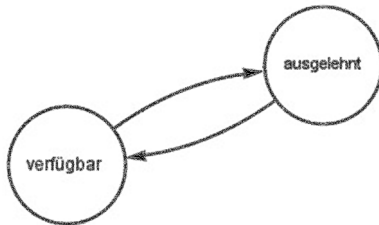


Automation: Automaten

Wir haben gesehen, dass Automaten verwendet werden können, um vielfältige Abläufe formal zu beschreiben. Eine Bibliothek möchte die Abläufe ihres Ausleihsystems mit Automaten beschreiben. Der erste Entwurf sieht wie folgt aus:



Dahinter stehen folgende Überlegungen: Ein Buch kann verfügbar sein (Zustand verfügbar). Ein Buch kann ausgelohnt sein (Zustand ausgelohnt). Ein verfügbares Buch kann ausgelohnt werden (Übergang verfügbar→ausgelohnt). Ein ausgelohntes Buch kann verfügbar werden (Übergang ausgelohnt→verfügbar).

Bücher sollen auch reserviert werden können, und ausgelohnte Bücher sollen verlängert ausgelohnt werden können. Entwerfen Sie für die Bibliothek einen Automaten, der folgende Überlegungen abdeckt:

Ein Buch kann verfügbar sein. Ein Buch kann ausgelohnt sein. Ein Buch kann reserviert sein. Ein Buch kann verlängert sein. Ein verfügbares Buch kann ausgelohnt oder reserviert werden. Ein ausgelohntes Buch kann verfügbar oder reserviert oder verlängert werden. Ein reserviertes Buch kann ausgelohnt oder verfügbar werden. Ein verlängertes Buch kann reserviert oder verfügbar werden.

Tragen Sie unten die Zustände und Übergänge wie im Bild oben ein:

- Zustände klar benannt.
- Übergänge eindeutig erkennbar.

Aufgabe: Kara Programm lesen

Das Programm in der Abbildung rechts führt Kara zu dem Kleeblatt. Zeichnen Sie den Weg ein, den Kara nimmt. Nummerieren Sie dazu die Felder, die Kara besucht. Das Feld, auf dem Kara startet, ist Feld Nummer 1.

	<div style="border: 1px solid black; padding: 5px;"> <p>suche</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;"></th> <th style="width: 15%;">Kara macht:</th> <th style="width: 15%;">Nächster Zustand:</th> <th style="width: 60%;"></th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;"> </td> <td style="text-align: center;"> </td> <td style="text-align: center;">suche</td> </tr> <tr> <td style="text-align: center;"> <input checked="" type="checkbox"/> no <input type="checkbox"/> yes </td> <td style="text-align: center;"> <input type="checkbox"/> no </td> <td style="text-align: center;"> </td> <td style="text-align: center;">suche</td> </tr> <tr> <td style="text-align: center;"> <input checked="" type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no </td> <td style="text-align: center;"> <input type="checkbox"/> yes <input type="checkbox"/> no </td> <td style="text-align: center;"> </td> <td style="text-align: center;">suche</td> </tr> <tr> <td style="text-align: center;"> <input checked="" type="checkbox"/> yes </td> <td style="text-align: center;"> <input type="checkbox"/> yes <input type="checkbox"/> no </td> <td style="text-align: center;"> </td> <td style="text-align: center;">suche</td> </tr> <tr> <td style="text-align: center;"> <input checked="" type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no </td> <td style="text-align: center;"> <input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no </td> <td style="text-align: center;"> </td> <td style="text-align: center;">Stop</td> </tr> </tbody> </table> </div>		Kara macht:	Nächster Zustand:					suche	<input checked="" type="checkbox"/> no <input type="checkbox"/> yes	<input type="checkbox"/> no		suche	<input checked="" type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> no		suche	<input checked="" type="checkbox"/> yes	<input type="checkbox"/> yes <input type="checkbox"/> no		suche	<input checked="" type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no		Stop
	Kara macht:	Nächster Zustand:																							
			suche																						
<input checked="" type="checkbox"/> no <input type="checkbox"/> yes	<input type="checkbox"/> no		suche																						
<input checked="" type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> no		suche																						
<input checked="" type="checkbox"/> yes	<input type="checkbox"/> yes <input type="checkbox"/> no		suche																						
<input checked="" type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no	<input type="checkbox"/> yes <input type="checkbox"/> or <input type="checkbox"/> no		Stop																						

Beschreiben Sie in zwei, drei Sätzen präzise, wie das Programm das Kleeblatt findet: Was ist die Idee hinter dem Programm? Ist das Programm effizient?

Programming: Arrays für Nicht-Informatiker erklärt

Wikipedia definiert Arrays (Felder) wie folgt ([http://de.wikipedia.org/wiki/Feld_\(Datentyp\)](http://de.wikipedia.org/wiki/Feld_(Datentyp))):

„Ein Feld (englisch Array [əˈjeɪ] (Betonung auf 2. Silbe) für ‚Anordnung‘, ‚Aufstellung‘, ‚Reihe‘, ‚Reihung‘, ‚Bereich‘) bezeichnet in der Informatik eine Datenstruktur. Mit Hilfe eines Feldes können Daten eines üblicherweise einheitlichen Datentyps so im Speicher eines Computers abgelegt werden, dass ein Zugriff auf die Daten über einen Index möglich wird. Das (Standard-)Feld verwendet einen ganzzahligen Index zur Adressierung.“

Das ist zwar prägnant, aber ziemlich abstrakt! Sie erinnern sich an die Arrays, die wir für Google Code Playground verwendet haben. Erklären Sie Ihren Schulkolleg/innen, die nicht das Ergänzungsfach Informatik besuchen, was Arrays sind. **Erklären Sie die Idee der Arrays anschaulich. Sie können dazu auch Skizzen verwenden:**