

Leitfaden zur Übertragung von Informatiklehrmaterialien in eine blindengerechte Form



Inhalt

02 Vorwort

03 Hinweise für den Umgang mit blinden Studierenden

04 Möglichkeiten blindengerechter Aufbereitung von Inhalten

05 Platzierung der Alternativtexten

06 Grundlagen textueller Aufbereitung von Inhalten

07 Programmcode

08 Mathematische Formeln | LaTeX

09 Bäume | Markdown-Listen

10 Graphen | PlantUML

11 Ereignisketten | Sortierte Liste

12 Beispiel: <Programmcode/>

13 Beispiel: Mathematische Formeln | LaTeX

14 Beispiel: Bäume Markdown

15 Beispiel: Graphen | PlantUML

16 Literatur

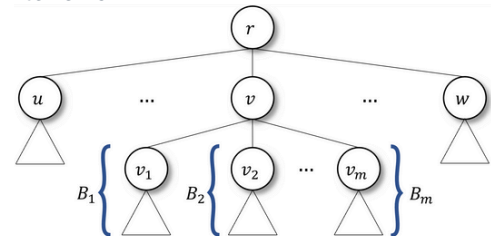


Vorwort

„Alternativtexte verwenden“ - Klingt selbsterklärend, ist es aber nicht immer

Die Grundlagen der Barrierefreiheit sind gut dokumentiert und in vielen Leitfäden nachzulesen, aber die Umsetzung ist nicht immer selbsterklärend.

Zum Beispiel ist das Erstellen eines Alternativtextes für den gezeigten Baum komplex. Eine Beschreibung wie „Baum mit den Knoten r, u, v, w, v1, v2, v3“ ist für blinde Studierende nicht ausreichend, da Kanten und Beschriftungen fehlen. Der Alternativtext muss alle relevanten Details und Verbindungen klar wiedergeben, um für blinde Personen zugänglich zu sein.



Beispielhafte Grafik eines Baumes [1, S. 6]

Wie lässt sich das am besten umsetzen?

Das Erstellen barrierefreier Inhalte ist keine unmögliche Aufgabe und erfordert keine starren Regeln. Dennoch gibt es grundlegende Prinzipien, die für ein barrierefreies Ergebnis wichtig sind. Dieser Leitfaden soll helfen, diese Prinzipien zu verstehen und zeigt effiziente Methoden, um insbesondere informatikbezogene Inhalte in eine textuelle Form zu bringen.

Der Leitfaden konzentriert sich auf die Bedürfnisse blinder Studierender und bietet Lösungen, die keine visuellen Informationen erfordern.

Es ist dennoch wichtig zu erwähnen, dass in der Praxis häufig modifizierte Grafiken auftreten, die nicht im Detail im Leitfaden behandelt werden können. Es geht deshalb vielmehr darum, Grundprinzipien, Anforderungen und Darstellungsmöglichkeiten zu verstehen, um situativ geeignete Lösungen erarbeiten zu können. Die Überarbeitung von Inhalten in eine blindengerechte Form ist nicht immer trivial, sondern erfordert teils eigene individuelle Lösungsansätze und Ideen.

Die TH Nürnberg Georg Simon Ohm (Ohm) hat bereits einen Leitfaden zur barrierefreien Lehre erstellt, der als Grundlage dieses Leitfadens dient [2]. Es wird empfohlen, den Abschnitt „Sehbeeinträchtigungen“ zusätzlich zu lesen, da er wichtige Themen wie Veranstaltungen, Leistungen und Nachteilsausgleiche beinhaltet.

Die beschriebenen Anweisungen setzen außerdem Grundkenntnisse der Informatik voraus. Weiterführende Links sind enthalten, um den Einstieg in weiterführende Recherchen zu erleichtern.

Der Leitfaden wurde im Rahmen der Bachelorarbeit „Leitfaden zur Gestaltung von Lehrmaterialien für blinde Studierende in der Informatik“ [3] erstellt. Details zu Quellen und Entstehung der Inhalte sind dort nachzulesen.

Frohes Überarbeiten,
Emilia Streck

Weiterführende Links

„[Barrierefreie Hochschullehre - Leitfaden für Lehrende](#)“ der Technischen Universität Dresden

„[Barrierefreie Lehre - Hinweise und Tipps für die inklusive Gestaltung von Medienformaten](#)“ der Goethe Universität Frankfurt am Main



Hinweise für den Umgang mit blinden Studierenden

Hilfe anbieten

Hilfe anbieten ist meist eine schöne Geste und zeigt Hilfsbereitschaft. Sie sollte jedoch immer erst mit “Kann ich Ihnen helfen?” oder “Brauchen Sie Unterstützung?” angeboten werden [4, S. 4], außer in Notfällen sollte niemals ohne Einverständnis eingegriffen werden.

Begrüßung und Verabschiedung

Bei der Begrüßung sollte sich immer vorgestellt werden, damit die blinde Person sich sicher sein kann, mit wem sie sich gerade unterhält [4, S. 10].

Vor dem Verlassen des Gesprächs sollte sich stets verabschiedet werden, damit die blinde Person weiß, wenn die Gesprächspartner*innen nicht mehr da sind [5, S. 2].

Unterstützung beim Finden des Weges

Um einer blinden Person dabei zu helfen, den Weg zu finden, eignet es sich am besten, den Arm zum Unterhaken anzubieten [4, S. 12]. Die blinde Person kann dann oberhalb des Ellenbogens greifen und leicht hinter der sehenden Person laufen.

