Leave The House - Checklisten App

Studienarbeit

im Rahmen der Prüfung zum

Bachelor of Science (B.Sc.)

des Studienganges Informatik

an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

Marius Huber

April 2021

Abgabedatum:	17. Mai 2021
Bearbeitungszeitraum:	01.10.2020 - 17.05.2021
Matrikelnummer, Kurs:	1286628, TINF18B2
Ausbildungsfirma:	SAP SE Dietmar-Hopp-Allee 16 69190 Walldorf, Deutschland
Gutachter der Dualen Hochschule:	Dr. Christian Bomhardt

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema:

Leave The House - Checklisten App

gemäß § 5 der „Studien- und Prüfungsordnung DHBW Technik“ vom 29. September 2017 selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, den 16. April 2021

Huber, Marius

Abstract

- English -

This is the starting point of the Abstract. For the final bachelor thesis, there must be an abstract included in your document. So, start now writing it in German and English. The abstract is a short summary with around 200 to 250 words.

Try to include in this abstract the main question of your work, the methods you used or the main results of your work.

Abstract

- Deutsch -

Dies ist der Beginn des Abstracts. Für die finale Bachelorarbeit musst du ein Abstract in deinem Dokument mit einbauen. So, schreibe es am besten jetzt in Deutsch und Englisch. Das Abstract ist eine kurze Zusammenfassung mit ca. 200 bis 250 Wörtern.

Versuche in das Abstract folgende Punkte aufzunehmen: Fragestellung der Arbeit, methodische Vorgehensweise oder die Hauptergebnisse deiner Arbeit.

Inhaltsverzeichnis

Formelverzeichnis

A	mm^2	Fläche
D	mm	Werkstückdurchmesser
$d_{\mathbf{min}}$	mm	kleinster Schaftdurchmesser
$L_{\mathbf{1}}$	mm	Länge des Werkstückes Nr. 1
	Grad	Freiwinkel
	Grad	Keilwinkel

Abkürzungsverzeichnis

AIR	Adobe Integrated Runtime
AJAX	Asynchronous Javascript and XML
ANSI	American National Standards Institute
API	Application Programming Interface
AR	Augmented Reality
BAPI	Business Application Programming Interface
BIOS	Basic Input Output System
CDMA	Code Division Multiple Access
HTTPS	Hypertext Transfer Protocol Secure
ISBN	Internationale Standardbuchnummer
OLAP	Online Analytical Processing
ORDBMS	Object-Relational DataBase Management System
SDK	Software Development Kit
SEO	Search Engine Optimization
SSH	Secure Shell
UEFI	Unified Extensible Firmware Interface
USB	Universal Serial Bus
VLAN	Virtual Local Area Network
WYSISWG	What You See Is What You Get
XSL	Extensible Stylesheet Language

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

1 Einleitung

Information bezüglich des Inhaltsverzeichnisses

Nachtrag zum Inhaltsverzeichnis: Dieses sollte wenn möglich nur eine Seite lang sein. Unterpunkte können dabei auch ausgelassen werden. Dies kann ganz einfach durchgeführt werden indem die section mit einen Stern geschrieben wird, wie bei der Sektion hier.

Des Weiteren solltest du beachten, dass du keine Unterpunkte alleine aufmachen darfst. Laut den DHBW Leitlinien sollte, wenn es einen Punkt 1.1 gibt, auch ein Punkt 1.2 existieren. Weitere Details dazu in den Leitlinien - zur Info: diese Vorlage hält sich aktuell nicht an diese Regelung, sie ist schließlich nur ein Template.

1.1 Grobe Struktur der Arbeit

Alle Details zur Struktur findest du in den Leitlinien ab Seite 21. Nachfolgend ist nur die verkürze Version mit allen wichtigen Punkten angegeben.

