

IU Internationale Hochschule

Weiterbildung: "Software Engineering - Python"

Modul : DLMCSPSE01_D - Projekt: Software Engineering

Tutor: Prof. Dr. David Kuhlen



Prüfungsleistung: Portfolio

Spezifikation

Eingereicht am 11.06.2025.

Verfasser:

Djahan Bayrami Latran

Denninger Straße 198

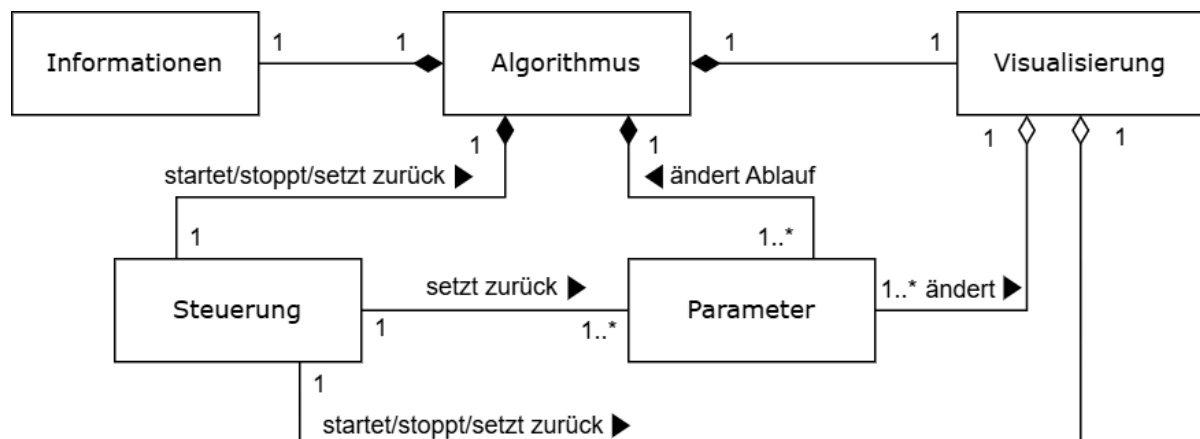
81927 München

E-Mail: djahan.latran@gmail.com

Matrikelnummer: UPS10672478

1. Datenmodell / Geschäftsobjekte:

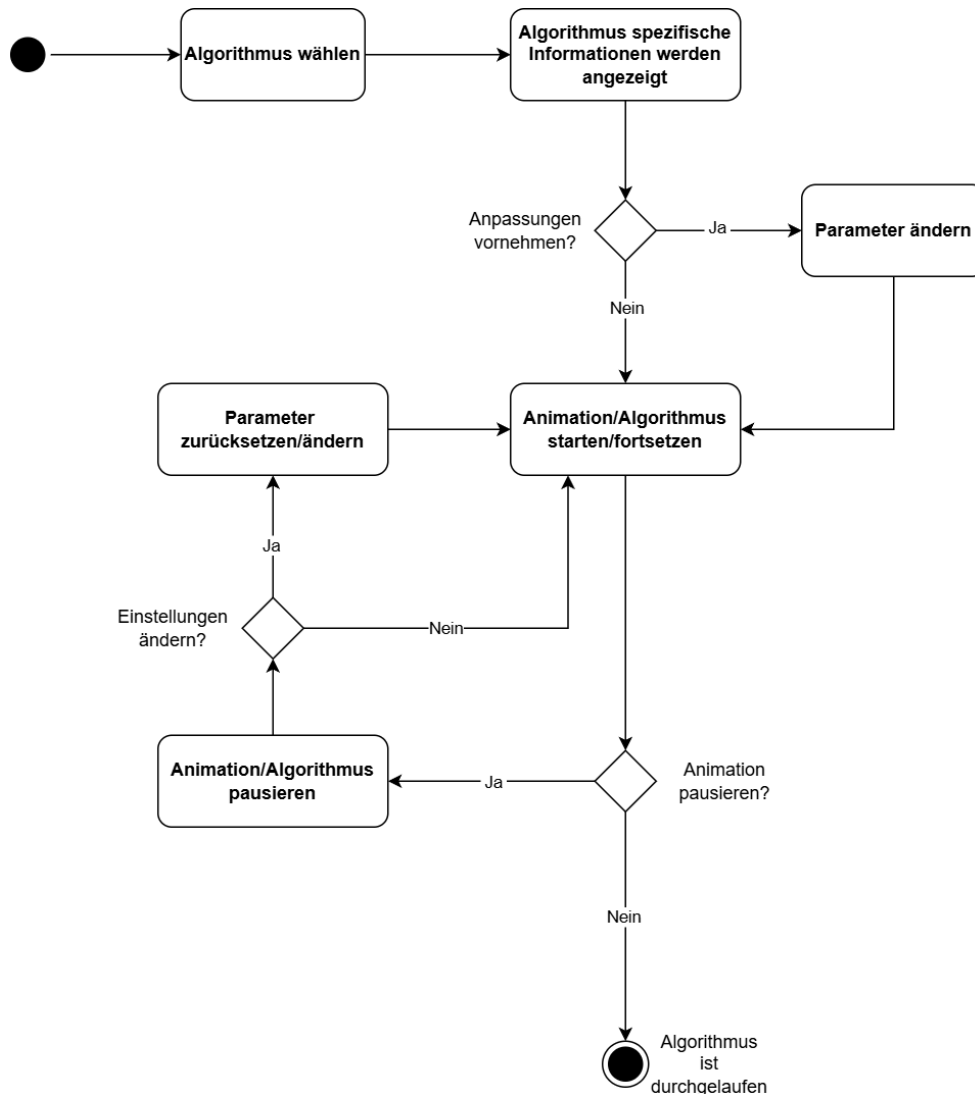
- **Nutzer:** Der Nutzer bedient die Oberfläche, wählt einen Algorithmus, verändert ggfs. Parameter und steuert die grafische Animation.
- **Grafische Darstellung/Animation:** Visualisiert den ausgewählten Algorithmus anhand einer zum Algorithmus passenden Darstellung
- **Algorithmus:** Einer der im Hauptmenü enthaltenen Algorithmen aus den Kategorien Such-/Sortier-/Pathfinding Algorithmen.
- **Informationen:** Zeigt die zusätzlichen Informationen zum jeweiligen Algorithmus, wie beispielsweise Definition in Form von Text, Zeitkomplexität und Code.
- **Eingabeparameter:** Parameter, die veränderlich sind, wie z.B. die Größe des Input-Arrays bei Sortier-Algorithmen, die Geschwindigkeit der Schritte des Algorithmus oder die Hindernisse bei Such-/Pathfinding Algorithmen.
- **Steuerung:** Beinhaltet die Funktionen, den Algorithmus zu starten, zu pausieren und die Standardeinstellungen wiederherzustellen.



