

# Portfolio

Mein Portfolio ist interaktiv und kann als abgeschlossenes, einzelnes Programm ausgeführt werden. Der Monsterkampfsimulator und die Vektorklasse können direkt im laufenden Programm im benötigten Umfang getestet werden.

Die Anweisungen an den AVL-Baum und an die (Sortable-) LinkedList sind aufgrund ihrer komplexeren Anwendungsmöglichkeiten nicht durch den Endnutzer ausführbar, sondern können nur direkt in den jeweiligen Testmethoden hinzugefügt, verändert oder entfernt werden.<sup>1</sup>

Dort sind einige Beispiele vorzufinden, welche die Grundlegende Bedienung der beiden Klassen veranschaulicht, damit man sich selbst ein Bild der Funktionalität machen kann, und ggf. Edge Cases u.Ä. testen kann.

Falls irgendwelche Fragen trotzdem auftreten, bin ich meistens unter 0174 3002950, oder [lesliejohnrichardson@gmail.com](mailto:lesliejohnrichardson@gmail.com), erreichbar.

Falls technische Probleme auftreten, hier ein paar Spezifikationen, um meinen "Versuchsaufbau" zu replizieren:

OS: Windows 10 Home 64-Bit

CPU: AMD-FX-8320E

GPU: Gigabyte Geforce GTX 760 4GB

RAM: 16GB DDR3

.NET Framework: 6.0

IDE: Visual Studio Community 2022 (64-Bit)

---

<sup>1</sup> Für alle geforderten Arbeitsproben ist nur die Program-Klasse notwendig. In dieser sind vier Testmethoden für die 4 Arbeitsproben vorhanden, in denen jeweils alles steht, was man wissen muss um diese zu starten/bedienen. Die Main-Methode muss dafür nicht verändert werden, da diese nur der Projekt(/Arbeitsproben)-Auswahl dient, welche im laufenden Programm erfolgt. Alles andere ist per sé nur Library, und soll bitte **nicht** zum Testen bearbeitet werden.

# Zeitmanagement Tabelle

	Monsterkam pf	Vektor Klasse	Linked List	Sortierer	AVL-Baum	Portfolio
Geplante Zeit	~1,5h	~2h	~4h	~3h	~2,5h	~5h
Tatsächl. Zeit	~4h	~2h	~2,5h	~3h	~3h	~2,5h

Ich habe außerdem, da ich selber nicht mehr die Zeit hatte zur übersichtlichen Darstellung der AVL-Bäume, eine Ausgabe-Methode zu erstellen, eine passende AVLTreePrinter-Klasse aus folgender Quelle bezogen und **nicht** selber programmiert, sondern lediglich auf schädlichen Code überprüft und angepasst, um mit meinem Projekt zu funktionieren:

1. Quelle "AVLTreePrinter.cs":

<https://stackoverflow.com/questions/36311991/c-sharp-display-a-binary-search-tree-in-console>