1.1.1 Einleitung - was gibt es zu beachten?

Die Einleitung sollte folgende Punkte beinhalten:

- **Gegenstand und Ziele der Arbeit & Aufgabenbeschreibung, Einführung in Thema, Stand der Technik & Forschung, Motivation der Aufgabenstellung & Vorausblick**
- Ausgangspunkt der Arbeit umreisen
- Hinführung zur Problemstellung + Interesse des Lesers wecken
- Allgemeine Einleitung ins Thema (keine Unternehmens-, Produktbeschreibungen oder Organigramme!)

- Fragestellung präsentieren, Motivation erläutern
- Randbedingungen und Betrachtungsgrenzen aufzeigen
- Stand der Technik und aktuelle Lösungsfindung beschreiben, Vor- und Nachteile bisheriger Lösung anhand von Literatur darlegen

1.1.2 Hauptteil - zeige was du kannst!

Folgende Punkte kannst du in deinen Hauptteil einarbeiten:

- **Anforderungsdefinition, Anforderungsanalyse, Lösungsgenerierung, Lösungsbewertung, Umsetzung) in sinnvollen Gliederungspunkten**
- Gewähltes Verfahren oder bestimmter Lösungsweg muss begründet werden
- Bei Versuchen (nicht alle müssen genannt werden) müssen Voraussetzungen, Vernachlässigungen sowie Anordnung, Leistungsfähigkeit und Messgenauigkeit der einzelnen Versuchsanordnung angeben werden
- Ergebnisse der Arbeit ausführlich diskutieren, Vergleiche mit Anschauungen und Erfahrungen vergleichen
- Ziel der Arbeit: Eindeutige Folgerungen und Richtlinien für die Praxis

1.1.3 Schluss - das Ziel ist nahe...

Nachfolgende Aspekte können in den Schluss eingearbeitet werden:

- **Zusammenfassung und Ausblick**
- Aufgabenstellung, Vorgehensweise, sowie wesentliche Ergebnisse kurz/präzise darstellen
- Zusammenfassung eigenständig verständlich
- Länge ca. 1-1,5 Seiten (Problem, Ziele, Vorgehensweise, Ergebnisse und Ausblick)

1.1.4 Sonstige Tipps

Es ist hilfreich, sich Notizen in das Dokument zu schreiben. LaTeXKommentare haben jedoch den Nachteil, dass sie in der PDF nicht erscheinen. Einfacher Text wird auch leicht überlesen. Deshalb wird in dieser Vorlage das Package `todonotes` eingebunden, welches Notizen in dem Dokument sichtbar macht. Auf der rechten Seite siehst du ein Beispiel.

So wie hier

2 Textformatierungen

2.1 Definitionen und Highlight Boxen

> **Definition:** Diese Hervorhebungen können für deine Arbeit an machen stellen sehr nützlich sein. Besonders bei Definitionen macht es einen guten Eindruck, wenn diese in solch einer Form dargestellt ist.

DHBW Richtlinie: Laut den aktuellen Angaben der DHBW sind diese Boxen nicht notwendig. Helfen können sie jedoch, um einen Faktor speziell hervorzuheben. Bitte beachte, dass deine Projektarbeit oder auch Bachelorarbeit kein Bilderbuch ist! Alles was eingebunden wird sollte schlicht und dezent dargestellt sein.

> **Wichtig:** Verwende kein „ich“, während der gesamten Arbeit. Jeder weiß, dass es deine Arbeit ist. Auch von Sätzen mit „man“, solltest du Abstand nehmen. Frage deinen Betreuer gerne, welche Vorzüge er oder sie hat. Jeder Dezi oder DHBW-Betreuer hat in diesem Zusammenhang unterschiedliche Meinungen.

2.2 Schriftbild

LARGE Text small Text normal Text

Fetter Text GROSSBUCHSTABEN *Kursiver Text*

`Schreibmaschinenschrift` serifenlose Schrift Serifenschrift

Manche Zeichen wollen einfach nicht so, wie der Autor das will: % & $ { }

2.3 Listen

- Erster geworden
  - Wir sind nicht oben oder?
  + Nummerierungen oder
  - Aufzählungen oder
  * Definitionen

1. Jetzt ist es offiziell ich bin die Nummer 1
42. Es ist die Antwort auf alles

Epidemie / Pandemie
: Als Epidemie bezeichnet man eine in einem bestimmten begrenzten Verbreitungsgebiet auftretende ansteckende Erkrankung; eine Seuche, für die typisch ist, dass eine große Zahl von Menschen gleichzeitig befallen wird.