2. Geschäftsprozesse:

- **Wahl des Algorithmus:** Der Nutzer wählt einen Algorithmus aus den vorhandenen Kategorien aus.
- **Eingabeparameter ändern:** Anschließend können die Eingabeparameter wie Geschwindigkeit und Inputgröße vom Nutzer verändert werden.
- **Algorithmus/Animation des Algorithmus starten:** Der Nutzer kann durch einen Button die grafische Animation bzw. den ausgewählten Algorithmus starten
- **Algorithmus/Animation des Algorithmus pausieren:** Während der Animation kann durch Klicken eines Buttons die Darstellung pausiert werden.

- **Standardeinstellungen wiederherstellen:** Die Standardeinstellung der Eingabeparameter können hiermit wiederhergestellt und der Zustand der grafischen Animation/des Algorithmus wieder zurückgesetzt werden.



3. Geschäftsregeln:

- **Wahl des Algorithmus:** Es kann nur einer der im Hauptmenü vorgegebenen Algorithmen gewählt werden. Erst nach Wahl des Algorithmus werden zusätzliche Fenster und Optionen angezeigt.
- **Info-Fenster:** Diese werden erst angezeigt, wenn ein Algorithmus aus der Liste ausgewählt wurde.
- **Eingabeparameter ändern:** Die Größe des Input Arrays, sowie die Geschwindigkeit der Verarbeitung für Such- und Sortieralgorithmen kann geändert werden, aber muss

sich immer zwischen einem festgelegten Minimum und Maximum befinden, um stets gute Performance zu gewährleisten.

- **Algorithmus starten:** Der Algorithmus kann erst gestartet werden, nachdem er aus der Liste gewählt wurde.
- **Algorithmus pausieren:** Der Algorithmus kann erst pausiert werden, nachdem er gestartet wurde.
- **Einstellungen/Algorithmus zurücksetzen:** Die Rückkehr zu den Standardeinstellungen ist erst möglich, nachdem diese verändert wurden, oder der Algorithmus gestartet wurde.

