaben zur Thesis im Online-Campus	3
Abbildung 3: Titel der Abbildung hier	7

Tabellenverzeichnis

Tabelle 1: Beispieltabelle 1	8
--	---

Abkürzungsverzeichnis

OC FOM Online Campus

WYSIWYG What you see is what you get

Symbolverzeichnis

A	Aufrechter Buchstabe
\mathbb{N}	Menge aller natuerlichen Zahlen ohne die Null

Glossar

Glossar In einem Glossar werden Fachbegriffe und Fremdwörter mit ihren Erklärungen gesammelt.. 12

Glossaries Glossaries ist ein Paket was einen im Rahmen von LaTeX bei der Erstellung eines Glossar unterstützt.. 12

1 Einleitung







Dies soll eine \LaTeX -Vorlage für den persönlichen werden. Sie hat weder einen Anspruch auf Richtigkeit, noch auf Vollständigkeit. Die Quellen liegen auf Github zur allgemeinen Verwendung. Verbesserungen sind jederzeit willkommen.

onZielsetzung Kleiner Reminder für mich in Bezug auf die Dinge, die wir bei der Thesis beachten sollten und \LaTeX -Vorlage für die Thesis.

1.1 Aufbau der Arbeit

Kapitel 2 enthält die Inhalte des Thesis-Days und alles, was zum inhaltlichen erstellen der Thesis relevant sein könnte. In Kapitel 3 Latex-Details findet ihr wichtige Anmerkungen zu \LaTeX , wobei die wirklich wichtigen Dinge im Quelltext dieses Dokumentes stehen (siehe auch die Verzeichnisstruktur in Abbildung 1).

Abbildung 1: Verzeichnisstruktur der \LaTeX -Dateien

Name	Änderungsdatum	Typ	Größe
 abbildungen	29.08.2013 01:25	Dateiordner	
 kapitel	29.08.2013 00:55	Dateiordner	
 literatur	31.08.2013 18:17	Dateiordner	
 skripte	01.09.2013 00:10	Dateiordner	
 compile.bat	31.08.2013 20:11	Windows-Batchda...	1 KB
 thesis_main.tex	01.09.2013 00:25	LaTeX Document	5 KB

Quelle: Eigene Darstellung

2 Informationen vom Thesis-Day

Siehe auch Wissenschaftliches Arbeiten¹. Damit sollten alle wichtigen Informationen abgedeckt sein ;-)² Hier gibt es noch ein Beispiel für ein direktes Zitat³

2.1 Pre-Anmeldephase

2.1.1 Vorüberlegungen

Trichtermethode: Man beginnt mit der eigentlichen Konklusion und überlegt dann, welche allgemeinen Teile dafür benötigt werden.

Welchen Mehrwert soll die Arbeit bieten⁴? Auch darüber nachdenken, wie die Arbeit einen selbst weiter bringen kann. Studienverlauf prüfen. Welche Vorlesungen hat mich besonders interessiert? Wo liegen meine Stärken etc.

1. Themenfindung
2. Literaturrecherche
3. Gliederung/Motivationspapier erstellen
4. Betreuerwahl (siehe Liste im FOM Online Campus (OC))
5. Anmeldung (ab 141 Credits möglich)

2.1.2 Anregungen finden

- www.diplom.de
- www.hausarbeiten.de
- Datenbanken aus Tools and Methods
- etc.

¹ Balzert.2008.

² Balzert.2008.