2.4 Abkürzungen

Während du schreibst benötigt du zu manchen Zeitpunkten einfach ein paar Abkürzungen. Doch wie mache ich das wenn ich eine Abkürzung für die Application Programming Interfaces (APIs) nutzen möchte. Ich könnte auch nur ein API gemeint haben. Daneben gibt es noch Hypertext Transfer Protocol Secure (HTTPS) oder Asynchronous Javascript and XML (AJAX). Willst du einen Begriff nochmals ausschreiben, dann verwende Hypertext Transfer Protocol Secure (HTTPS) einfach als Kommando.

Du kannst auch den Plural von Abkürzungen verwenden. Dazu einfach ein `p` an den jeweiligen Befehl anhängen, z.B. Internationale Standardbuchnummern (ISBNs).

Im Text können gewisse Dinge auch nützlich sein wie z. B. diese Abkürzung, d. h. du kannst sie so direkt in den Text eintragen. Die Namen kann jeder selbst festlegen.

2.5 Anführungszeichen

Normale Anführungszeichen (¨) können in LaTeX nicht verwendet werden. Dafür muss das entsprechende Wort in `\enquote{...}` gesetzt werden. Beispiel: „Ich stehe ich Anführungszeichen. ‚Schachtelungen funktionieren auch.‘“

Alternativ können Anführungszeichen auch von Hand gesetzt werden: `\glqq{}` entspricht „ und `\grqq{}` entspricht “

2.6 Verweise und URLs

URLs können mit dem Package „hyperref“ und den Befehlen `href` und `url` dargestellt werden. Beispiele:

- Mit eigenem Text: **Klick mich**, um diese Vorlage auf Github zu sehen!
- Anzeigen der URL: **https://www.google.com/**

Verweise sind eines der wichtigsten Werkzeuge von LaTeX. Mit dem Package „hyperref“ gibt es verschiedene Verweise, die in Tabelle 2.1 gelistet sind.

`\ref`	Zeigt die Nummer	Bsp.: 2.6
`\autoref`	Zeigt Typ + Nr.	Bsp.: Abschnitt 2.6
`\autopageref`	Zeigt „Seite Seitennummer“	Bsp.: Seite 6
`\nameref`	Zeigt den Namen (bzw. Caption)	Bsp.: Verweise und URLs
`\hyperref`	Eigener Text	Bsp.: Klick mich!

Tabelle 2.1: Verweise im Dokument

3 Abbildungen

Alle Abbildungen und Tabellen sind laut Richtlinien fortlaufend mit Nummern zu versehen. Diese Aufgabe übernimmt LaTeX für dich. Jedoch solltest du den Text unter dem Bild (Legende) auf das jeweilige Bild anpassen. Hast du das Bild irgendwo entnommen, dann muss der Quellenverweis auch direkt in der Legende mit eingebaut sein.

Im Text selbst solltest du auf die Abbildung verweisen. Neben dem reinen verweisen, solltest du dich auch damit auseinandersetzen. Beschreibe die Grafik, hebe die Relevanz einzelner Teile hervor oder nenne andere wichtige Informationen im Text davor oder danach.

3.1 Tabellen

Die Legende steht bei den Tabellen darüber, während diese bei anderen Grafiken darunter ihren Platz einnimmt.

