

Projekt zu Sortierverfahren

Ziel

Ziel ist es, ein komplettes Projekt zu programmieren und damit verschiedene Sortierverfahren zu implementieren. Am Ende sollen die Projekte in einer kurzen Präsentation vorgestellt werden.

1. Allgemeine Kriterien

- Auch zu Hause kann und soll weitergearbeitet werden!
- Am 17. Oktober ist die Abgabe des Projektes und finden die Präsentationen statt.
- Präsentiert auch Probleme, auf die ihr gestoßen seid und berichtet, wie ihr diese umgehen konntet.
- Die Präsentation sollte nicht länger als ca. 10-15 Minuten dauern.

2. MUSS-Kriterien

Die *MUSS-Kriterien* müssen auf jeden Fall erfüllt werden. Werden nur diese erfüllt, so liegt die Endnote im Bereich von ca. 7 Punkten.

- Eine **Sortieren**-Klasse mit **main**-Methode. Hier soll ein Array einer festen Länge mit Zufallszahlen befüllt werden und anschließend mit den Sortierverfahren sortiert werden.
- Das Sortierverfahren **SelectionSort** muss **in-place** programmiert werden.
- Das Sortierverfahren **InsertionSort** muss programmiert werden.
- Der Quellcode muss kommentiert werden.

3. SOLL-Kriterien

Werden zusätzlich noch die *SOLL-Kriterien* erfüllt, so liegt die Endnote in etwa bei 11 Punkten.

- **MergeSort** als Beispiel für die „divide-and-conquer“-Technik soll implementiert werden.
- **BubbleSort** soll programmiert werden.
- Es soll eine Zeitmessung programmiert werden um die Schnelligkeit der verschiedenen Sortierverfahren miteinander vergleichen zu können.
- Die *durchschnittlich* benötigte Zeit soll für alle Sortierverfahren für unterschiedlich große Arrays gemessen und präsentiert werden.

4. DARF-Kriterien

Werden **alle** Kriterien erfüllt, so liegt die Endnote bei etwa 15 Punkten.

- Einer der folgenden Algorithmen darf programmiert werden:
 - **HeapSort**
 - **QuickSort**
 - **TimSort**
 - oder ein selbst gewählter Sortieralgorithmus.
- Der Ablauf des Algorithmus muss dann auch präsentiert werden.
- Informiert euch und präsentiert, wann man von einem **stabilen** Sortierverfahren spricht.
- Begründet für alle Algorithmen, ob diese stabil oder nicht-stabil funktionieren.