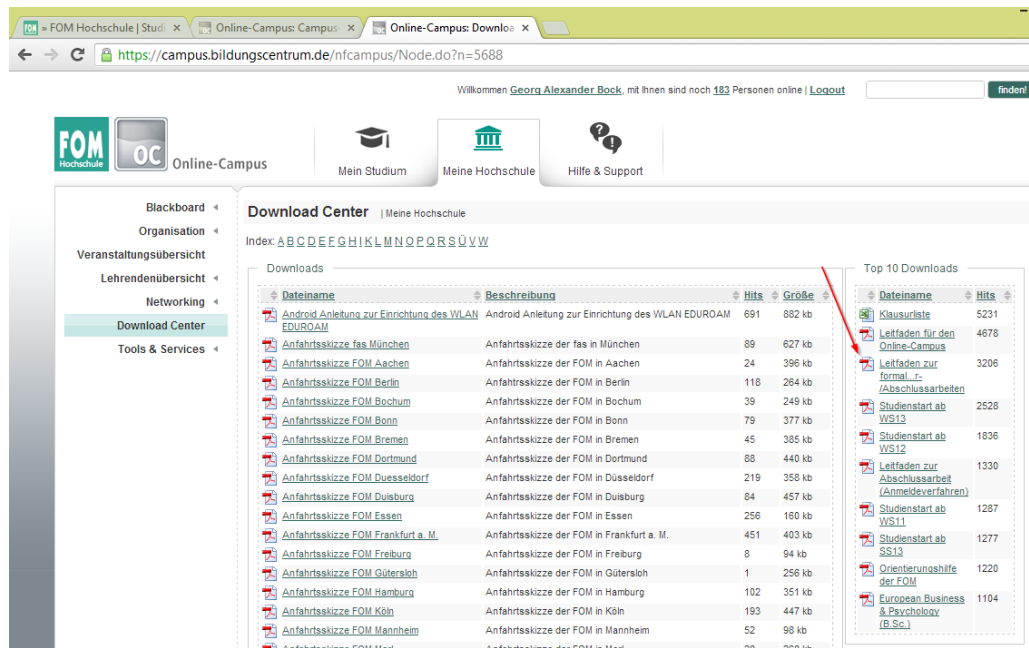
³ Balzert.2008.

⁴ Diese Fußnote hat inhaltlich keinen Sinn. Es soll nur ein langer Text generiert werden, dass dieser Vermerk über zwei Zeilen reicht und bündig dargestellt wird.

2.2 Anfertigungsphase

Die Anmeldung ist mittlerweile jeden Mittwoch möglich.

Abbildung 2: FOM-Vorgaben zur Thesis im Online-Campus



FOM

Laut Herrn Keller sollte der Umfang der Thesis (für eine gute Note) eher im Bereich der 60 Seiten liegen. Wie immer ist das vermutlich mit dem Betreuer abzustimmen. Die Liste der Dozenten, die Abschlussarbeiten betreuen, findet sich auch im OC.

Zeit zur Erstellung der Thesis 2-4 Monate.

Es müssen zwei gedruckte Arbeiten abgegeben werden. Flüchtige Quellen als PDF ausgeben lassen und auf CD abgeben. Thesis zusätzlich digital einreichen. Beim Binden der Thesis auf Qualität achten. Haptik und erster Eindruck sind in der Bewertung „auch“ wichtig. Arbeiten können in jedem FOM Studienzentrums abgegeben werden.

2.3 Post-Abgabephase

Nach Abgabe ca. 2 Wochen bis zum Kolloquium.

Kolloquium:

- Dauer: 30 Minuten

- Präsentation (manche Prüfer wollen eine, andere nicht)
- Betreuer vorher fragen was er möchte
- Es gibt einen Frageteil, dieser bezieht sich auf die Arbeit, kann aber auch darüber hinaus gehen.
- Der Tag des Kolloquiums steht auf der Endbenotung
- Thesis und Kolloquium sind zwei getrennte Prüfungsbereiche. Für beide gibt es nur zwei Versuche.
- Am Tag des Kolloquiums erhält man die Bestätigung, ob bestanden oder nicht

3 Latex-Details

3.1 Verwendete Software, Editor und Zusatzpakete

3.1.1 Windows 8+

- MikTeX: 2.9, 32-bit
- Biblatex: 3.5, Zusatz: Biber.exe
- Editor: TexStudio (kann ich empfehlen), Notepad++

3.1.2 Mac OSX und iOS

- MacTeX: <https://tug.org/mactex>
- Editor: TeXPad <https://www.texpadapp.com>

3.1.3 Online

Overleaf ist eine Online-Anwendung mit der Ihr direkt im Browser an eurer Thesis schreiben könnt. Bis 1GB Größe und maximal 60 Einzeldateien könnt ihr Overleaf kostenlos nutzen: <https://www.overleaf.com/>

3.2 Dokumentenklasse

Eigentlich hatte Prof. Finke empfohlen die Dokumentklassen „Book“ oder „Report“ für die Erstellung der Bachelor-Thesis zu verwenden, da diese über weitere Gliederungsebenen verfügen. Ich verwende dennoch eine leicht modifizierte Komaskript-Klasse „scrartcl“, mit der Erweiterung um eine Ebene. Siehe (skripte/weitereEbene.tex). Das Skript stammt irgendwo aus den Netz und übersteigt meine \LaTeX -Fähigkeiten. Dadurch kann ich über eine weitere Ebene in der Arbeit verfügen, ohne mich mit der Modifikation von Kapitel-Seiten rumschlagen⁵ zu müssen. Diese Quelle ist nur zur Demonstration und hat keinen inhaltlichen Bezug hierzu. Es werden übrigens nur die Quellen im Literaturverzeichnis angezeigt, die auch referenziert sind.

