Sortieren 1

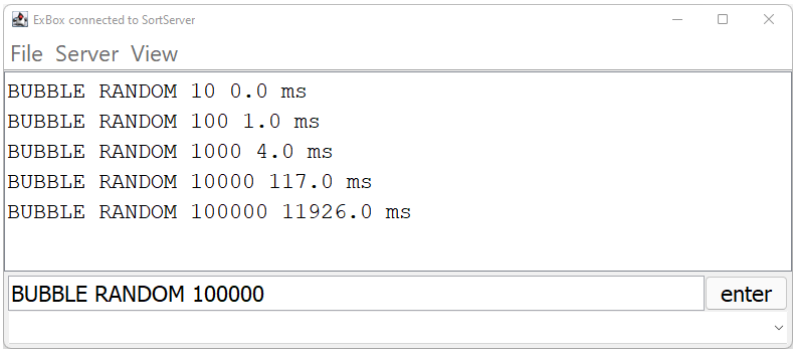
Die drei einfachen Sortierverfahren Bubble-Sort, Selection-Sort und Insertion-Sort sind alle für ungeordnete Datenbestände von der Ordnung $O(n^2)$. Dennoch unterscheiden sie sich in ihren Laufzeiten, was sich aus der Theorie ableiten lässt. Sie sollen in diesem Praktikum diese Unterschiede durch Laufzeit-Messungen quantifizieren.

Aufgabe 1: Bubble-Sort mit Testdaten

Erstellen Sie eine Klasse SortServer (Vorlage), die den CommandExecutor implementiert. Es sollen Datenbestände unterschiedlicher Grösse sortiert werden können, deren Schlüssel durch einen Array von Ganzzahlen mit Werten im Bereich [0..10'000'000[repräsentiert werden. Implementieren Sie den Bubble-Sort aus dem Skript. Die Anzahl der zu sortierenden Daten (Grösse des Arrays) soll als Parameter mitgegeben werden können. Schreiben Sie eine Methode isSorted, die überprüft, ob das Array korrekt sortiert wurde. Die Laufzeit der Sortierung soll das Resultat der execute-Methode sein:

Hinweis:

- Um die Laufzeit zu bestimmen, kann die Methode currentTimeMillis der Klasse System verwendet werden.
- Der execute Methode soll angegeben werden können, welcher Algorithmus angewandt werden soll (als Vorbereitung auf die nachfolgenden Aufgaben), wie das Test-Arry befüllt werden soll (RANDOM, ASC oder DESC) und wie viele Elemente das zu sortierende Array enthalten soll. Z.B. «BUBBLE RANDOM 10000».



Die Aufgabe muss nicht abgegeben werden.

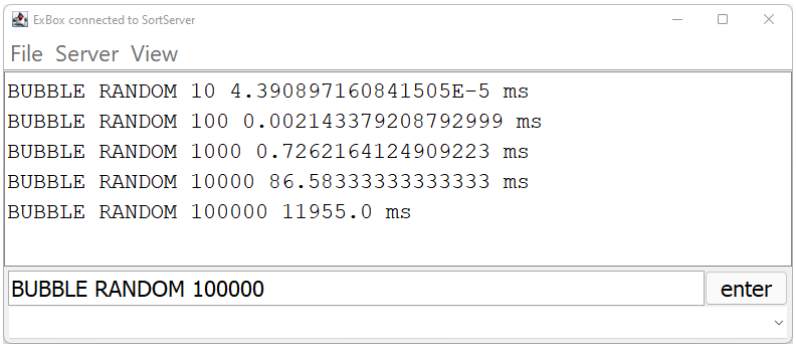
Aufgabe 2: Messen der Laufzeit von Bubble-Sort

Da lediglich in einer Auflösung von Millisekunden gemessen werden kann, muss für kleine Datenbestände der Aufruf der Sortiermethode mehrmals wiederholt werden. Da wir über einen grossen Bereich messen wollen, ist es praktischer, nicht die Anzahl Wiederholungen, sondern die Messzeit konstant zu halten. Ein weiteres Problem der Zeitmessung kann man folgendem Auszug aus der Java Dokumentation zu obiger Methode entnehmen: «The granularity of the value depends on the underlying operating system.».

Hinweis:

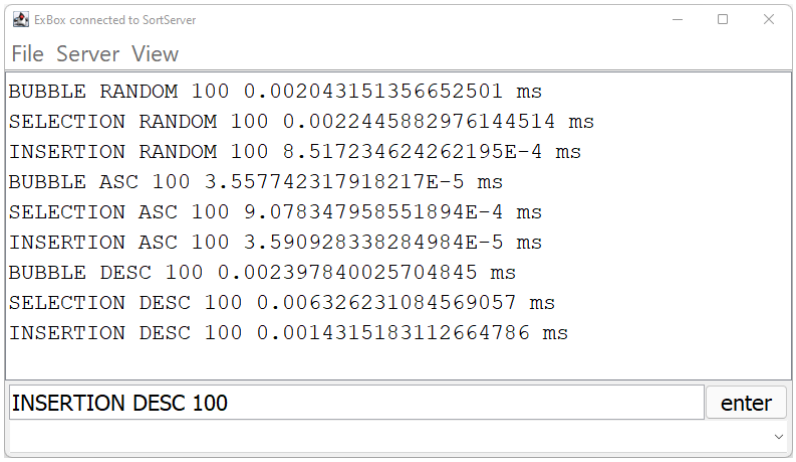
- Damit die Erstellung des Datenbestandes nicht die Messung verfälscht, muss dieser einmal erzeugt und anschliessend jeweils eine Kopie sortiert werden. Mittels der Methode arraycopy der Klasse System lässt sich ein Array schnell kopieren.
- Verwenden Sie folgendes Gerüst für Ihre Messroutine:

```
long startTime = System.currentTimeMillis()
long endTime = startTime;
int count = 0;
while (endTime < startTime + 1000) {
   <ArrayCopy>
   <Sortieren>
   count++;
   endTime = System.currentTimeMilliss();
}
elapsed = (double)(endTime - startTime) / count);
```



Aufgabe 3: Selection-Sort und Insertion-Sort

Es sollen die beiden anderen Sortieralgorithmen (Selection-Sort und Insertion-Sort) ebenfalls implementiert werden.



Aufgabe 4: Darstellung in Excel

Stellen Sie die erhaltenen Daten graphisch dar. Sie können die Resultate für diesen Zweck ins Excel übernehmen. Wählen Sie einen geeigneten Diagrammtyp.

Die Aufgabe muss nicht abgegeben werden.