Tabelle 3.1: Eine dreispaltige Tabelle

linke Spalte	**mittlere Spalte**	**rechte Spalte**
A	B	C
!	2	3
a	b	c
i	ii	iii

Bitte beachte auch, dass LaTeX deine Tabellen an eine andere Position verschiebt, wenn diese dort besser aussehen. Vermeide deshalb Textpassagen wie beispielsweise „in der nachfolgenden Tabelle/Grafik“, denn es könnte sein, dass die Tabelle vor den Text rutscht. Nutze deshalb immer Verweise wie: „in der Tabelle 3.2 ist . . . “!

Die Tabelle 3.2 nutzt mehrere Packages. Mithilfe des Befehls `\midrule` aus dem Package „booktabs“ erstellt man eine horizontale Linie, die die vertikalen unterbricht. Diese sorgt

für ein schöneres Schriftbild. Möchtest du Zahlen auflisten, so kannst du diese nach dem Dezimalkomma ausrichten. Dies geht mit dem Package „siunitx“.

Tabelle 3.2: Zahlenausrichtung

Nr.	Datum	Euro	USD	Zahlen
1	01.06.2017	1,00€	1,13$	11,158
2	02.06.2017	2,00€	2,26$	2,18
3	03.06.2017	3,00€	3,39$	9,155 68
4	04.06.2017	4,00€	4,52$	5,868 668
5	05.06.2017	5,00€	5,65$	1,4
6	06.06.2017	6,00€	6,78$	6,58
7	07.06.2017	7,00€	7,91$	7,998
8	08.06.2017	8,00€	9,04$	4,358
9	09.06.2017	9,00€	10,17$	3,5458
10	10.06.2017	10,00€	11,30$	302,8

Die Tabelle 3.3 zeigt die Möglichkeit, einzelne Zellen miteinander zu verbinden. Mit dem Befehl `\multicolumn {Anzahl Spalten}{Ausrichtung}{Inhalt}` kannst du mehrere Spalten verbinden. Dafür müssen die überschriebenen Spalten entfernt werden, es darf also keinen „&“-Trennzeichen dafür geben. Beim Verbinden mehrerer Reihen wird der Befehl `\multirow{Anzahl Reihen}{*}{Inhalt}` verwendet. Hierbei darauf achten, dass die Zellen, die zusammengefasst werden sollen, keinen Inhalt haben, aber vorhanden sind, also einen „&“-Trennzeichen besitzen.

Tabelle 3.3: Verbinden von Zellen

Text 1	Mittiger Text 2	Text 3
Linksbündig		1,00 €
2		2,00 €
3	03.06.2017	3,00 €
4	04.06.2017	4,00 €
		Rechtsbündiger Text
6	06.06.2017	6,00 €
7	Zusammen	7,00 €
8		8,00 €
9	09.06.2017	9,00 €
10	10.06.2017	10,00 €

3.2 Grafiken

Abbildung 3.1:
Wrap Figure mit einer PDF als Grafik

Eine gute Projektarbeit kommt nicht ohne einige Abbildungen in Form von Skizzen, Diagrammen oder ähnlichem aus. Am besten ist es, wenn du diese Grafiken selbst als SVG Dateien erstellst und diese in Form eines PDFs einbindest, somit ist die Grafik auch beim Drucken scharf.

Gerade solche Kleinigkeiten zeugen von Professionalität und können auch nochmals einige Punkte für eine gute Note raus holen. Die `figure` Umgebung eignet sich für das Einbinden von Grafiken, die die volle Seitenbreite ausnutzen. Sie erscheinen nicht im Fließtext.

Dagegen gibt es die `wrapfigure` Umgebung, die das Einbinden von Grafiken im Fließtext erlaubt. Diese Umgebung ist allerdings ab und zu problematisch zu benutzen. Tipp: Die `wrapfigure` Umgebung vor dem Absatz platzieren und darauf achten, dass sie nicht in der Nähe eines Seitenumbruchs ist. Dann erscheint sie rechts oder links des Paragraphen. Mit `vspace` kann noch der Abstand nach oben und unten angepasst werden, um

leeren Platz zu vermeiden. Siehe auch https://tex.stackexchange.com/questions/56176/handling-of-wrapfig-pictures-in-latex.