⁵ Tanenbaum.2003.

3.3 Grafiken

Das Paket `\usepackage{float}` ermöglicht es die Grafiken und Tabellen an der Stelle im Text zu positionieren, wo diese im Quelltext stehen (Option H). Ansonsten würde \LaTeX diese dort unterbringen, wo es typographisch sinnvoll wäre - das wollen wir ja nicht ;-).

Die Breite der Grafiken am Besten relativ zum Text angeben.

3.4 Quellcode

Quellcode kann auf unterschiedliche Arten eingebaut werden. Zum einen kann es hier durch direktives Einbinden in der Kapitel-Datei geschehen.

```
1 | % Hier wird aufgezeigt, wie man eine Grafik einbindet, es wird also in der PDF
   |   angezeigt,
2 | % da es in einem Quellcode-Listing steht.
3 | % Auch wenn es hier faelschlicherweise als LaTeX-Befehl angezeigt wird.
4 | \includegraphics[width=0.9\textwidth]{sup}
```

Bei längeren Quellcode-Listings empfiehlt es sich jedoch auf eine externe Datei im Ordner Quellcode zu verlinken und diese einzubauen:

```
1 | <!-- So können Tabs definiert werden -->
2 | <ul class="tabs">
3 |     <li class="tab-title">
4 |         <div class="tab-content">
5 |             </div>
6 |     </li>
7 | </ul>
```

Statt dem Package `Istlisting`, welches direkt auf Tex basiert, kann auch das Package `minted` verwendet werden. Dieses Package basiert auf `python-pygments` und unterstützt weit mehr Sprachkonstrukte als `Istlisting`. Um das Paket zu verwenden muss es eingebunden werden und zusätzlich `python-pygments` installiert sein. (Dies ist mit im Dockerfile vorhanden. Für die anderen Compile-Methoden, wie das native verwenden von Tex Live findet sich hier die Installationsanleitung für das `minted` Paket: <https://ctan.org/pkg/minted?lang=de>)

Damit das kompilieren ohne Python trotzdem möglich ist, ist die Funktion standardmäßig ausgebaut. Deshalb muss zusätzlich in der Datei

```
thesis_main.tex \usepackage{minted}
```

wieder einkommentiert werden.

Minted lässt sich dann ganz ähnlich zu `Istlisting` verwenden:

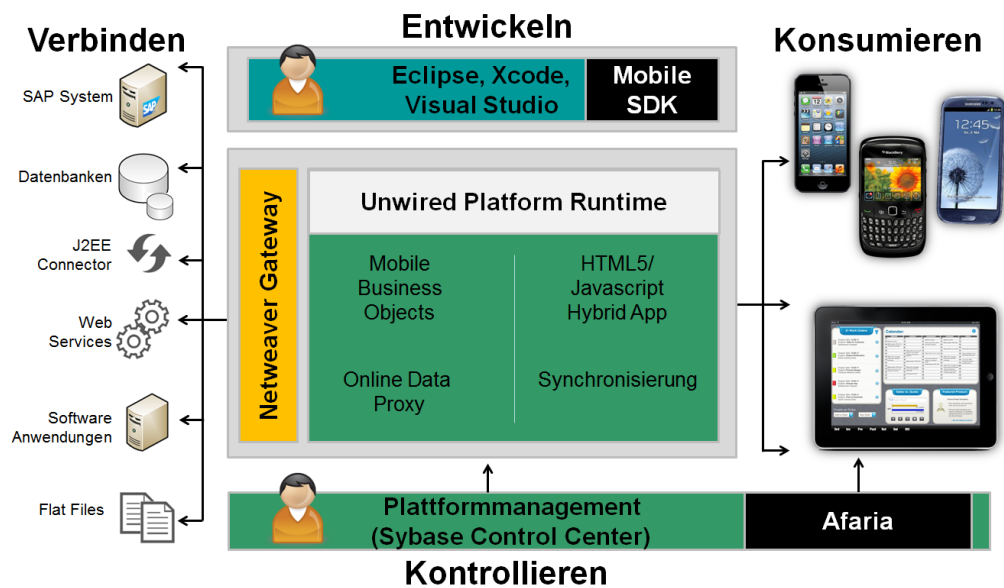
```

1      \begin{minted}{c}
2          int main() {
3              printf("hello, world");
4              return 0;
5          }
6      \end{minted}

```

Da der Pfad zu den Abbildungen im Hauptdokument definiert wurde, muss hier nur noch der Name des Bildes ohne Dateiendung stehen (sup).

