



informatik +
digitale bildung

BACHELORARBEIT / MASTERARBEIT

Titel der Arbeit

„Titel“

vorgelegt von

Vorname Zuname

angestrebter akademischer Grad

Bachelor of Science (B.Sc.)

Karlsruhe, 1.1.2022

Studiengang
Studienfach
Matrikelnummer
Erstgutachter:
Zweitgutachter:

Bachelorstudiengang Lehramt Sekundarstufe I
Informatik
000000
Akademische(r) Grad(e) Vorname Zuname
Akademische(r) Grad(e) Vorname Zuname

Danksagung

Vielen Dank!

Abstract

This is an English abstract of the written work.

Kurzfassung

Das ist eine deutsche Kurzfassung der verfassten Arbeit.

Inhaltsverzeichnis

Danksagung	i
Abstract	ii
Kurzfassung	iii
Tabellenverzeichnis	v
Abbildungsverzeichnis	vi
Liste von Algorithmen	vii
Listings	viii
1. Abschnitt 1	1
1.1. Überschrift: Test	1
1.1.1. Unterkapitel 1	1
2. Code	2
3. Algorithmen	3
4. Tabellen und Bilder	4
4.1. Tables	4
4.2. Images	4
5. Abkürzungen und Glossar	5
Literaturverzeichnis	6
A. Anhang	9

Tabellenverzeichnis

4.1. some description of the table	4
--	---

Abbildungsverzeichnis

4.1. Image Example	4
------------------------------	---

List of Algorithms

1. How to write algorithms	3
--------------------------------------	---

Listings

2.1. Python example	2
-------------------------------	---

1. Abschnitt 1

1.1. Überschrift: Test

Seit Wing (2006) den Begriff *Computational Thinking* einführte, ist dieser Gegenstand der informatikdidaktischen Forschung (Denning and Tedre, 2022).

1.1.1. Unterkapitel 1

2. Code

If you want to show program code within your thesis you can use the `\texttt{verbatim}` environment or for a more complex display take a look at https://www.overleaf.com/learn/latex/Code_listing

Text enclosed inside `\texttt{verbatim}` environment is printed directly and all `\LaTeX{}` commands are ignored.

```
1 import numpy as np
2
3 def incmatrix(genl1,genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r,c] = np.where(M2 == M1[i,j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
20                VT[(j)*n + c[k]] = 1;
21
22            if M is None:
23                M = np.copy(VT)
24            else:
25                M = np.concatenate((M, VT), 1)
26
27            VT = np.zeros((n*m,1), int)
28
29    return M
```

Listing 2.1: Python example

3. Algorithmen

If you want to show algorithms in your Thesis take a look at the <https://www.overleaf.com/learn/latex/algorithms> page. The `algorithm2e` package is already included in the template. You can list algorithms in the same way as you can list Tables and Figures.

Data: this text

Result: how to write algorithm with $\text{\LaTeX}2\text{e}$ initialization;

```
while not at end of this document do  
  read current;  
  if understand then  
    go to next section;  
    current section becomes this one;  
  else  
    go back to the beginning of current section;  
  end  
end
```

Algorithm 1: How to write algorithms

4. Tabellen und Bilder

One of the great advantages of \LaTeX is that all it needs to know is the structure of a document, and then it will take care of the layout and presentation itself. So, here we shall begin looking at how exactly you tell \LaTeX what it needs to know about your document.

4.1. Tables

In this sub-section, a simple table is inserted. To add reference to the table, see (cf. Table 4.1):

Command	Level
<code>\part{part}</code>	-1
<code>\chapter{chapter}</code>	0
<code>\section{section}</code>	1
<code>\subsection{subsection}</code>	2
<code>\subsubsection{subsubsection}</code>	3
<code>\paragraph{paragraph}</code>	4
<code>\subparagraph{subparagraph}</code>	5

Tabelle 4.1.: some description of the table

4.2. Images



informatik +
digitale bildung

Abbildung 4.1.: Image Example

When an image is inserted, you can refer to it like this (cf. Figure ??).

5. Abkürzungen und Glossar

if you want to use Acronyms or a Glossary check the page here: <https://www.overleaf.com/learn/latex/glossaries>

The Latex typesetting markup language is specially suitable for documents that include mathematics. are rendered properly an easily once one gets used to the commands.

Given a set of numbers, there are elementary methods to compute its Greatest Common Divisor, which is abbreviated GCD. This process is similar to that used for the Least Common Multiple (LCM).

Literaturverzeichnis

Denning, P. J. and Tedre, M. (2022). Computational thinking: A disciplinary perspective. *Informatics in Education*, 20(3):361–390.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.

Acronyms

GCD Greatest Common Divisor. 5

LCM Least Common Multiple. 5

Glossar

latex Is a mark up language specially suited for scientific documents. 5

mathematics Mathematics is what mathematicians do. 5

A. Anhang

here you can put further things you want to add like transcripts, questionnaires, raw data...