Abbildung 3.2: Desktop Ansicht mit einem Symbol

Das kostenlose Tool **Inkscape** hilft dir beim erzeugen dieser SVG Grafiken. Solltest du noch nie damit gearbeitet haben, dann schau dir am besten einige kurze Tutorials an. Im Github Repository unter dem Reiter Wiki findest du eine Kurzanleitung, wie du ein PDF in Inkscape generieren kannst. Solltest du weitere gute kostenlose Software für die Bildbearbeitung in SVG kennen, dann kannst du diese gerne uns per E-Mail mitteilen. Danke!

3.3 Diagramme, Mockups, Software Design

Um Software Designs mit einzubinden eignet sich das Online Tool **DRAW.IO** sehr gut. Mit diesem Tool lassen sich Diagramme, Mockups, Flow Charts, technische Zeichnungen, Wireframe Skizzen sowie Software Designs zeichnen, welche du ohne Verpixelung im PDF-Format wieder, wie in Abbildung 3.3 gezeigt, einbinden kannst.

Ein weiteres kostenloses Tool für die Darstellung von Klassendiagrammen oder auch Sequenzdiagrammen ist **StarUML**. Dieses generiert auch SVG Dateien, welche du mithilfe von Inkscape zu PDF Dateien konvertieren kannst.

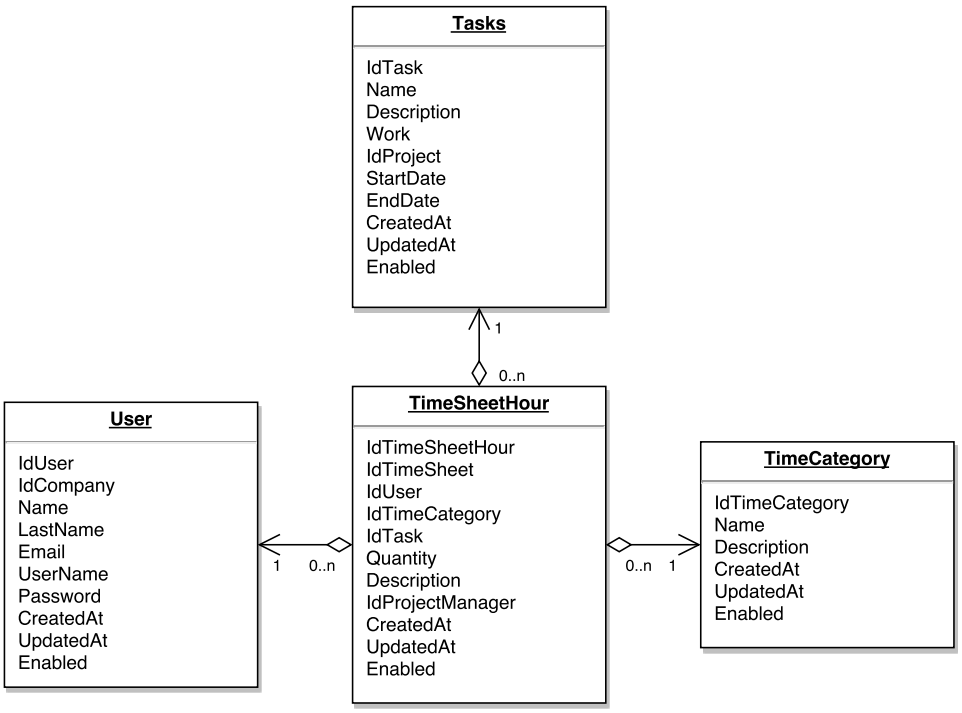

Abbildung 3.3: Diagramm eines Software Designs

4 Mathematik

4.1 Text

Hier steht ein beispielhafter Text bei dem nun auf eine sehr bekannte und durchaus vertraute Formel im Text direkt mit $c = \sqrt{a^2 + b^2}$ eingegangen wird. Dabei können auch Winkel wie: α, β, γ gerne verwendet werden. Weiterhin werden hier nur Formeln im Bereich der $\mathbb{N}$ dargestellt, gerne können diese aber durch Formeln aus diesem Bereich der $\mathbb{R}$ ergänzt werden.