Abbildung 3: Titel der Abbildung hier



Quelle: Eigene Darstellung

3.5 Tabellen

Tabelle 1: Beispieltabelle 1

Abkürzung	Beschreibung	Berechnung
MEK	Materialeinzelkosten	
MGK	Materialgemeinkosten	+ ↑ *
FEK	Fertigungseinzelkosten	
FGK	Fertigungsgemeinkosten	+ ↑ *
SEKF	Sondereinzelkosten der Fertigung	
= Herstellungskosten		
VwGK	Verwaltungsgemeinkosten	+ ↑ *
VtGK	Vertriebsgemeinkosten	+ ↑ *
SEKVt	Sondereinzelkosten des Vertriebes	
= Selbstkosten		
+ Gewinnaufschlag		
+ Rabatte		
= Nettoverkaufspreis (NVP)		
+ Umsatzsteuer		
= Bruttoverkaufspreis (BVP)		

Beckert.2012

3.6 Biblatex

3.6.1 Erklärung

Von den vielen verfügbaren Literatur-Paketen habe ich mich für Biblatex entschieden. Die Anforderungen der FOM sollten hiermit erfüllt sein. Ich habe bisher nur Einträge „@book“ getestet. Wie immer steckt der Teufel hier im Detail und es wird sich später herausstellen, ob Biblatex eine gute Wahl war. Die Anpassungen hierfür liegen unter skripte/modsBiblatex. Ich verwende das Backend Biber, welches bib-Dateien in UTF-8 verarbeiten kann.

In der für den Leitfaden 2018 aktualisierten Version sind außerdem Beispiele für „online“,⁶ also Webseiten, und „article“,⁷ also wissenschaftliche Artikel, enthalten.

Laut Leitfaden sollen in einer Fußnote maximal 2 Autoren genannt werden. Bei Quellen mit drei oder mehr Verfassern wird nur der erste Autor aufgeführt, gefolgt von „et. al.“ bzw. „u.a.“. Im Literaturverzeichnis hingegen müssen alle Autoren angegeben werden. Beim Zitieren dieser Werke werden in der Fußnote die Autoren korrekt genannt, wie

⁶ [website:angular:aboutAngular](#).

⁷ Decker2009.

in dieser Fußnote⁸, dieser Fußnote⁹ und dieser Fußnote¹⁰ zu sehen ist. In der Datei `literatur/literatur.bib` können daher alle Autoren ohne Bedenken erfasst werden. BibTeX findet die richtige Zitierweise automatisch. Die maximale Anzahl der angezeigten Autoren in Zitaten wird übrigens durch die Option „maxcitenames=3“ des *biblatex*-Packages festgelegt.

3.6.2 Beispielfußnoten

Diese Fussnote soll zeigen, wie mit einem „von“ vor dem Namen des Autors umgegangen wird¹¹. Man muss für die korrekte Sortierung eines solchens Namens im Literaturverzeichnis einen „sortkey“ setzen.

Diese Fussnote soll zeigen, wie mit einer Online-Quelle ohne Jahresangabe umgegangen wird¹².

Diese Fußnote¹³ ist nur dazu da zu zeigen, wie mit mehreren Quellen des selben Autors aus dem selben Jahr umgegangen wird, wenn das Stichwort gleich bleibt¹⁴ oder sich ändert¹⁵. Laut Leitfaden sollte bei gleichem Autor, Jahr und Stichwort ein Buchstabe an die Jahreszahl gehen. Zum Beispiel 2012a.

Die folgenden Fußnoten dienen dazu zu zeigen, dass die Nummern von zwei direkt aufeinanderfolgende Fußnoten mit Komma getrennt werden.^{16,17}

3.6.3 KI-generierte Quellen

Gemäß den FOM-Richtlinien müssen KI-generierte Inhalte transparent dokumentiert werden. Die Zitierung erfolgt über das Schlüsselwort `ai-generated` und wird in einem separaten Verzeichnis ausgegeben.

⁸ Balzert0.2008.

⁹ Balzert1.2008.

¹⁰ Balzert2.2008.

¹¹ Lucke2018.

¹² Belastungsdienst.

¹³ Beckert.2012.

¹⁴ Beckert.2012.1.

¹⁵ Beckert.2012.2.

¹⁶ Beckert.2012.1.