4. Benutzerschnittstellen:

GUI-Konzept:

The screenshot displays the AlgoLab application interface. On the left, a sidebar lists various algorithms under three categories: Search Algorithms (Linear Search, Binary Search), Sort Algorithms (Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort), and Pathfinding Algorithms (Astar Pathfinding, Dijkstra Pathfinding, Depth-First-Search, Breadth-First-Search). The 'Sort Algorithms' section is active, and 'Bubble Sort' is selected. The main area is divided into three panels. The top panel, 'Sort Algorithms: Bubble Sort', shows a 'Realtime Visualization' of an array of 15 numbers: 13, 21, 25, 27, 32, 34, 36, 40, 47, 7, 10, 12, 14, 16, 18. The bars are colored in a gradient from dark blue to light blue. The bottom panel, 'Code highlighting', shows the Python code for the bubble sort algorithm. The line 'if nums[j] > nums[j+1]:' is highlighted in blue. The right panel, 'Settings', contains two sliders: 'Adjust Array Size' and 'Adjust Speed'. Below the sliders are buttons for 'Reset' and a pause icon. At the bottom of the right panel, there is an information icon and a section titled 'Complexity' with the text: 'Best Case (already sorted): $O(n)$ ' and 'Average & Worst Case: $O(n^2)$ '. Below this is a 'Definition' section with a detailed explanation of the bubble sort algorithm.

AlgoLab

Sort Algorithms: Bubble Sort

Settings

Search Algorithms

- Linear Search
- Binary Search

Sort Algorithms

- Bubble Sort**
- Selection Sort
- Insertion Sort
- Merge Sort
- Quick Sort

Pathfinding Algorithms

- Astar Pathfinding
- Dijkstra Pathfinding
- Depth-First-Search
- Breadth-First-Search

Realtime Visualization

13 21 25 27 32 34 36 40 47 7 10 12 14 16 18

Code highlighting

```
def bubble_sort(nums):
    for i in range(len(nums), 0, -1):
        for j in range(i-1):
            if nums[j] > nums[j+1]:
                tmp = nums[j]
                nums[j] = nums[j+1]
                nums[j+1] = tmp
```

Adjust Array Size

Adjust Speed

Reset

Complexity

Best Case (already sorted): $O(n)$
Average & Worst Case: $O(n^2)$

Definition

Bubble Sort is a simple sorting algorithm that repeatedly compares adjacent elements and swaps them if they are in the wrong order. This process continues until no more swaps are needed, meaning the list is fully sorted. With each pass, the largest value "bubbles up" to its correct position on the right. The algorithm then repeats for the remaining unsorted elements until the entire list is ordered. Due to its inefficient time complexity of $O(n^2)$, Bubble Sort is primarily used for educational purposes rather than practical applications.

Bereiche:

The screenshot displays the AlgoLab interface with the following sections:

- Hauptmenü (Main Menu):** A sidebar menu on the left containing categories: Search Algorithms (Linear Search, Binary Search), Sort Algorithms (Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort), and Pathfinding Algorithms (Astar Pathfinding, Dijkstra Pathfinding, Depth-First-Search, Breadth-First-Search). The 'Bubble Sort' option is highlighted.
- Animationsfenster (Animation Window):** The central area titled 'Search Algorithms: Bubble Sort' showing a 'Realtime Visualization' of an array of bars. The bars are colored green and black, representing the current state of the array during the sorting process.
- Eingabeparameter (Input Parameters):** A settings panel on the right with sliders for 'Adjust Array Size' and 'Adjust Speed', and a 'Steuerung' (Control) section with buttons for 'Reset' and 'Pause'.
- Info-Bereiche (Info Areas):** A bottom panel containing:
 - Code highlighting:** A Python code snippet for the bubble sort algorithm:


```
def bubble_sort(nums):
    for i in range(len(nums), 0, -1):
        for j in range(i-1):
            if nums[j] > nums[j+1]:
                tmp = nums[j]
                nums[j] = nums[j+1]
                nums[j+1] = tmp
```
 - Complexity:** Text indicating 'Best Case (already sorted): $O(n)$ ' and 'Average & Worst Case: $O(n^2)$ '.
 - Definition:** A descriptive paragraph explaining the bubble sort algorithm and its educational purpose.

- **Hauptmenü:** Hier kann ein Algorithmus aus der Liste gewählt werden.
- **Algorithmus Kategorien:** Kategorien von unterschiedlichen Algorithmen
- **Animationsfenster/grafische Darstellung:** Zeigt die grafische Animation des Algorithmus.
- **Eingabeparameter Bereich:** Hier können Änderungen an Parametern getätigt werden
- **Steuerungsbereich:** Enthält Funktionen zum Starten und Pausieren des Algorithmus, sowie auf Standardparameter zurücksetzen.
- **Info-Fenster:** Enthält zusätzliche Informationen, wie Code, Beschreibungen, Infos zur Zeitkomplexität.