Verbale Beschreibungen

Beschreibungen von Umgebungen oder Ereignissen können sehr hilfreich sein [4, S. 8]. In Vorlesungen kann es deshalb hilfreich sein, wenn Lehrkräfte verbal Handlungen beschreiben, um auch blinden Studierenden eine Teilnahme zu ermöglichen.

Auf jeden Fall zu vermeiden

- Ungefragte Berührungen der blinden Person, des Blindenstocks [4, S. 16] oder Blindenhundes [4, S. 20]
- Nonverbale Kommunikation mit Dritten [5, S. 2]
- Gespräche still verlassen und blinde Personen allein stehen lassen [4, S. 8]

Weiterführende Links

[Nicht so - sondern so](#) (DBSV)

[Helfen - aber wie?](#) (Blindenverband WNB)

[Die 30 goldenen Regeln im Umgang mit blinden und sehbehinderten Menschen](#) (mühlehalde)



Hinweise für den Umgang mit blinden Studierenden

“I think the advice I’d give to any teacher who had a blind student in their class would be to just treat them as normally as possible.”

- Tommy Edison in [Going To School As A Blind Person \(YouTube\)](#) [4]

Hilfe anbieten

Hilfe anbieten ist meist eine schöne Geste und zeigt Hilfsbereitschaft. Sie sollte jedoch immer erst mit “Kann ich Ihnen helfen?” oder “Brauchen Sie Unterstützung?” angeboten werden, außer in Notfällen sollte niemals ohne Einverständnis eingegriffen werden [4, S. 4].

Begrüßung und Verabschiedung

Bei der Begrüßung sollte sich immer vorgestellt werden, damit die blinde Person sich sicher sein kann, mit wem sie sich gerade unterhält [4, S. 10].

Vor dem Verlassen des Gesprächs sollte sich stets verabschiedet werden, damit die blinde Person weiß, wenn die Gesprächspartner nicht mehr da sind [5, S. 2].

Unterstützung beim Finden des Weges

Um einer blinden Person dabei zu helfen, den Weg zu finden, eignet es sich am besten, den Arm zum Festhalten anzubieten [4, S. 12]. Die blinde Person kann dann oberhalb des Ellenbogens greifen und leicht hinter der sehenden Person laufen.

Verbale Beschreibungen

Beschreibungen von Umgebungen oder Ereignissen können sehr hilfreich sein [4, S. 8]. In Vorlesungen kann es deshalb hilfreich sein, wenn Lehrkräfte verbal Handlungen beschreiben, um auch blinden Studierenden eine Teilnahme zu ermöglichen.

Auf jeden Fall zu vermeiden

- Ungefragte Berührungen der blinden Person, des Blindenstocks [4, S. 16] oder Blindenhundes [4, S. 20]
- Nonverbale Kommunikation mit Dritten [5, S. 2]
- Gespräche still verlassen und blinde Personen allein stehen lassen [4, S. 8]

Weiterführende Links

[Nicht so - sondern so](#) (DBSV)

[Helfen - aber wie?](#) (Blindenverband WNB)

[Die 30 goldenen Regeln im Umgang mit blinden und sehbehinderten Menschen](#) (mühlenthalde)



Möglichkeiten blindengerechter Aufbereitung von Inhalten

Um Inhalte für blinde Menschen zugänglich zu machen, sind primär textuelle oder taktile Darstellungen geeignet.

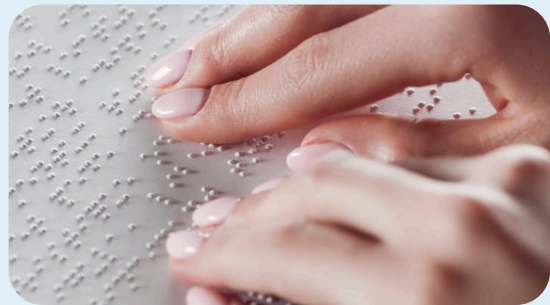
In der **textuellen Darstellung** werden die Informationen in Textform beschrieben, wie bei Alternativtexten. Diese Texte können dann mit einem Screenreader und/oder einer Braillezeile ausgegeben werden.



Laptop mit angeschlossener
Braillezeile [7]

Braillezeilen geben die Informationen des Bildschirms in der Blindenpunktschrift Braille aus. Höhenveränderbare Stifte stellen dabei die Braillepunkte dar, die von dem Anwender mit den Fingerkuppen ertastet werden. *„Die Braillezeile ist in der Regel nur für das Lesen des Computerbildschirms zuständig. Die Eingabe von Daten findet über eine herkömmliche Tastatur statt“* [8, S.1].

Grafiken können außerdem durch **taktile Darstellungsmethoden** wie Brailledruck oder Schwelldruck aufbereitet werden, um anschließend ertastet werden zu können.



Finger, die Brailledruck ertasten [9]

„Brailledrucker drucken bzw. stanzen erhabene Braille-Zeichen auf spezielles Papier, dem Braillepapier“ [10].

In diesem Leitfaden liegt der Fokus auf der textuellen Aufbereitung von Lehrmaterialien.

Weiterführende Links

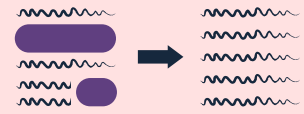
[Braillezeile - Was ist das?](#) (Marlem-Software Barrierefreiheit in der Informatik)

[Informationen zur Arbeitsplatzausstattung - Braillezeilen](#) (DBSV, incobs)



Platzierung der Alternativtexte

Der Leitfaden gibt konkrete Anweisungen dazu, wie verschiedene Inhalte in eine **rein textuelle Darstellung** überführt werden können. Diese muss **zusätzlich zu vorhandenen visuellen Darstellungen** zur Verfügung gestellt werden.