¹⁷ Lucke2018.

3.6.3.1 Struktur für KI-Einträge:

```

1 @misc{KI_Model_Jahr_Beschreibung,
2   usera = {KI-generierte Inhalte},
3   author = {{Modellname und Version}},
4   title = {Genaue Aufgabenbeschreibung},
5   year = {Jahr},
6   month = {Monat},
7   howpublished = {\url{plattform-url}},
8   note = {Generiert am TT.MM.JJJJ},
9   addendum = {Modellname und Version},
10  keywords = {ai-generated}
11 }

```

3.6.3.2 Wichtige Hinweise:

- Jede KI-Nutzung muss im Literaturverzeichnis erfasst werden
- Das verwendete Modell und die Version sind anzugeben
- Der genaue Prompt oder die Aufgabenbeschreibung ist zu dokumentieren
- KI-Quellen werden automatisch in einem separaten Verzeichnis aufgeführt
- In der Eigenständigkeitserklärung ist die KI-Nutzung zu bestätigen

3.6.3.3 Beispiel für die Zitierung:

Die Erstellung der tabellarischen Taxonomie-Übersicht für die Literaturrecherche erfolgte durch Anthropic Claude 4 Sonnet¹⁸. Dieser Eintrag wird im separaten Verzeichnis „KI-generierte Quellen“ aufgeführt und nicht im regulären Literaturverzeichnis.

3.7 Abkürzungen

Abkürzungen werden mithilfe des Pakets Acronym eingebunden. Alle Abkürzungen sollten in der Datei `acronyms.tex` mithilfe des

`\acro`

Befehls festgelegt werden. Im Text werden diese dann mit

¹⁸ Vgl. *Anthropic Claude 4 Sonnet*, KI-generierte Inhalte, 2025.

`\ac{Abkürzung}`

benutzt. Bei der ersten Verwendung einer Abkürzung wird der Begriff in beiden Formen dargestellt. So wie hier: What you see is what you get (WYSIWYG). Nur wenn eine Abkürzung tatsächlich verwendet wird erscheint sie auch im Abkürzungsverzeichnis.

Sollte es im Abkürzungsverzeichnis zu Anzeigefehlern kommen kann dies daher rühren, dass eine Abkürzung verwendet wird, die länger ist als WYSIWYG. In diesem Fall müsst ihr in der Datei `acronyms.tex` den Parameter `[WYSIWYG]` durch eure längere Abkürzung ersetzen.

3.8 Formeln

Um eine Formel nach links auszurichten muss sie zwischen `&` und `&` eingesetzt werden:

Formel 1: Erste Formel

$$L_P = 10 \lg \cdot \frac{P}{1mW} \tag{1}$$

Beckert.2012

Etwas mehr Text.

Ansonsten wird sie mittig ausgerichtet test.

Formel 2: Zweite Formel

$$L_P = 10 \lg \cdot \frac{P}{1mW} \tag{2}$$

Beckert.2012

3.9 Symbole

Das hier ist ein definiertes Symbol: \mathbb{N} und das hier auch \mathbb{A} . Symbole werden in der Datei `Skripte/symboldef.tex` zentral definiert.

3.10 Glossar

Begriffserklärungen bzw. das Glossar wird mithilfe des Pakets Glossaries eingebunden. Alle Begriffe die erklärt werden sollen, sollten in der Datei glossar.tex mithilfe des

```
\newglossaryentry
```

Befehls festgelegt werden. Im Text werden diese dann mit

```
\gls{Begriff}
```

benutzt.

3.11 Listen und Aufzählungen

3.11.1 Listen

- ein wichtiger Punkt
- noch ein wichtiger Punkt
- und so weiter

3.11.2 Aufzählungen

1. Reihenfolge ist hier wichtig
2. Dieser Punkt kommt nach dem ersten
3. Da sollte jetzt eine 3 vorne stehen

3.11.2.1 Tiefste Ebene 1

Dies ist die tiefste Gliederungsebene. Sollten doch mehr Ebenen benötigt werden, muss eine andere Dokumentenklasse verwendet werden.

3.11.2.2 Tiefste Ebene 2

Der zweite Punkt in dieser Ebene ist zur Erinnerung daran, dass es nie nie niemals nur einen Unterpunkt geben darf.

3.12 Skript zum Kompilieren

Latex will ja bekanntlich in einer bestimmten Reihenfolge aufgerufen werden:

```
1 | lualatex thesis_main.tex
2 | biber thesis_main
3 | lualatex thesis_main.tex
4 | lualatex thesis_main.tex
5 | thesis_main.pdf
```

Dies ist der Inhalt der Batchdatei „compile.bat“.

