

Abgabe 1

Abgabe: 09.01.2024

Ziel der Aufgabe: In dieser praktischen Programmieraufgabe sind die vorgegebenen Java-Interfaces zu implementieren, gemäß der in den Java-Kommentaren spezifizierten Beschreibung:

- a) **NFAFactory:** Dieses Interface dient zur Erzeugung von NFAs.
- b) **NFA:** Dieses Interface spezifiziert die Implementierung eines nicht-deterministischen endlichen Automaten.

Es gibt zwei Sets von Unit tests: öffentliche und versteckte. Die öffentlichen Tests werden mit der Abgabe mitgeliefert und können lokal zum Verifizieren der Abgabe herangezogen werden. Wenn alle Tests korrekt durchlaufen gibt das **5 Punkte**. Versteckte Tests sind Erweiterungen zu den öffentlichen und prüfen die Abgabe tiefergehend. Diese sind ebenfalls **5 Punkte** wert.

Die letzten 5 Punkte bewerten die Code Qualität. Hier fallen Code smells, unbenutzte Variablen, Code Duplikate, etc. ins Gewicht.

Gewichtung der Testklassen:

- a) **LOGICTESTS:** 2 Punkt
- b) **SIMPLETESTS:** 0.5 Punkte
- c) **FINALIZETESTS:** 0.5 Punkte
- d) **COMPLEX TESTS:** 2 Punkte

Vorbereitung: Laden Sie die in Moodle bereitgestellte Ab1-ZIP-Datei herunter. Es wird das Build-Tool **Maven** verwendet, welches Sie beim Arbeiten und auch beim Erzeugen Ihrer hochzuladenden ZIP-Datei unterstützt. Als Java Version wird 17 oder höher vorausgesetzt. Die ZIP-Datei enthält folgende Bestandteile:

- Die Interfaces **NFA** und **NFAFactory** im Java-Paket **ab1**.
- “Leere” Implementierung (als Grundgerüst) des oben genannten **NFA**-Interfaces im Paket **ab1.impl.GRUPE** (im Folgenden *Implementierungspaket* genannt). Ihre Aufgabe besteht in der Implementierung des entsprechenden Interfaces.
- Ein Testpackage **ab1.tests**, 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 Maven-Projekt importieren. Ihre IDE hat damit alle nötigen Informationen.

Durchführung: Die Programmieraufgabe kann in Gruppen bis zu 3 Personen bearbeitet werden. Alle Namen und die Gruppennummer müssen in der Abgabe enthalten sein (siehe Punkte “Abgabe” unten). Die geforderten Algorithmen sind im Implementierungspaket (impl) zu implementieren. Hierfür stehen Ihnen alle Standard-Java-Bibliotheken (insbesondere z.B. `java.util`) zur Verfügung. An den vorgegebenen Interfaces und an der Testklasse dürfen, bis auf die Umbenennung Ihres Implementierungspakets, keine Änderungen vorgenommen werden. Ändern Sie ausschließlich die Datei “NFAImpl.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 `./mvn 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: Erstellen Sie eine zip Datei des Projekts und laden Sie diese im Moodle hoch. Pro Gruppe ist nur eine Abgabe nötig. Benennen Sie vor der Abgabe das Paket `ab1.impl.GRUPPE` entsprechend um, indem Sie `GRUPPE` durch die Gruppennummer und die Nachnamen Ihrer Gruppenmitglieder ersetzen (z.B. `ab1.impl.gruppe1_huber_mueller`).

Bewertung: Diese Programmieraufgabe wird mit maximal 15 Punkten bewertet. Die erreichte Punktezahl wird allen Gruppenmitgliedern gleichermaßen angerechnet.

- 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 aus 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.
- Formatierung: Formatieren Sie Ihren Code (Standard-Formatierung Ihrer IDE ist ausreichend) und achten Sie darauf, dass sie keine nicht verwendeten Imports verwenden.