



Template \LaTeX Wiki von BAzubis für BAzubis

Projektarbeit 1 (T3_1000)

im Rahmen der Prüfung zum
Bachelor of Science (B.Sc.)

des Studienganges Informatik
an der Dualen Hochschule Baden-Württemberg Karlsruhe

von

Yannik Schiebelhut

Abgabedatum:	04. Oktober 2021
Bearbeitungszeitraum:	01.10.2020 - 03.10.2021
Matrikelnummer, Kurs:	3354235, TINF20B1
Ausbildungsfirma:	SAP SE Dietmar-Hopp-Allee 16 69190 Walldorf, Deutschland
Betreuer der Ausbildungsfirma:	Helge Dickel
Gutachter der Dualen Hochschule:	DH-Vorname DH-Nachname

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit 1 (T3_1000) mit dem Thema:

Template L^AT_EX Wiki von BAzubis für BAzubis

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 11. Dezember 2020

Schiebelhut, Yannik

Inhaltsverzeichnis

Formelverzeichnis	IV
Abkürzungsverzeichnis	V
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VII
Quellcodeverzeichnis	VIII
1 Einleitung	1
1.1 Was ist der ewm-sim?	1
1.2 Kritikpunkte der ursprünglichen Implementierung	1
2 Theoretische Grundlagen	3
2.1 Docker	3
2.2 Node.js	4
2.3 Postman	5
2.4 Kubernetes / Google Cloud Platform	5
2.5 Git	5
2.6 Travis-CI	5
2.7 Jira	5
3 Textformatierungen	6
3.1 Definitionen und Highlight Boxen	6
3.2 Schriftbild	6
3.3 Listen	7
3.4 Abkürzungen	7
3.5 Anführungszeichen	8
3.6 Verweise und URLs	8
4 Abbildungen	9
4.1 Tabellen	9
4.2 Grafiken	11
4.3 Diagramme, Mockups, Software Design	12
5 Mathematik	14
5.1 Text	14
5.2 Formeln	14

5.3	Arrays und Matrizen	15
5.4	Klammern und Kästchen	15
6	Programm- bzw. Quellcode	16
6.1	Quellcode	16
6.2	Pseudocode	19
7	Literaturhinweise	20
7.1	FuSSnoten und Literaturverweise	20
	Literaturverzeichnis	IX

Formelverzeichnis

A	mm ²	Fläche
D	mm	Werkstückdurchmesser
d_{\min}	mm	kleinster Schaftdurchmesser
L_1	mm	Länge des Werkstückes Nr. 1
	Grad	Freiwinkel
	Grad	Keilwinkel

Abkürzungsverzeichnis

AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
EWM	Extended Warehouse Management
HTTPS	Hypertext Transfer Protocol Secure
npm	Node Package Manager
SDK	Software Development Kit
SSH	Secure Shell
WSL	Windows-Subsystem für Linux

Abbildungsverzeichnis

1.1	Aufbau des ewm-sim v1	2
4.1	Wrap Figure mit einer PDF als Grafik	11
4.2	Desktop Ansicht mit einem Symbol	12
4.3	Diagramm eines Software Designs	13

Tabellenverzeichnis

3.1	Verweise im Dokument	8
4.1	Eine dreispaltige Tabelle	9
4.2	Zahlenausrichtung	10
4.3	Verbinden von Zellen	11

Quellcodeverzeichnis

6.1	Algorithmus zur Berechnung der Quersumme	16
6.2	Initialisierung im Programm	17
6.3	Zuweisung von Variablen	17
6.4	Algorithmus zum schätzen einer Zahl in Python	18

1 Einleitung

1.1 Was ist der ewm-sim?

In der vierten industriellen Revolution verändert sich auch der Arbeitsalltag in Lagerhallen. Mobile Roboter finden verstärkt Einsatz, um die Arbeiter zu unterstützen. Das Projekt Extended Warehouse Management (EWM) Cloud Robotics der SAP hat das Ziel, ein Roboternetzwerk auf Basis von Google Cloud Robotics an ein SAP EWM-System anzubinden. Zu Demonstrationszwecken wird eine Simulationsumgebung erstellt, in der ein virtuelles Warenlager präsentiert werden, in dem Roboter beispielhafte Aufträge bearbeiten. Um nun zu vermeiden, dass ein vollständiges EWM-System für solch eine Simulation deployed werden muss, wurde „ewm-sim“ eingeführt. Er stellt einen kleinen Web-Server dar, welcher die Schnittstelle, über die die Roboter ihre Aufträge vom EWM-System erhalten, detailgetreu nachbildet.