In der Praxis stellt sich jedoch zu Beginn der Überarbeitung meistens die Frage, wie die Alternativtexte und Ergänzungen in die aktuellen Lehrmaterialien für sehende Studierende integriert werden können.

Organisationen, die sich auf die Übertragung von Lehrinhalten in eine blindengerechte Form spezialisiert haben, wie z. B. mediablis, stellen blinden Schüler*innen die Inhalte meistens als **separates Word-Dokument** zur Verfügung [11, S. 7]. Diese sind für sie am besten zugänglich, bringen jedoch in der Praxis den Nachteil mit sich, dass mehrere Dokumente gepflegt werden müssen.

Ist der Aufwand eines separaten Dokumentes für blinde und sehende Studierende zu hoch, können die zusätzlichen textuellen Informationen auch in die schon bestehenden Lehrmaterialien eingefügt werden.

Bei Abbildungen können Sie die Texte als **Alternativtext** hinterlegen. Machen Sie hierfür einen Rechtsklick auf das entsprechende Objekt und wählen anschließend „Alternativtext anzeigen“. Sie können nun den Alternativtext eingeben oder das Objekt als dekorativ kennzeichnen.

Alternativtext anzeigen...

Alternativtexte werden von sehende Studierende typischerweise nicht gelesen, was zu Nachteilen führen kann. Aus diesem Grund bietet es sich ebenfalls an, die Texte in die **Bildbeschreibung** oder auf PowerPoint in die **Notizzettel** zu schreiben.

Klicken Sie, um eine Notiz hinzuzufügen

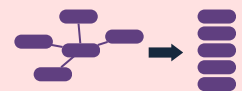


Die bestmögliche Platzierung der textuellen Alternativdarstellungen kann in der Praxis variieren. Es wird deshalb immer empfohlen mit **betroffenen Studierenden zu sprechen**, um die von ihnen präferierte Lösung zu finden.

Inhalte dürfen außerdem nicht wie in einer Mindmap oder auf einer Präsentationsfolie zweidimensional angeordnet sein, sondern müssen eine **eindeutige, lineare Reihenfolge** haben, in der sie gelesen werden sollen.

Textdokumente erfüllen dieses Kriterium meist automatisch, da sie linear aufgebaut sind und der Lesefluss Zeile für Zeile von oben nach unten verläuft.

Bei einer Präsentation muss die Reihenfolge mithilfe der „**Lesereihenfolge**“ überprüft und angepasst werden.



Weiterführende Links

[Alles, was Sie wissen müssen, um effektiven Alternativtext zu schreiben](#) (Microsoft)

[Hinzufügen von Sprechernotizen zu Folien](#) (Microsoft)

[Erstellen von Folien mit einer barrierefreien Lesereihenfolge](#) (Microsoft)



Grundlagen textueller Aufbereitung von Inhalten



„Ganz allgemein kann man sich vorstellen, man würde die Inhalte jemandem am Telefon erklären.“ – Michael Engel (BliZ) [12]

Für Überschriften, Text und Aufzählungen sollten Sie immer **Formatvorlagen** verwenden. „Dadurch wird dem Dokument eine Struktur gegeben. Diese ist für den blinden und sehbehinderten Nutzer unbedingt erforderlich“ [11, S. 7].

Zu Beginn können Sie bei Unsicherheit **Volltext** verwenden, um Inhalte, wie am Telefon, zu beschreiben [12].

Sie sollten grundsätzlich nur Zeichen verwenden, die auf der deutschen oder englischen **Tastatur** zu finden sind. Es ist davon auszugehen, dass Screenreader diese Zeichen kennen und verarbeiten können [12]. Sonderzeichen können mit LaTeX beschrieben werden.

Sollte eine zusätzliche **Beschreibung** nötig sein, kann diese mit **eckigen Klammern** dazugeschrieben werden [12].

Hiermit können auch textuelle Beschreibungen, wie die einer Grafik, angekündigt werden.

[Farbe: rot]

[Zur Beschreibung der Grafik wurde PlantUML verwendet]

Eine Alternative dazu ist eine Auszeichnung mithilfe von Tags, wie sie von mediablis verwendet werden, um Abschnitte zu beschreiben [11, S. 10].

<L>LaTeX-Element</>

<Bild>Bildbeschreibung</>

Sie sollten blinden Studierenden immer die ursprünglichen Dateien (LaTeX-, Word-, PowerPoint-Dokument) zur Verfügung stellen, nicht das exportierte PDF, da dieses für blinde Menschen im Allgemeinen schlechter zugänglich ist [12].



Weiterführende Links

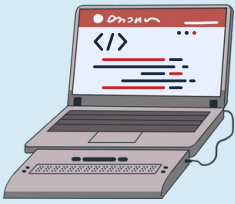
[Anwenden von Formatvorlagen](#) (Microsoft)

[E-Buch Standard 2016](#) (mediablis)



< Programmcode />

Codeelemente, wie sie in vielen Informatikfächern vorkommen, sind bereits textuell und erfordern daher keine umfangreiche Übertragung. So kann das **gleiche Dokument** für sehende und blinde Studierende zur Verfügung gestellt werden. Dennoch gibt es einige Details, die berücksichtigt werden sollten, um den Code für blinde Menschen leichter zugänglich zu machen:



Blinde Informatiker*innen können Code lesen und schreiben, indem der Text Zeile für Zeile mithilfe von Sprachausgabe oder einer Braillezeile ausgegeben wird.

