

# PT2: Serie 6

Abgabetermin	10.07.2018, 21:00 Uhr
Übungstermin	05.07.2018

---

## Stabiler Quicksort

Implementieren Sie eine stabile Version von Quicksort für verkettete Listen. Verwenden Sie zum Vergleichen die Annahme, dass die Elemente in der Liste `java.lang.Comparable` konsistent implementieren. Verzichten Sie auf indexbasierte Zugriffe und vermeiden Sie unnötiges Anlegen und Löschen von Listenknoten.

Speichern Sie Ihre Listenimplementierung mitsamt einer Sortiermethode als

```
StableQuickSortableList.java
```

## Testanwendungen

1. Testen Sie Ihren Algorithmus mit `SortRandom.java`, in dem Sie eine zufällige Anzahl an Zufallsexemplaren von `java.lang.Integer` generieren, sortieren und deren Reihenfolge prüfen.
2. Die Anwendung `SortStdin.java`, soll die Zeilen der Standardeingabe sortieren und ausgeben, wie das Unixtool `sort`.
3. Ein Aufruf von `SortStdin -c N` soll die Eingabezeilen nach der n-ten Spalte sortieren (erster Index ist die 1). Die Spalten sind durch ein oder mehrere Whitespaces (Leerzeichen, Tabulatoren, etc.) getrennt (`String.split("\\s+)`). Diese Whitespaces sollen unverändert mit ausgegeben werden.

Hinweise:

- Ihr Programm soll sich äquivalent zu `sort -s -k N,N` unter Unix verhalten
- Sie dürfen davon ausgehen, dass jede Zeile die gleiche Anzahl Spalten hat.
- Es empfiehlt sich die Zeilen zuerst in ein eigenen Hilfsdatentyp zu überführen, der u.A.:
  - den Originaltext
  - die Spaltenwerte enthält
  - `java.lang.Comparable` implementiert
  - (optional) die Nummer der Schlüsselspalte enthält

Beispiele finden Sie im [Validatorskript](#).

## Code-Review

Führen Sie eine Codedurchsicht der Ihnen [zugeteilten Lösung der Serie 5](#) durch.

Das vorbereitete Archiv enthält alle Java-Dateien unter dem Ordner `src`, sowie das Ant-File (wenn vorhanden) einer der Abgaben aus der letzten Serie. Andere Dateien wurden entfernt.

Ihr Bericht soll festhalten welche systematische Überprüfungen Sie durchgeführt haben und welche Defekte entdeckt wurden. **Ihr Text wird den Originalautoren zur Verfügung gestellt.**

## Abgabe

Reichen Sie Ihre Lösung in Form eines gzip-komprimierten Tarfiles ein, welches sämtliche Klassen, sowie ein Ant-File zur Übersetzung mit `ant all` enthält. Ausgabedateien sollen im Ordner `build` erzeugt werden.

Wählen Sie als Dateinamen für die Codedurchsicht `review_vorname.nachname.txt` entsprechend. **Auch wenn Sie als Gruppe einreichen, muss jedes Gruppenmitglied einen eigenen Bericht erstellen.**

Es sollen keine Binärdateien, Validationsskripte, Editorbackups, lokale Repositories, o.Ä. enthalten sein.

```
/
|--build.xml
|--review_vorname.nachname.txt
|--[review_vorname2.nachname2.txt]
|--src
    |--SortRandom.java
    |--SortStdin.java
    |--StableQuickSortableList.java
```

## Zusatzaufgabe

Implementieren Sie Mergesort für verkettete Listen.