3.13 PlantUML

```
1 | \begin{plantuml}
2 | @startuml
3 | Class01 <|-- Class02
4 | Class03 *-- Class04
5 | Class05 o-- Class06
6 | Class07 .. Class08
7 | Class09 -- Class10
8 | @enduml
9 | \end{plantuml}
```

4 Fazit

Wünsche Euch allen viel Erfolg für das 7. Semester und bei der Erstellung der Thesis. Über Anregungen und Verbesserung an dieser Vorlage würde ich mich sehr freuen.

Anhang

Anhang 1: Beispielanhang

Dieser Abschnitt dient nur dazu zu demonstrieren, wie ein Anhang aufgebaut sein kann.







Anhang 1.1: Weitere Gliederungsebene

Auch eine zweite Gliederungsebene ist möglich.

Anhang 2: Bilder

Auch mit Bildern. Diese tauchen nicht im Abbildungsverzeichnis auf.

Abbildung 4: Beispielbild

Name	Änderungsdatum	Typ	Größe
 abbildungen	29.08.2013 01:25	Dateiordner	
 kapitel	29.08.2013 00:55	Dateiordner	
 literatur	31.08.2013 18:17	Dateiordner	
 skripte	01.09.2013 00:10	Dateiordner	
 compile.bat	31.08.2013 20:11	Windows-Batchda...	1 KB
 thesis_main.tex	01.09.2013 00:25	LaTeX Document	5 KB

KI-generierte Quellen

Anthropic Claude 4 Sonnet (KI-generierte Inhalte, 2025): Erstellung einer tabellarischen Taxonomie-Übersicht für Literaturrecherche, claude.ai, Generiert am 02.08.2025, o. O., 2025-08, Anthropic Claude 4 Sonnet

Anhang KI: Dokumentation der KI-Prompts

Im Rahmen dieser Arbeit wurden KI-Tools zur Unterstützung bei verschiedenen Aufgaben eingesetzt. Gemäß den Richtlinien der FOM Hochschule werden hier alle verwendeten Prompts transparent dokumentiert.

KI.1 KI-Assistent - Tabellenerstellung

Verwendungszweck: Erstellung einer strukturierten Tabelle zur Übersicht der Forschungsmethodik

Prompt:

„Erstelle eine Übersichtstabelle für die Forschungsmethodik in LaTeX. Die Tabelle soll alle relevanten Kategorien zeigen und die für diese Arbeit gewählten Ansätze hervorheben. Stelle sicher, dass die Tabelle gut lesbar formatiert ist und alle notwendigen Informationen enthält.“

Gewählte Einordnung für diese Arbeit: - Forschungsansatz: Qualitativ-quantitativ - Datenerhebung: Systematische Literaturrecherche - Analysemethoden: Inhaltanalyse und Statistik - Zielgruppe: Akademische Forscher und Praktiker - Zeitrahmen: Repräsentative Quellen der letzten Jahre

Antwort:

Clear up
Appendi
Example
with ins-
tructions
from
Discord

Anhang B: Code-Listing - Beispiel für Datenverarbeitungsskript

Das folgende Python-Skript zeigt ein Beispiel für die automatisierte Verarbeitung und Analyse von Daten aus verschiedenen Dateiformaten. Das Skript demonstriert grundlegende Techniken der Datenextraktion, -analyse und -berichterstellung und kann als Vorlage für ähnliche Projekte angepasst werden.

Listing 1: data_processing_example.py - Beispiel für ein Datenverarbeitungsskript

```

1  #!/usr/bin/env python3
2  """
3  Beispiel für ein Datenverarbeitungsskript
4  Verarbeitet verschiedene Dateiformate und generiert Analyseberichte
5  """
6
7  import os
8  import re
9  import json
10 import csv
11 from collections import Counter, defaultdict
12 from pathlib import Path
13 from typing import List, Dict, Set, Tuple
14 import logging
15 from datetime import datetime
16
17 # Logging konfigurieren
18 logging.basicConfig(
19     level=logging.INFO,
20     format='%(asctime)s - %(levelname)s - %(message)s'
21 )
22 logger = logging.getLogger(__name__)
23
24 class DataProcessor:
25     """Verarbeitet Daten aus verschiedenen Quellen und generiert Analysen"""
26
27     def __init__(self, input_directory: str, output_directory: str):
28         self.input_directory = Path(input_directory)
29         self.output_directory = Path(output_directory)
30         self.processed_data = defaultdict(list)
31         self.analysis_results = {}
32
33     def load_data_from_files(self, file_pattern: str = "*.txt") -> Dict:
34         """Lädt Daten aus Dateien mit einem bestimmten Muster"""
35         data = {}
36         files = list(self.input_directory.glob(file_pattern))
37         logger.info(f"Found {len(files)} files to process")
38
39         for file_path in files:
40             try:
41                 with open(file_path, 'r', encoding='utf-8') as file:
42                     content = file.read()
43                     data[file_path.name] = content