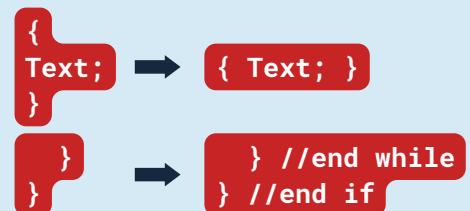
Dabei verläuft die Navigation innerhalb des Projekts über Suchfunktionen, Navigation zwischen den Dateien und das lineare Durchlaufen der einzelnen Zeilen, bis die richtige Stelle im Code erreicht wurde.

Projekte sollten deshalb möglichst in **viele kleinere Dateien** unterteilt werden.

Die Suche nach der richtigen Stelle im Code über Dateien ist leichter als über einzelne Zeilen [13].

Es empfiehlt sich außerdem, **Klammern nicht allein** in eine Zeile zu setzen, da der Code sonst vertikal gestreckt wird und mehr Zeilen einzeln durchlaufen werden müssen [12].

Bei mehreren aufeinanderfolgenden Klammern bietet es sich an, die Klammern mithilfe von Kommentaren zuzuordnen [13].



Kommentare können normal notiert werden, da sie meistens mit Zeichen der Tastatur beschrieben werden.

`// Kommentar`

`<!-- Kommentar -->`

`# Kommentar`

In Editoren, wie Word:

← Einrückungen sollten immer mit **Tabstops** gelöst werden, nie mit Leerzeichen.
→

Generell gilt es, nur Zeichen zu verwenden, die auf der Tastatur zu finden sind.

Enthält Pseudocode zum Beispiel Pfeile, sollten diese mit `-->` oder dem LaTeX-Befehl `\to` oder `\leftarrow`, `\rightarrow` ersetzt werden. Bei automatischen Umwandlungen der Pfeile durch die verwendete Software kann es in manchen Programmen, wie z. B. Word oder PowerPoint, helfen, Strg+Z zu drücken.

Auf PowerPoint-Folien sollten immer **Textfelder** verwendet werden, keine besonderen Formatelemente oder Screenshots.

Elemente **fett** oder *kursiv* hervorzuheben ist möglich.

Mathematische Formeln | LaTeX

„**LaTeX** ist eine Markupsprache und ein Dateiformat, um insbesondere Texte, die mathematische Formeln enthalten, zu schreiben und für den Druck oder die Bildschirmansicht zu formatieren“ [14]. Sie kann genutzt werden, um visuelle Elemente in Formeln, wie zum Beispiel einen Bruchstrich, textuell darzustellen und eignet sich deshalb bei der Übertragung mathematischer Formeln in eine barrierefreie, textuelle Form. Die LaTeX-Darstellung kann als **Alternativtext**, in den **Notizzetteln** der PowerPoint oder im **Fließtext** der Word-Datei hinterlegt werden.

Formeln, die mithilfe des Formeleditor in Word erstellt wurden, sind für blinde Personen nicht gut lesbar und sollte deshalb bei einer Übertragung der Inhalte **nicht** verwendet werden!



Um visuelle Unterstützung und Feedback bei der Erstellung der LaTeX-Formel zu bekommen, eignen sich **online LaTeX-Editoren**: [Lagrida](#)



LaTeX-Cheatsheet [16]

Alle Zeichen auf der Tastatur können zunächst normal verwendet werden.

+ **-** **=** **<** **>** **/** **:** ***** **'** **|** **[** **]** **(** **)**

% und {} haben in LaTeX eine gesonderte Funktion und müssen deshalb mit \ angekündigt werden.

\% **\{** **\}**

Exponenten: **x^3** → **`x^3`** **x^{n+m}** → **`x^{n+m}`**

Indizes: **x_1** → **`x_1`** **x_{n-1}** → **`x_{n-1}`**

Brüche: **$\frac{a+b}{a-b}$** → **`\frac{a+b}{a-b}`**

Wurzeln: **$\sqrt{a+b}$** → **`\sqrt{a+b}`**

Summe: **$\sum_{n=1}^5 n$** → **`\sum_{n=1}^5 n`**

Unendlich: **∞** → **`\infty`**

L^AT_EX

[10]

Weiterführende Links

[LaTeX-Manual-LaTeX-Konzept](#) (Augenbit)

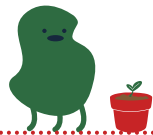
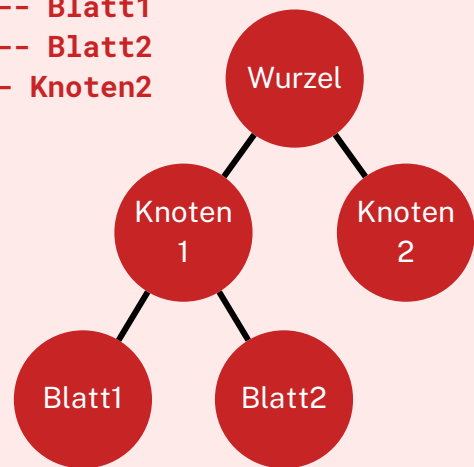


Bäume | Markdown-Listen

Markdown ist eine Markdown-Sprache mit der Formatierungen zu einfachen Textdokumenten hinzugefügt werden können [17].