1.2 Kritikpunkte der ursprünglichen Implementierung

Wie bereits erwähnt, soll der ewm-sim die Schnittstelle eines EWM-Systems nachbilden. Hierbei handelt es sich um einen OData-Service. Für die Implementierung wurde hier auf den bestehenden MockServer von SAPUI5 gesetzt. Dieser ist jedoch zur Frontend- und nicht zur Backend-Entwicklung vorgesehen. Leider bringt er das Problem mit sich, dass er sich nur innerhalb einer SAPUI5-App verwenden lässt, welche wiederum in einem Browser läuft. In der Abbildung 1.1 ist der Aufbau des bisherigen ewm-sim veranschaulicht. Er stellt eine Art gekapseltes System dar. Die Anfragen, die gegen den nach außen hin geöffneten WebServer geschickt werden, landen über den WebSocket Server bei einer headless Instanz von Google Chrome, in welchem wiederum die SAPUI5-App ausgeführt wird. Diese ist mit dem SAPUI5 Mockserver verknüpft, welcher die Daten, die er bereitstellen soll, aus json Dateien einliest.

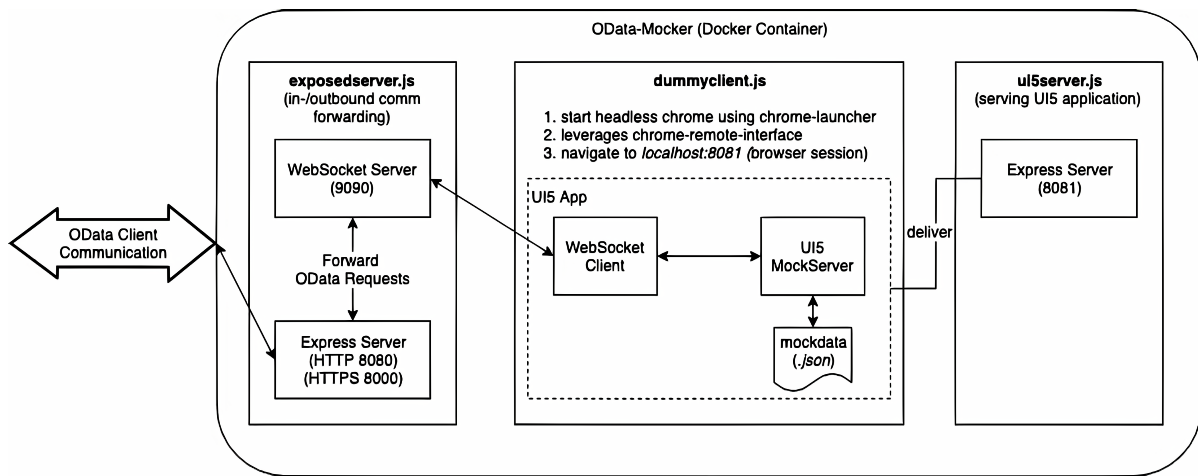


Abbildung 1.1: Aufbau des ewm-sim v1

Wie gut zu erkennen ist, bringt diese Implementierung einen groSSen Overhead und mögliche Fehlerquellen mit sich. Ziel des in dieser Praxisarbeit behandelten Projekts soll es sein, das Konzept des ewm-sim zu optimieren und diesen neu zu implementieren.

2 Theoretische Grundlagen

Zur Bearbeitung dieses Projekts wurden einige Technologien eingesetzt, für die hier zunächst die theoretischen Grundlagen erläutert werden sollen.