```

```

44         logger.info(f"Loaded data from {file_path.name}")
45     except Exception as e:
46         logger.error(f"Error loading {file_path.name}: {e}")
47
48     return data
49
50 def extract_patterns(self, text: str, patterns: Dict[str, str]) -> Dict:
51     """Extrahiert Muster aus Text mit regulären Ausdrücken"""
52     results = {}
53
54     for pattern_name, pattern in patterns.items():
55         matches = re.findall(pattern, text, re.IGNORECASE)
56         results[pattern_name] = matches
57
58     return results
59
60 def analyze_data(self, data: Dict) -> Dict:
61     """Analysiert die geladenen Daten"""
62     analysis = {
63         'total_files': len(data),
64         'file_sizes': {},
65         'pattern_counts': {},
66         'summary_statistics': {}
67     }
68
69     # Dateigrößen analysieren
70     for filename, content in data.items():
71         analysis['file_sizes'][filename] = len(content)
72
73     # Muster definieren und zählen
74     patterns = {
75         'dates': r'\d{4}-\d{2}-\d{2}',
76         'emails': r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b',
77         'numbers': r'\b\d+\.\d*\b',
78         'words': r'\b[A-Za-z]+\b'
79     }
80
81     all_matches = defaultdict(list)
82     for filename, content in data.items():
83         for pattern_name, pattern in patterns.items():
84             matches = self.extract_patterns(content, {pattern_name: pattern})
85             all_matches[pattern_name].extend(matches[pattern_name])
86
87     # Muster zählen
88     for pattern_name, matches in all_matches.items():
89         analysis['pattern_counts'][pattern_name] = Counter(matches)
90
91     # Zusammenfassende Statistiken
92     analysis['summary_statistics'] = {
93         'total_words': len(all_matches['words']),
94         'unique_words': len(set(all_matches['words'])),
95         'total_numbers': len(all_matches['numbers']),
96         'total_dates': len(all_matches['dates']),
97         'total_emails': len(all_matches['emails'])
98     }
99
100    return analysis

```

```

101
102 def generate_report(self, analysis: Dict, output_file: str = None) -> str:
103     """Generiert einen Analysebericht"""
104     if output_file is None:
105         timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')
106         output_file = f"analysis_report_{timestamp}.json"
107
108     report = {
109         "analysis_date": datetime.now().isoformat(),
110         "input_directory": str(self.input_directory),
111         "output_directory": str(self.output_directory),
112         "results": analysis
113     }
114
115     # Bericht speichern
116     output_path = self.output_directory / output_file
117     self.output_directory.mkdir(exist_ok=True)
118
119     with open(output_path, 'w', encoding='utf-8') as f:
120         json.dump(report, f, indent=2, ensure_ascii=False)
121
122     logger.info(f"Report saved to {output_path}")
123     return str(output_path)
124
125 def export_to_csv(self, data: Dict, output_file: str = None) -> str:
126     """Exportiert Daten in CSV-Format"""
127     if output_file is None:
128         timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')
129         output_file = f"exported_data_{timestamp}.csv"
130
131     output_path = self.output_directory / output_file
132     self.output_directory.mkdir(exist_ok=True)
133
134     # Beispiel für CSV-Export
135     with open(output_path, 'w', newline='', encoding='utf-8') as csvfile:
136         fieldnames = ['filename', 'word_count', 'line_count', 'size_bytes']
137         writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
138         writer.writeheader()
139
140         for filename, content in data.items():
141             writer.writerow({
142                 'filename': filename,
143                 'word_count': len(content.split()),
144                 'line_count': len(content.splitlines()),
145                 'size_bytes': len(content.encode('utf-8'))
146             })
147
148     logger.info(f"CSV export saved to {output_path}")
149     return str(output_path)
150
151 def create_visualization_data(self, analysis: Dict) -> Dict:
152     """Bereitet Daten für Visualisierungen auf"""
153     viz_data = {
154         'file_size_distribution': {},
155         'pattern_frequency': {},
156         'timeline_data': []
157     }