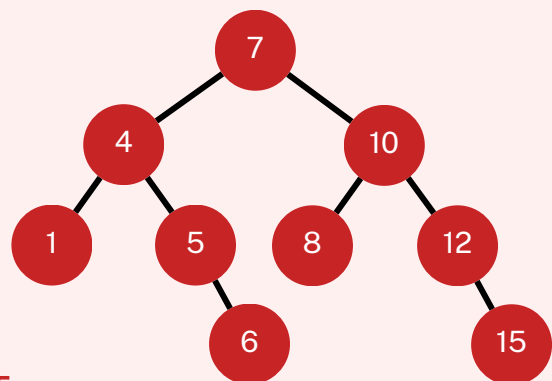
Um einen Baum mit Markdown zu beschreiben, können leicht abgeänderte, **unsortierte Listen** verwendet werden. Beginnen Sie hierfür mit der **Wurzel ohne Listenzeichen**. Durchlaufen Sie anschließend die Knoten **preorder** bis zu einem Blatt. Dabei wird für jede Ebene im Baum eine Ebene in der Liste eingerückt, indem **ein zusätzliches Minus vor das Element** geschrieben wird.

```
Wurzel
- Knoten1
-- Blatt1
-- Blatt2
- Knoten2
```



Beispiel

```
7
- 4
-- 1
-- 5
--- 6
-10
-- 8
-- 12
--- 15
```



Weiterführende Links

[Markdown Guide](#)

Wikipedia: [Baum \(Datenstruktur\)](#)

Wikiwand: [Baum \(Datenstruktur\)](#)

[Preorder Traversierung](#) (Uni Wuppertal)



Graphen | PlantUML

PlantUML ist ein Werkzeug, um eine Vielzahl von Diagrammen beschreiben und darstellen zu können. Es wird zur Übertragung von Graphen für blinde Studierende empfohlen.

Bei Bäumen, einer Sonderform von Graphen, hat sich Markdown bewährt, da sie eine eindeutige Hierarchie aufweisen. Sie sind jedoch ebenfalls mit PlantUML umsetzbar. Dies kann aus Gründen der Konsistenz Vorteile haben.

Ein Graph kann mit PlantUML wie folgt beschrieben werden:

- Start mit **@startuml**
- Baumstruktur
 - Startzustand mit: **[*] --> Knoten**
 - Alle Kanten mit:
 - Ungerichtete, ungewichtete Kante: **Startknoten -- Endknoten**
 - Ungerichtete, gewichtete Kante: **Startknoten -- Endknoten : Gewicht**
 - Gerichtete, ungewichtete Kante: **Startknoten --> Endknoten**
 - Gerichtete, gewichtete Kante: **Startknoten --> Endknoten : Gewicht**
- Ende mit **@enduml**



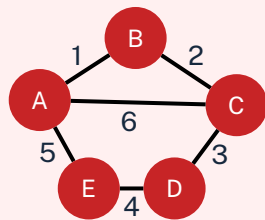
Um ein visuelles Feedback der Übersetzung zu bekommen, empfehlen sich **Online Editoren** sehr. Die unten stehenden Beispiele sind mit [PlantText UML Editor](#) umgesetzt. Die visuelle Ausgabe hilft dabei nur zur Kontrolle der Überarbeitung. Für blinde Studierende ist nur der PlantUML-Text relevant.

Beispiele

Graph mit ungerichteten, gewichteten Kanten:

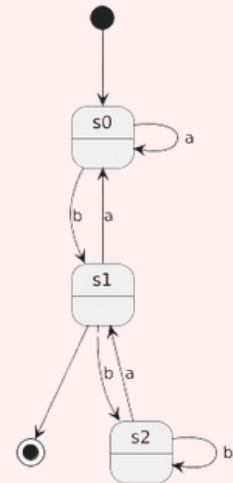
```
@startuml
A -- B : 1
B -- C : 2
C -- D : 3
D -- E : 4
E -- A : 5
A -- C : 6
@enduml
```

Achtung! Hierbei handelt es sich um ein I, nicht um eine II!



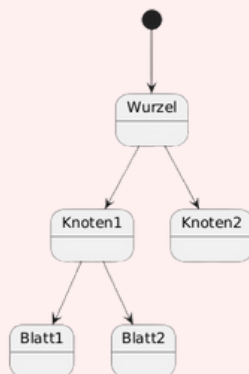
Zustandsübergangsdiagramm:

```
@startuml
[*] --> s0
s0 --> s0 : a
s0 --> s1 : b
s1 --> [*]
s1 --> s0 : a
s1 --> s2 : b
s2 --> s1 : a
s2 --> s2 : b
@enduml
```



Baum:

```
@startuml
[*] --> Wurzel
Wurzel --> Knoten1
Wurzel --> Knoten2
Knoten1 --> Blatt1
Knoten1 --> Blatt2
@enduml
```



Weiterführende Links

Kap. 7 State Diagramm in [PlantUML Sprachreferenz](#)



Ereignisketten | Sortierte Liste

In der Informatik werden Abfolgen von **Ereignissen** häufig visuell dargestellt. In solchen Fällen eignen sich sortierte Listen, die die **Chronologie** verdeutlichen können, um die Darstellungen für blinde Studierende aufzubereiten.

Hierfür können **Formatvorlagen** in Word verwendet werden oder die Nummerierung händisch getippt werden.

1. Ereignis 1
2. Ereignis 2
3. Ereignis 3
4. Ereignis 4

Sollen zusätzlich zu den Ereignissen weitere Informationen angegeben werden, wie z. B. die Auswirkung des Ereignisses, muss die Notation individuell betrachtet werden.