4.2 Formeln

Beachte bei Formeln keine konkreten Werte anzugeben, sondern die Formel stets nur wie in der Literatur nur als Größengleichungen anzugeben.

$$\sum_{n=0}^{\infty} x = b + n \tag{4.1}$$

$$\frac{b * x}{c} = y \tag{4.2}$$

Trotz unterschiedlicher Länge kann man die Gleichheitszeichen auf der gleichen Höhe anbringen wie in Gleichung (4.3) und Gleichung (4.4) dargestellt, zusätzlich kann man diese auch mit Informationen versehen wie in Formel Gleichung (4.5) zu sehen.

$$a + b = c \tag{4.3}$$

$$5c + 3f = 4h \tag{4.4}$$

$$\overbrace{5y}^{y=0} + \underbrace{42x}_{x=1} = b \tag{4.5}$$

4.3 Arrays und Matrizen

$$\begin{array}{cc|c} a & b & c \\ \hline x & y & z \\ c & a & b \end{array}$$

$$\begin{aligned} z &= a \\ a+b &= c \\ f(x,y,z) &= x+y+z \end{aligned} \tag{4.6}$$

Hier noch ein paar Matrizen Beispiele in LaTeX.

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \tag{4.7}$$

$$\begin{bmatrix} 100 & 250 \\ 300 & 499 \end{bmatrix} \tag{4.8}$$

$$\begin{Bmatrix} 100 & 250 \\ 300 & 499 \end{Bmatrix} \tag{4.9}$$

$$\begin{Vmatrix} 100 & 250 \\ 300 & 499 \end{Vmatrix} \tag{4.10}$$

4.4 Klammern und Kästchen

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$$

$$\left.\begin{array}{cc|c} a & b & c \\ \hline x & y & z \\ c & a & b \end{array}\right\} \Rightarrow z, b$$

$$\boxed{\sin^2\varphi + \cos^2\varphi = 1}$$

5 Programm- bzw. Quellcode

5.1 Quellcode

Ein wichtiger Punkt ist auch, dass man Quellcode Stücke mit in seinen Praxisbericht einbaut. Hier nun einfach mal ein Beispiel in Form eines kleinen JAVA Codes, welcher aus einer Datei gelesen wird:

```
1  public class CrossTotal {
2      static String nutzerEingabe;
3      static int ergebnis = 0;
4
5      // Methode berechnet die Quersumme der Zahl
6      public static void main(String[] args) {
7          Scanner scan = new Scanner(System.in);
8          System.out.println("Bitte Zahl eingeben:");
9          nutzerEingabe = scan.next();
10         scan.close();
11         for (char c : nutzerEingabe.toCharArray()) {
12             ergebnis = ergebnis + Character.getNumericValue(c);
13         }
14         System.out.println("Die Quersumme von " + ↙
               ↳ nutzerEingabe + " ist " + ergebnis);
15     }
16 }
```

Listing 5.1: Algorithmus zur Berechnung der Quersumme

Zu beachten ist, dass jedes Stück Code kommentiert werden sollte. Was wird in diesem Abschnitt genau durchgeführt. Wo könnten Probleme auftreten und warum wurde dieses Stück hier hinzugefügt.