```

```
158
159     # Dateigrößenverteilung
160     size_ranges = {'small': 0, 'medium': 0, 'large': 0}
161     for size in analysis['file_sizes'].values():
162         if size < 1000:
163             size_ranges['small'] += 1
164         elif size < 10000:
165             size_ranges['medium'] += 1
166         else:
167             size_ranges['large'] += 1
168
169     viz_data['file_size_distribution'] = size_ranges
170
171     # Musterhäufigkeit
172     for pattern_name, counts in analysis['pattern_counts'].items():
173         viz_data['pattern_frequency'][pattern_name] = {
174             'total_occurrences': sum(counts.values()),
175             'unique_values': len(counts)
176         }
177
178     return viz_data
179
180
181 def main():
182     """Hauptfunktion zur Ausführung der Datenverarbeitung"""
183     # Konfiguration
184     INPUT_DIR = "./input_data"
185     OUTPUT_DIR = "./output_results"
186
187     # Initialisiere Processor
188     processor = DataProcessor(INPUT_DIR, OUTPUT_DIR)
189
190     # Lade Daten
191     print("Loading data from files...")
192     data = processor.load_data_from_files("*.txt")
193
194     if not data:
195         print("No data files found. Please check the input directory.")
196         return
197
198     # Analysiere Daten
199     print("Analyzing data...")
200     analysis = processor.analyze_data(data)
201
202     # Generiere Bericht
203     print("Generating analysis report...")
204     report_file = processor.generate_report(analysis)
205
206     # Exportiere CSV
207     print("Exporting data to CSV...")
208     csv_file = processor.export_to_csv(data)
209
210     # Erstelle Visualisierungsdaten
211     print("Preparing visualization data...")
212     viz_data = processor.create_visualization_data(analysis)
213
214     # Zeige Zusammenfassung
```

```
215     print("\n" + "="*50)
216     print("DATA PROCESSING COMPLETE")
217     print("="*50)
218     print(f"Files processed: {analysis['total_files']}")
219     print(f"Total words: {analysis['summary_statistics']['total_words']}")
220     print(f"Unique words: {analysis['summary_statistics']['unique_words']}")
221     print(f"Numbers found: {analysis['summary_statistics']['total_numbers']}")
222     print(f"Dates found: {analysis['summary_statistics']['total_dates']}")
223     print(f"Emails found: {analysis['summary_statistics']['total_emails']}")
224
225     print(f"\nAnalysis report: {report_file}")
226     print(f"CSV export: {csv_file}")
227
228     return analysis
229
230
231 if __name__ == "__main__":
232     main()
```

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die angemeldete Prüfungsleistung in allen Teilen eigenständig ohne Hilfe von Dritten anfertigen und keine anderen als die in der Prüfungsleistung angegebenen Quellen und zugelassenen Hilfsmittel verwenden werde. Sämtliche wörtlichen und sinngemäßen Übernahmen inklusive KI-generierter Inhalte werde ich kenntlich machen.

Diese Prüfungsleistung hat zum Zeitpunkt der Abgabe weder in gleicher noch in ähnlicher Form, auch nicht auszugsweise, bereits einer Prüfungsbehörde zur Prüfung vorgelegen; hiervon ausgenommen sind Prüfungsleistungen, für die in der Modulbeschreibung ausdrücklich andere Regelungen festgelegt sind.

Mir ist bekannt, dass die Zuwiderhandlung gegen den Inhalt dieser Erklärung einen Täuschungsversuch darstellt, der das Nichtbestehen der Prüfung zur Folge hat und daneben strafrechtlich gem. § 156 StGB verfolgt werden kann. Darüber hinaus ist mir bekannt, dass ich bei schwerwiegender Täuschung exmatrikuliert und mit einer Geldbuße bis zu 50.000 EUR nach der für mich gültigen Rahmenprüfungsordnung belegt werden kann.

Ich erkläre mich damit einverstanden, dass diese Prüfungsleistung zwecks Plagiatsprüfung auf die Server externer Anbieter hochgeladen werden darf. Die Plagiatsprüfung stellt keine Zurverfügungstellung für die Öffentlichkeit dar.

Düsseldorf, 1.12.2025

(Ort, Datum)

A handwritten signature in black ink, consisting of a large, stylized 'H' followed by a series of loops and a final horizontal stroke.

(Eigenhändige Unterschrift)