

Abgabe 2

Abgabe: 02.06.2019

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

- QUEUE:** Implementieren Sie eine Warteschlange, die Elemente nach dem LIFO- und FIFO-Prinzip speichert und ausgibt. Implementieren und verwenden Sie das jeweils vorgegebene Interface.
- HASH SET:** Implementieren Sie ein Hash Set gemäß dem vorgegebenen Interface. Achten Sie darauf, dass die Einfüge- und Such-Operationen in $O(1)$ arbeiten (unter der Annahme eines perfekten Hash-Verfahrens).
- SORTED BINARY TREE:** Implementieren Sie eine Set-Datenstruktur in Form eines sortierten Binärbaumes gemäß dem vorgegebenen Interface. Achten Sie darauf, dass die vorgegebenen Funktionen so implementiert sind, dass Sie auch im schlechtesten Fall korrekt funktionieren (z.B. keine Stack Overflows wenn viele Elemente in aufsteigender Reihenfolge eingefügt werden).

Vorbereitung: Laden Sie die in Moodle bereitgestellte AB2-ZIP-Datei herunter. Sie enthält eine JUnit-JAR-Datei ¹ im `lib` Ordner sowie folgende Java-Dateien im `src` Ordner:

- `AuDQueue`, `AuDHashSet`, `AuDSortedTree`, und `Ab2` im Java-Paket `ab2`.
- “Leere” Implementierung des oben genannten `Ab1`-Interfaces im Paket `ab2.impl.Nachnamen` (im Folgenden *Implementierungspaket* genannt). Ersetzen Sie `Nachnamen` durch die Namen Ihrer Gruppenmitglieder (z.B. `ab2.impl.HuberMeierMueller`). Ihre Aufgabe besteht in der Implementierung der entsprechenden Interfaces und Klassen.
- Eine JUnit-Testklasse `Tests` im Paket `ab2.test`, mit der Sie Ihre Implementierungen testen können. Diese Testklasse (mit zusätzlichen Testfällen) wird auch zur Bewertung Ihrer Abgabe verwendet. Die Ausführung von JUnit-Tests in einer integrierten Entwicklungsumgebung (z.B. Eclipse) wird vorab in einer UE-Einheit demonstriert.

Durchführung: Die Programmieraufgabe kann in Gruppen bis zu 3 Personen bearbeitet werden. Die geforderten Algorithmen sind im Implementierungspaket ohne Verwendung von Java-Bibliotheken und Systempaketen (außer `java.lang`) zu implementieren. Insbesondere darf das `java.util` Paket nicht verwendet werden (bis auf vorgegebene Ausnahmen). An den vorgegebenen Interfaces und an der Testklasse sollten, bis auf die Umbenennung Ihres Implementierungspakets, keine Änderungen vorgenommen werden. Ihr Code muss innerhalb einer Standard-JVM-Instanz korrekt funktionieren (wir verwenden

¹in Eclipse ist das händische Einbinden nicht nötig, im Editor wird bei JUnit-Imports eine Option auf Integration der JUnit-Bibliothek vorgeschlagen

beim Testen keine zusätzlichen Parameter wie z.B. veränderte Stack-Größen oder maximalen Speicher). Bei Unklarheiten bezüglich der Aufgabenstellung kontaktieren Sie bitte Ihren LV-Leiter.

Abgabe: Packen Sie Ihren Quellcodeordner (`src`) in eine ZIP-Datei und geben Sie diese in Moodle ab. Pro Gruppe ist nur eine Abgabe nötig.

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 JUnit-Tests sind hierfür ausschlaggebend. Beachten Sie, dass für die Bewertung nicht nur die Testfälle der ausgegebenen Testklasse verwendet werden. Beim Testen wird von Ihrer ZIP-Datei 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.