2.1 Docker

Docker ist eine freie Implementierung des Konzepts der Containervirtualisierung. Ein sogenannter Docker Container ist plattformübergreifend lauffähig. Voraussetzung ist lediglich, dass der Docker Daemon auf dem Host-System installiert ist. Damit die Container, wie beschrieben, auf allen Betriebssystemen laufen können, setzen sie üblicherweise auf Linux auf. Somit wird unter Windows das Windows-Subsystem für Linux (WSL) genutzt, um die Container auszuführen. Dies ist eine schlanke Integration des Linux-Kernels in Windows und erspart den Overhead von herkömmlicher Virtualisierung.

Kennzeichnend für das Konzept von Docker sind die sogenannten Container. Diese stellen eine Zusammenstellung aller Komponenten dar, die eine bestimmte Anwendung zur Ausführung benötigt. Dadurch ist es sehr einfach, eine Applikation schnell und einheitlich auf einem neuen System zu deployen und dabei direkt alle Abhängigkeiten mit zu installieren und richtig zu konfigurieren.

Docker bietet außerdem eine Plattform an, welche Docker Hub genannt wird. Sie bietet eine Möglichkeit, eigene Docker Images hochzuladen, sowie die von anderen Nutzern hochgeladenen Images zu nutzen. Diese können zum Beispiel mit einem einzigen Kommandozeilenbefehl herunter geladen werden und sogar als Basis für neue, eigene Images dienen.

2.2 Node.js

In den letzten Jahren hat JavaScript in der Web-Entwicklung zunehmend an Bedeutung gewonnen. Node.js stellt eine Möglichkeit dar, wie JavaScript nicht mehr nur clientseitig im Browser ausgeführt werden kann, sondern auch serverseitig. Ein großer Vorteil davon liegt darin, dass Web-Entwickler, die bereits viel mit JavaScript arbeiten und dementsprechend damit vertraut sind, nur noch eine Sprache benötigen, um sowohl Front- als auch Backend zu entwickeln. Zudem kann JavaScript asynchron ausgeführt werden. Dies bedeutet, dass ein Web-Server nicht wie traditionell üblich eine Schlange von Anfragen bilden muss und diese nacheinander beantworten, sondern er kann die Anfragen gleichzeitig beantworten.

2.2.1 npm-Module

Eine weitere nützliche Funktion von Node.js ist der Node Package Manager (npm). Mit ihm können sehr einfach, von der Community erstellte, Bibliotheken installiert und in das Programm eingebunden werden. Auf diese Weise stehen beispielsweise fertige Frameworks für Web-Server, Unit-Tests oder erweiterte Logger zur Verfügung.

Die verwendeten Pakete werden in der `package.json` aufgezeichnet. Diese Datei dient zudem als eine Konfigurationsdatei für das Projekt. Dort können unter anderem Daten zum Autor, Lizenzen und Repository hinterlegt werden, sowie Skripte für npm definiert werden (zum Beispiel zum Starten, Testen oder Bauen des Projekts).

Fachliche
Richtig-
keit
prüfen
und kor-
rigieren

2.3 Postman

2.4 Kubernetes / Google Cloud Platform

2.5 Git

2.5.1 GitHub

2.6 Travis-CI

2.7 Jira

3 Textformatierungen

3.1 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.

3.2 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: % & \$ { }

3.3 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.

3.4 Abkürzungen

Während du schreibst benötigst 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. **ISBN!s** (**ISBN!s**).

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.

3.5 Anführungszeichen

Normale Anführungszeichen (") können in L^AT_EX 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 “

3.6 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 L^AT_EX. Mit dem Package „hyperref“ gibt es verschiedene Verweise, die in Tabelle 3.1 gelistet sind.

<code>\ref</code>	Zeigt die Nummer	Bsp.: 3.6
<code>\autoref</code>	Zeigt Typ + Nr.	Bsp.: Abschnitt 3.6
<code>\autopageref</code>	Zeigt „Seite Seitennummer“	Bsp.: Seite 8
<code>\nameref</code>	Zeigt den Namen (bzw. Caption)	Bsp.: Verweise und URLs
<code>\hyperref</code>	Eigener Text	Bsp.: Klick mich!

Tabelle 3.1: Verweise im Dokument

4 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.