Eine Möglichkeit ist, die zusätzlichen Informationen in die gleiche Zeile wie das Ereignis zu schreiben:

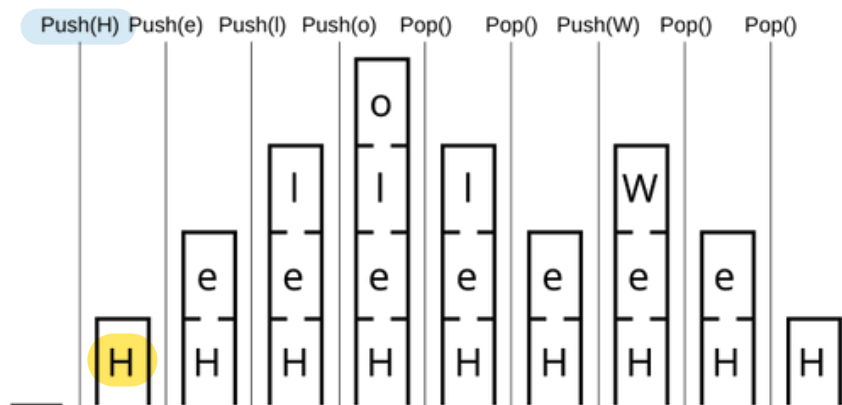
1. Ereignis 1 => Folge
2. Ereignis 2 => Folge
3. Ereignis 3; Informationen
4. Ereignis 4 => Folge

Werden dadurch die Zeilen unübersichtlich oder zu lang, können zusätzliche Listenelemente verwendet werden:

1. Ereignis 1
2. Folge
3. Ereignis 2
4. Folge
5. Ereignis 3
6. Informationen
7. Ereignis 4
8. Folge

Beispiel

1. Keller: [leer]
2. Push(H) => Keller: H
3. Push(e) => Keller: He
4. Push(l) => Keller: Hel
5. Push(o) => Keller: Helo
6. Pop() => Keller: Hel
7. Pop() => Keller: He
8. Push(W) => Keller: HeW
9. Pop() => Keller: He
10. Pop() => Keller: H



Beispielhafte Darstellung eines Stapelspeichers [13]

Beispiel: < Programmcode />

IV. Laufzeit von Algorithmen

IV.4.1 Nicht-rekursive Algorithmen mit Methodenaufruf

SelectionSort2:

- Variante von *SelectionSort*
- Kapselung der Minimumsberechnung in eigener Methode *MinIndex*:
 - Berechnet *Index* des Minimums im Teilbereich $a[l..r]$ des Arrays $a[1..n]$
 - *MinIndex*: Vorbedingung: $r > l$

```
void SelectionSort(item a[1..n])  
[Knebl (2021), S.2.]  
1  index i, j, k; item m  
2  for i ← 1 to n - 1 do  
3      k ← i; m ← a[i]  
4      for j ← i + 1 to n do  
5          if a[j] < m  
6              then k ← j; m ← a[j]  
7      exchange a[i] and a[k]
```

Methode als reiner Text

```
void SelectionSort2(item a[1..n])  
1  index i, k; item m  
2  for i ← 1 to n - 1 do  
3      k ← MinIndex(a[1..n], i, n)  
4      exchange a[i] and a[k]
```

```
index MinIndex(item a[1..n], index l, r)  
1  index j, k; item m  
2  k ← l; m ← a[l]  
3  for j ← l + 1 to r do  
4      if a[j] < m  
5          then k ← j; m ← a[j]  
6  return k
```

Prof. Dr. Thomas Fuhr – TH Nürnberg

3

Beispielhafte Folie aus dem Fach Algorithmen und Datenstrukturen
[19, S. 3]

<Methode SelectionSort>

[Methodenkopf] void SelectionSort (item a[1..n]) [Knebl (2021), S. 2]

[Methodenrumpf]

1 index i, j, k; item m

2 for i ← 1 to n - 1 do

3 k ← i; m ← a[i] [orange]

4 for j ← i + 1 to n do [orange]

5 if a[j] < m [orange]

6 then k ← j; m ← a[j] [orange]

7 exchange a[i] and a[k]

</Methode SelectionSort>

<Methode SelectionSort2>

[Methodenkopf] void SelectionSort2 (item a[1..n])

[Methodenrumpf]

1 index i, k; item m

2 for i ← 1 to n - 1 do

3 k ← MinIndex (a[1..n], i, n) [orange]

4 exchange a[i] and a[k]

</Methode SelectionSort2>

Beispiel: Mathematische Formeln | LaTeX

IV. Laufzeit von Algorithmen
IV.4.1 Nicht-rekursive Algorithmen mit Methodenaufruf

Worst Case – Gesamtansatz

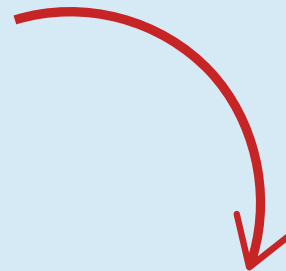
Alle Zeilen	$a_{\text{MinIndex}}(l, r) = 1 + \sum_{j=l+1}^r (1 + 1 + 1) + 1 + 1$
-------------	--

z.2 z.3 z.3 z.4 z.5 z.3 z.6
 / / / / / / /
 Durchläufe for-Schleife Abbruchtest

```

void MinIndex(item a[1..n], index l, r)
1  index j, k; item m
2  k ← l; m ← a[l]
3  for j ← l + 1 to r do
4      if a[j] < m
5          then k ← j; m ← a[l]
6  return k
  
```

Mathematische Formel mit LaTeX



Beispielhafte Folie aus dem Fach Algorithmen und Datenstrukturen [20, S. 8]

$$a_{\text{MinIndex}}(l, r) = 1 + \sum_{j=l+1}^r (1+1+1) + 1 + 1$$

Ausführlicher:

[Darstellung einer Formel in Verbindung mit der Methode MinIndex. Beides wird zunächst separat beschrieben. Anschließend wird der Zusammenhang beschrieben.]