EVENT 01 *INIT*

```
1       TRY.
2         DATA(test)   = |Hallo|.
3         DATA(test_2) = `Hallo`.
4         "Get global instance of Web GUI
5         go_fti_webgui_sscui = ↙
              ↳ /fti/cl_webgui_sscui=>create_instance(
6               iv_activity_id   = 'SIMG_CFMENUOLMROMRG'
7               iv_object_name   = x_header-viewname
8               iv_cust_obj_type = vim_view
9         ).
10        "Check whether content change is allowed for current Web ↙
              ↳ GUI
11        IF go_fti_webgui_sscui->is_obj_edit_allowed(
12              iv_object_id          = x_header-viewname
13              iv_object_type        = vim_view
14        ) = abap_false.
15
16          view_action = anzeigen.
17        ENDIF.
18      CATCH /fti/cx_webgui_error INTO DATA(lx_webgui_error).
19          "Do something here
20      ENDTRY.
```

Listing 5.2: Initialisierung im Programm

Es ist auch möglich, innerhalb des Listings LaTeX Befehle zu verwenden. Dazu muss aber eine Escape-Sequenz angegeben werden. Das folgende Beispiel enthält ein Label, auf das dann verwiesen werden wird: Auch hier funktioniert `\autoref`: „In Zeile 2 wird der Variable `b` ...“.

```
1 int a = 10;
2 int b = a + 20;
3 return;
```

Listing 5.3: Zuweisung von Variablen

Weitere Programmiersprachen können auch eingebunden werden. Hier mal ein Beispiel in der Programmiersprache Python:

```
1  import random
2
3  guesses_made = 0
4  name = raw_input('Hello! What is your name?\n')
5
6  number = random.randint(1, 20)
7  print 'Well, {0}, I am thinking of a number between 1 and 20.'.format(name)
8
9  while guesses_made < 6:
10    guess = int(raw_input('Take a guess: '))
11    guesses_made += 1
12    if guess < number:
13       print 'Your guess is too low.'
14    if guess > number:
15       print 'Your guess is too high.'
16    if guess == number:
17       break
18
19 # Show overall status after several trys:
20 if guess == number:
21    print 'Good job, {0}! You guessed my number in {1} guesses!'.format(name, guesses_made)
22 else:
23    print 'Nope. The number I was thinking of was {0}'.format(number)
```

Listing 5.4: Algorithmus zum schätzen einer Zahl in Python

5.2 Pseudocode

Pseudocode kann hilfreich sein, wenn „richtig“ implementierte Algorithm in einer Programmiersprache zu lang sind und diese mittels Pseudocode bündig zusammengefasst werden können. Algorithmus 1 zeigt Pseudocode.

Die Doku für das Package und damit eine Liste aller Befehle findet sich unter
http://tug.ctan.org/macros/latex/contrib/algorithmicx/algorithmicx.pdf

Algorithmus 1 Euclid's algorithm

```
1: procedure EUCLID(a, b)                          ▷ The g.c.d. of a and b
2:     r ← a mod b
3:     while r ≠ 0 do                              ▷ We have the answer if r is 0
4:         a ← b
5:         b ← r
6:         r ← a mod b
7:     end while
8:     return b                                    ▷ The gcd is b
9: end procedure
```

6 Literaturhinweise

Hinweis: Verwendest du eine Quelle nicht, dann nimmt LaTeX diese nicht mit ins Literaturverzeichnis auf!

6.1 Fußnoten und Literaturverweise

Hamburger, Döner, Currywurst[1] - jeder kennt sie, jeder liebt sie und jeder isst sie. Weil die Zeit drängt[**Bonnen.2016**], der Hunger groß ist und der nächste Schnellimbiss[**Forsthuber.2016**] nur drei Schritte voraus. Und nach dem Essen? Sind wir zwar satt, aber meist nicht wirklich glücklich, weil Fastfood[**Friedl.2017**] meist eben auch nicht wirklich gut ist [**Kuhnlein.2016**].

Ja, uns ist bewusst, dass die Literatur nicht zum Text passt. Deswegen hier nochmals der Rest.[**Mukherjee.2017**, **Preuss.2017**, **Schell.2017**, **Smith.2017**, **Visser.2017**, **Zaidi.2017**, **o.V..o.J.**]

Übrigens: Internationale Standardbuchnummern (ISBNs) gehören nicht in die Bibliographie und werden deshalb auch nicht angezeigt.

[1] Hier fehlt eindeutig das Lieblingsessen der Informatiker, die Pizza!