4.1 Tabellen

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

Tabelle 4.1: 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 4.2 ist ...“!

Die Tabelle 4.2 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 4.2: 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 4.3 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 4.3: 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

4.2 Grafiken



Abbildung 4.1:
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 4.2: 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!

4.3 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 4.3 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.

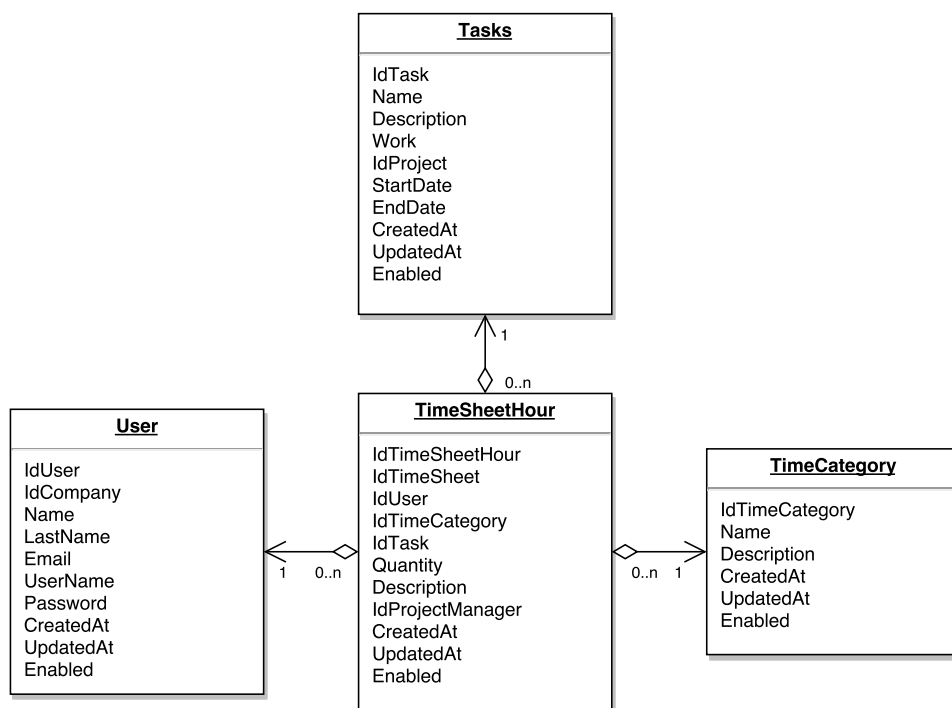


Abbildung 4.3: Diagramm eines Software Designs

5 Mathematik

5.1 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.

5.2 Formeln

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

$$\sum_{n=0}^{\infty} x = b + n \quad (5.1)$$

$$\frac{b * x}{c} = y \quad (5.2)$$

Trotz unterschiedlicher Länge kann man die Gleichheitszeichen auf der gleichen Höhe anbringen wie in Gleichung (5.3) und Gleichung (5.4) dargestellt, zusätzlich kann man diese auch mit Informationen versehen wie in Formel Gleichung (5.5) zu sehen.

$$a + b = c \quad (5.3)$$

$$5c + 3f = 4h \quad (5.4)$$

$$\overbrace{5y}^{y=0} + \underbrace{42x}_{x=1} = b \quad (5.5)$$

5.3 Arrays und Matrizen

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

$$\begin{aligned} z &= a \\ a + b &= c \\ f(x, y, z) &= x + y + z \end{aligned} \tag{5.6}$$

Hier noch ein paar Matrizen Beispiele in L^AT_EX.