<L> $a_{\text{MinIndex}}(l, r) = 1 + \sum_{j=l+1}^r (1+1+1) + 1 + 1$ </L>

<Methode MinIndex>

[Methodenkopf] index MinIndex (item a[1..n, index l, r)

[Methodenrumpf]

1 index j, k; item m

2 $k \leftarrow l; m \leftarrow a[l]$

3 for j $\leftarrow l + 1$ to r do

4 if $a[j] < m$

5 then $k \leftarrow j; m \leftarrow a[l]$

6 return k

</Methode MinIndex>

[Zusammenhang, zunächst Beschreibung der Formel und dann der entsprechenden Codezeile]

$a_{\text{MinIndex}}(l, r) =$

1+ : Z. 2 $k \leftarrow l; m \leftarrow a[l]$

$\sum_{j=l+1}^r$ (1 : Durchläufe for Schleife Z. 3 for j $\leftarrow l + 1$ to r do

+1 : Z. 4 if $a[j] < m$

+1) : Z. 5 then $k \leftarrow j; m \leftarrow a[l]$

+1 : Abbruchtest Z. 3 for j $\leftarrow l + 1$ to r do

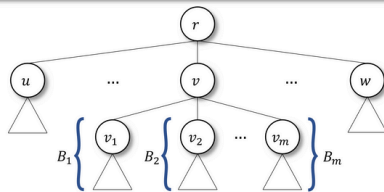
+1 : Z. 6 return k

Beispiel: Bäume | Markdown

II.2 Dictionaries - Bäume
II.2.1 Grundlagen

Definition:

Sei B ein Wurzelbaum. Sei v ein beliebiger Knoten von B .
Hat v die Nachfolger v_1, \dots, v_m , so heißen die Teilbäume B_1, \dots, B_m mit den Wurzeln v_1, \dots, v_m , die **Teilbäume von v** .



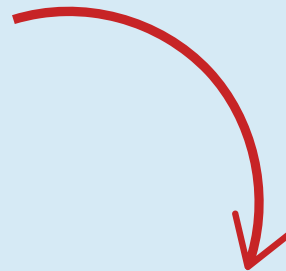
Bemerkung:

Jeder Knoten v eines Wurzelbaums B ist selbst Wurzel eines Teilbaums von B . Wurzelbäume sind eine rekursive Struktur.

Prof. Dr. Thomas Fuhr – TH Nürnberg

6

Darstellung des Baums mit Markdown



Beispielhafte Folie aus dem Fach Algorithmen und Datenstrukturen [21, S. 6]

[Grafik, die die Benennung der Knoten und Teilbäume darstellt. Teilweise sind Knoten mit einem Dreieck mit Spitze nach oben darunter dargestellt. Das bedeutet, dass an dieser Stelle ein Teilbaum mit dem beschriebenen Knoten als Wurzel vorhanden ist. Zwischen den einzelnen Knoten befinden sich außerdem drei Punkte, welche signalisieren, dass weitere Knoten dazwischenliegen können.]

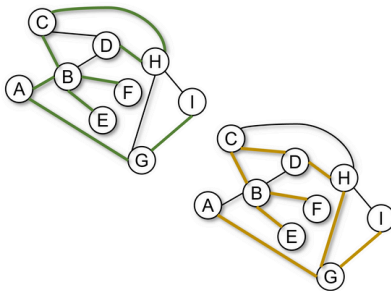
```
r
- Teilbaum mit Wurzel u
- ...
- v
-- Teilbaum B_1 mit Wurzel v_1
-- Teilbaum B_2 mit Wurzel v_2
-- ...
-- Teilbaum B_m mit Wurzel v_m
- ...
- Teilbaum mit Wurzel w
```

Beispiel: Graphen | PlantUML

VI. Graphen
VI.4.1 Grundlegende Definitionen

Definition:

Ein **Spannbaum** (*Gerüst, spanning tree*) eines ungerichteter Graphen G ist ein Teilgraph von G , welcher ein freier Baum ist und alle Knoten von G umfasst.



Bemerkungen:

- Ein ungerichteter Graph G kann mehrere unterschiedliche Spannäume besitzen.
- Unzusammenhängende ungerichtete Graphen besitzen keine Spannäume.

Beispielhafte Folie aus dem Fach Algorithmen und Datenstrukturen [22, S. 11]

Textuelle Beschreibung des ersten Graphens mit PlantUML

```
A -- B [dick]
A -- G [dick]
B -- C [dick]
B -- D
B -- E [dick]
B -- F [dick]
C -- D
C -- H [dick]
D -- H [dick]
G -- H
G -- I[dick]
H -- I
```

