

## Abgabe 1

**Abgabe: 26.04.2020**

**Ziel der Aufgabe:** In dieser Programmieraufgabe sollten Sie folgende Algorithmen und Datenstrukturen in Java implementieren, passend zu den ersten fünf Kapiteln der Vorlesung:

- a) **BINARY SEARCH (MAX. 2 PUNKTE):** Implementieren Sie den Algorithmus für die binäre Suche nach einem gegebenen Element in einem Array.
- b) **SHELLSORT (MAX. 3 PUNKTE):** Implementieren Sie den Shellsort-Algorithmus, welcher In-Place (dh max  $O(1)$  zusätzlicher verwendeter Speicher) auf einem Array operiert.
- c) **QUICKSORT (MAX. 5 PUNKTE):** Implementieren Sie den Quicksort-Algorithmus. Achten Sie darauf, dass Ihre Implementierung stabil ist! *Hinweis: der klassische Quicksort-Algorithmus sortiert nicht stabil. Recherchieren bzw. überlegen Sie, wie Stabilität garantiert werden kann.*
- d) **MATRIZENMULTIPLIKATION (MAX. 5 PUNKTE):** Implementieren Sie den Strassen-Algorithmus zur Multiplikation von  $n \times n$ -Matrizen. Achtung: Der vollständige Strassen-Algorithmus würde die Eingabe-Matrizen solange rekursiv aufteilen, bis alle Matrizen nurmehr die Größe  $1 \times 1$  haben. Dadurch wird der Algorithmus allerdings sehr langsam. Wählen Sie zur Beschleunigung selbstständig eine geeignete Dimension  $k$  und verwenden Sie bei Matrizen, die kleiner als  $k \times k$  sind, die klassische Matrix-Multiplikation.

**Vorbereitung:** Laden Sie die in Moodle bereitgestellte AB1-ZIP-Datei herunter. Es wird das Build-Tool `gradle` verwendet, welches Sie beim Arbeiten und auch beim Erzeugen Ihrer hochzuladenden ZIP-Datei unterstützt. Die ZIP-Datei enthält folgende Bestandteile:

- Interface `Ab1` im Java-Paket `ab1`.
- “Leere” Implementierung (als Grundgerüst) des oben genannten `Ab1`-Interfaces im Paket `ab1.impl.Nachnamen` (im Folgenden *Implementierungspaket* genannt). Ihre Aufgabe besteht in der Implementierung des entsprechenden Interfaces.
- Eine Testklasse `Tests` im Paket `ab1.test`, mit der Sie Ihre Implementierungen testen können. Diese Testklasse (mit zusätzlichen Testfällen) wird auch zur Bewertung Ihrer Abgabe verwendet (siehe Punkt “Testen”).

**Integration in Ihre IDE (optional):** Sie können (zB in Eclipse bzw. IntelliJ) den Inhalt der ZIP-Datei als Gradle-Projekt öffnen. Ihre IDE hat damit alle nötigen Informationen.

**Durchführung:** Die Programmieraufgabe kann in Gruppen bis zu 3 Personen bearbeitet werden. Alle Namen müssen in der Abgabe enthalten sein (siehe unten). Bei mehr als 3 Personen werden die erreichten

Punkte anteilmäßig pro Person reduziert. Die geforderten Algorithmen sind im Implementierungspaket (impl) ohne Verwendung von Java-Bibliotheken und Systempaketen (außer `java.lang`) zu implementieren. Insbesondere darf das `java.util` Paket nicht verwendet werden. An den vorgegebenen Interfaces und an der Testklasse dürfen, bis auf die Umbenennung Ihres Implementierungspakets, keine Änderungen vorgenommen werden. Ändern Sie ausschließlich die Datei "Ab1Impl.java" (bzw., wenn benötigt, können auch neue Dateien im Ordner "ab1/impl" erstellt werden). Bei Unklarheiten bezüglich der Aufgabenstellung kontaktieren Sie bitte Ihren LV-Leiter.

**Testen:** Rufen Sie `./gradlew test` bzw. `gradlew.bat test` auf um die Testfälle durchlaufen zu lassen. Alternativ können Sie die Tests auch in Ihrer Entwicklungsumgebung laufen lassen. Zum Bewerten Ihrer Abgabe werden zusätzliche, nicht ausgegebene Testfälle verwendet. Es ist daher von Vorteil, wenn Sie selbst zusätzliche Testfälle implementieren, und Ihre Lösung damit testen.

**Abgabe:** Rufen sie `./gradlew zip` bzw. `gradlew.bat zip` auf. Sie finden nun die hochladbare ZIP-Datei im Ordner `build`. Geben Sie die ZIP-Datei via Moodle ab. Pro Gruppe ist nur eine Abgabe nötig. Benennen Sie vor der Abgabe das Paket `ab1.impl.Nachnamen` entsprechend um, indem Sie `Nachnamen` durch die Nachnamen Ihrer Gruppenmitglieder ersetzen (z.B. `ab1.impl.HuberMeierMueller`).

**Bewertung:** Diese Programmieraufgabe wird mit maximal 15 Punkten bewertet. Die erreichte Punktzahl wird allen Gruppenmitgliedern gleichermaßen angerechnet. Die Bewertung ergibt sich aus folgenden Komponenten:

- Funktionalität Ihres Codes. Die Anzahl erfolgreicher Unit-Tests sind hierfür ausschlaggebend. Beachten Sie, dass für die Bewertung nicht nur die Testfälle der ausgegebenen Testklasse verwendet werden, sondern auch zusätzliche, erweiterte Testfälle. Beim Testen wird von Ihrer ZIP-Datei wird nur das Implementierungspaket verwendet.
- Qualität Ihres Quellcodes (Inspektion). Achten Sie auf kommentierten, gut verständlichen und effizienten Quellcode. Die Verwendung von `System.out` ist nicht gestattet.
- Ehrlichkeit: Falls Sie unzulässige Java-Bibliotheken (siehe oben) verwenden, oder Sie den selben Quellcode wie eine andere Gruppe abgeben, kann Ihre Abgabe mit 0 Punkten bewertet werden.
- Formatierung: Formatieren Sie Ihren Code (Standard-Formatierung Ihrer IDE ist ausreichend) und achten Sie darauf, dass sie keine nicht verwendeten Imports haben.