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \tag{5.7}$$

$$\begin{bmatrix} 100 & 250 \\ 300 & 499 \end{bmatrix} \tag{5.8}$$

$$\left\{ \begin{array}{cc} 100 & 250 \\ 300 & 499 \end{array} \right\} \tag{5.9}$$

$$\left\| \begin{array}{cc} 100 & 250 \\ 300 & 499 \end{array} \right\| \tag{5.10}$$

5.4 Klammern und Kästchen

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

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

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

6 Programm- bzw. Quellcode

6.1 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 " + ↵
15             ↵ nutzerEingabe + " ist " + ergebnis);
16     }

```

Listing 6.1: 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 6.2: Initialisierung im Programm

Es ist auch möglich, innerhalb des Listings L^AT_EX 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 6.3: 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 tries:
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 6.4: Algorithmus zum schätzen einer Zahl in Python

6.2 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: $r \leftarrow a \bmod b$	
3: while $r \neq 0$ do	▷ We have the answer if r is 0
4: $a \leftarrow b$	
5: $b \leftarrow r$	
6: $r \leftarrow a \bmod b$	
7: end while	
8: return b	▷ The gcd is b
9: end procedure	

7 Literaturhinweise

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

7.1 FuSSnoten und Literaturverweise

Hamburger, Döner, Currywurst¹ - jeder kennt sie, jeder liebt sie und jeder isst sie. Weil die Zeit drängt[1], der Hunger groSS ist und der nächste Schnellimbiss[2] nur drei Schritte voraus. Und nach dem Essen? Sind wir zwar satt, aber meist nicht wirklich glücklich, weil Fastfood[3] meist eben auch nicht wirklich gut ist [4].

Ja, uns ist bewusst, dass die Literatur nicht zum Text passt. Deswegen hier nochmals der Rest.[5, 6, 7, 8, 9, 10, 11]

Übrigens: **ISBN!**s (**ISBN!**s) gehören nicht in die Bibliographie und werden deshalb auch nicht angezeigt.

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

Literaturverzeichnis

- [1] Bönner, C. u. a. *SAP Gateway und OData*. 2., aktualisierte und erweiterte Auflage. SAP Press. Bonn: Rheinwerk Publishing, 2016.
- [2] Forsthuber, H./ Siebert, J. *SAP ®-Finanzwesen: Das Praxishandbuch*. 6., aktualisierte und erweiterte Auflage. Rheinwerk Publishing. Bonn: Rheinwerk, 2016.
- [3] Friedl, G./ Pedell, B. *Controlling mit SAP ®: Eine praxisorientierte Einführung mit umfassender Fallstudie und beispielhaften Anwendungen*. 7. Aufl. 2017. Wiesbaden: Springer Fachmedien Wiesbaden, 2017.
- [4] Kühnlein, S./ Seubert, H. *Native Anwendungsentwicklung mit SAP HANA*. 1. Auflage. SAP Press. Bonn: Rheinwerk Publishing, 2016.
- [5] Mukherjee, S./ Mukherjee, S. *SAP MII: Functional and Technical Concepts in Manufacturing Industries*. Berkeley, CA: Apress und Imprint: Apress, 2017.
- [6] Preuss, P., Hrsg. *In-Memory-Datenbank SAP HANA*. SpringerLink : Bücher. Wiesbaden: Springer Gabler, 2017.
- [7] Schell, O. u. a. *Industrie 4.0 mit SAP: Digitale Transformation und das Internet of Things (IoT)*. 1. Auflage, neue Ausgabe. SAP Press. Bonn: Rheinwerk und SAP PRESS, 2017.
- [8] Smith, A. *SAP Integrates the New Blockchain Technologies: SAP Tests Different Blockchains*. Hrsg. von blogs.sap.com. <https://blogs.sap.com/2017/10/24/sap-integrates-the-new-blockchain-technologies/>. o.O., 2017. (Einsichtnahme: 24.10.2017).
- [9] Visser, C. *Das Ende eines perfekten Teams*. Hrsg. von Zeit Online. <http://www.zeit.de/wirtschaft/unternehmen/2013-07/sap-jim-hagemann-snabe-ruecktritt>. o.O., 2017. (Einsichtnahme: 24.10.2017).
- [10] Zaidi, R. *JavaScript essentials for SAP ABAP developers: A guide to mobile and desktop application development*. [New York, NY]: Apress, 2017.

- [11] o.V. *LaTeX arrows: Informationen zu LaTeX*. Hrsg. von Frank Marketing UG. <http://www.sascha-frank.com/Arrow/latex-arrows.html>. o.O., o.J. (Einsichtnahme: 30.10.2017).