Literatur

- [1] T. Fuhr, *II.2.1+2-AlgoDat-Fuhr-Dict-Baeume-v1.5 (Präsentation)*, Technische Hochschule Nürnberg Georg Simon Ohm, o. D.
- [2] U. Messmann, *Barrierefreie Lehre - Leitfaden für Lehrende*, Technische Hochschule Nürnberg Georg Simon Ohm, nicht veröffentlicht.
- [3] E. Streck, *Leitfaden zur Gestaltung von Lehrmaterialien für blinde Studierende in der Informatik*, Bachelorarbeit, Technische Hochschule Nürnberg Georg Simon Ohm, 2024.
- [4] M. Putzer-Schimack und M. Tree, *Helfen - aber wie?*, Blinden- und Sehbehindertenverband Wien, Niederösterreich und Burgenland, o. D. [Online]. Verfügbar unter: https://www.blindenverband-wnb.at/fileadmin/user_upload/Blindenverband/Downloads/4_-_Helfen_aber_wie.pdf (Zugriff am: 9.05.2024).
- [5] mühlehalde, *Die 30 goldenen Regeln im Umgang mit blinden und sehbehinderten Menschen*, o. D. [Online]. Verfügbar unter: https://www.muehlehalde.ch/fileadmin/documents/muehlehalde.ch/PDF/NEU_30-goldene-Regeln-Muehlehalde.pdf (Zugriff am: 22.06.2024).
- [6] H. van Dyck, *Nicht so - sondern so - Kleiner Ratgeber für den Umgang mit blinden Menschen*, 22., aktualisierte Aufl., Deutscher Blinden- und Sehbehindertenverband e. V., 2020. [Online]. Verfügbar unter: <https://www.dbsv.org/broschueren.html> (Zugriff am: 16.07.2024).
- [7] sehkreis.at, *IMG_20220422*, [Online]. Verfügbar unter: https://www.sehkreis.at/wp-content/uploads/2022/06/IMG_20220422_160813-frei-1024x576.jpg (Zugriff am: 12.09.2024).
- [8] Deutscher Blinden- und Sehbehindertenverband e.V., *Informationen zur Arbeitsplatzausstattung – Braillezeilen*, 2007. [Online]. Verfügbar unter: http://archiv.incobs.de/incobs.de/downloads/broschueren/pdf_braillezeilen.pdf (Zugriff am: 29.07.2024).
- [9] irie-at, *TactileGraphics-1.jpg*, [Online]. Verfügbar unter: <https://irie-at.com/wp-content/uploads/2018/06/TactileGraphics-1.jpg> (Zugriff am: 12.09.2024).
- [10] Wikipedia - Die freie Enzyklopädie, *Brilledrucker*. [Online]. Verfügbar unter: <https://de.wikipedia.org/wiki/Brilledrucker> (Zugriff: 12.09.2024).
- [11] mediablis, *E-Buch Standard 2016 - für die Übertragung von Schulbüchern sowie von Zentralen Prüfungen und anderen schriftlichen Leistungsfeststellungen im Rahmen des Nachteilsausgleichs*, 2016. [Online]. Verfügbar unter: https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fmediablis-bayern.de%2Fwp-content%2Fuploads%2F2023%2F05%2Fe-buch-standard-2016_finale-fassung.docx&wOrigin=BROWSELINK (Zugriff am 5.07.2024).
- [12] E. Streck, *Experteninterview zur blindengerechten Übertragung von Informatiklehrmaterialien mit Michael Engel*, Mai 2024, nicht veröffentlicht.
- [13] E. Streck, *Experteninterview zur blindengerechten Übertragung von Informatiklehrhalten mit einem blinden Fachinformatiker*, Juli 2024, nicht veröffentlicht.
- [14] Wikipedia - Die freie Enzyklopädie, *LaTeX*. [Online]. Verfügbar unter: <https://de.wikipedia.org/wiki/LaTeX> (Zugriff am: 12.09.2024).
- [15] Logos Download, *LaTeX_Logo.png*. [Online]. Verfügbar unter: https://logos-download.com/wp-content/uploads/2019/07/Latex_Logo.png (Zugriff am: 12.09.2024).
- [16] Augenbit, *LaTeX-Manual-LaTeX-Konzept*. [Online]. Verfügbar unter: <https://www.augenbit.de/wiki/index.php?title=LaTeX-Manual-LaTeX-Konzept> (Zugriff am: 16.09.2024).
- [17] Markdown Guide, *Getting Started - An overview of Markdown, how it works, and what you can do with it.* [Online]. Verfügbar unter: <https://www.markdownguide.org/getting-started/> (Zugriff am: 17.06.2024).
- [18] Wikiwand, *Skizze eines Stapelspeicher*. [Online]. Verfügbar unter: https://www.wikiwand.com/de/articles/Praktische_Informatik (Zugriff am: 12.09.2024).
- [19] T. Fuhr, *IV.4-AlgoDat-Fuhr-Laufzeit-v1.2 (Präsentation)*, Technische Hochschule Nürnberg Georg Simon Ohm, WiSe 2023/24.
- [20] T. Fuhr, *IV.4-AlgoDat-Fuhr-Laufzeit-v1.2 (Präsentation)*, Technische Hochschule Nürnberg Georg Simon Ohm, WiSe 2023/24.
- [21] T. Fuhr, *II.2.1+2-AlgoDat-Fuhr-Dict-Baeume-v1.5 (Präsentation)*, Technische Hochschule Nürnberg Georg Simon Ohm, WiSe 2023/24.
- [22] T. Fuhr, *VI.4-AlgoDat-Fuhr-Graphen-v1.5 (Präsentation)*, Technische Hochschule Nürnberg Georg Simon Ohm, WiSe 2023/24.

Herausgeber:
Emilia Streck, 2024

Der Leitfaden wurde im Rahmen der Bachelorarbeit „Leitfaden zur Gestaltung von Lehrmaterialien für blinde Studierende in der Informatik“ an der Technischen Hochschule Nürnberg Georg Simon Ohm erstellt. Details zu Quellen und Entstehung der Inhalte sind dort nachzulesen.

Das Design wurde inspiriert durch den Leitfaden „Inklusiv studieren - Studieren mit Behinderung“ des Deutschen Studentenwerkes.

Die grafischen Elemente basieren auf dem Plakat „Vielfalt“ von Alexandra Becker (damals Wilhelm), 3. Platz beim Plakatwettbewerb ‚Studieren mit Behinderung oder chronischer Krankheit‘ des Deutschen Studentenwerks im Jahr 2012, gefördert vom Bundesministerium für Bildung und Forschung.

