Algorithmen und Datenstrukturen

Dr. M. Lüthi, Dr. G. Röger
Frühjahrssemester 2018

Universität Basel
Fachbereich Informatik

Übungsblatt 4

Abgabe: 30. März 2018

Für dieses Übungsblatt benötigen Sie eine Java Umgebung, wie in Übungsblatt 1 beschrieben. Das kompilieren und ausführen der Testfälle erfolgt wie in den vorhergehenden Übungen durch `gradlew build` respektive `gradlew test`.

Als Grundlage für Ihre Implementation verwenden Sie das Projekt im Zip-File "blatt04.zip", welches Sie auf dem Adam-Workspace finden. Schreiben Sie ihre Lösungen in die dafür vorbereiteten Dateien und laden sie das Projekt mit Ihren Lösungen auf Courses hoch.

Aufgabe 4.1 (Operationen auf Stapel und Warteschlangen, 1 + 1 Punkte)

(a) Angenommen ein Client führt eine abwechselnde Folge von Push- und Pop-Operationen (auf einem Stapel) aus. Die Push-Operationen legen die ganzen Zahlen von 0-9 der Reihenfolge nach auf dem Stapel ab. Die Pop-Operationen geben den Rückgabewert aus. Welche Sequenzen von Zahlen können erzeugt werden und welche nicht. Wenn die Sequenz erzeugt werden kann, geben Sie die Operationen an um diese zu erzeugen. Schreiben Sie ihre Lösung in die Datei `stack-sequence.txt`, die sie im heruntergeladenen Projekt finden.

   (a) 4 3 2 1 0 9 8 7 6 5
   (b) 4 6 8 7 5 3 2 9 0 1
   (c) 2 5 6 7 4 8 9 3 1 0
   (d) 4 3 2 1 0 5 6 7 8 9

(b) Angenommen, ein Client führt abwechselnd enqueue- und dequeue-Operationen (für Warteschlangen) durch. Die enqueue-Operationen legen die ganzen Zahlen von 0 bis 9 der Reihe nach in der Warteschlange ab. Die dequeue-Operationen geben den Rückgabewert aus. Welche Sequenzen von Zahlen können erzeugt werden und welche nicht. Wenn die Sequenz erzeugt werden kann, geben Sie die Operationen an um diese zu erzeugen. Schreiben Sie ihre Lösung in die Datei `queue-sequence.txt`, die sie im heruntergeladenen Projekt finden.

   (a) 0 1 2 3 4 5 6 7 8 9
   (b) 4 6 8 7 5 3 2 9 0 1
   (c) 2 5 6 7 4 8 9 3 1 0
   (d) 4 3 2 1 0 5 6 7 8 9

Aufgabe 4.2 (Nutzung des Stack Interface (2 Punkte))

In dieser Aufgabe nutzen Sie die Klasse `java.util.Stack` aus der Java Standardbibliothek. Implementieren sie einen Stack-Client der mithilfe eines Stapels prüft, ob in einem gegebenen String die Klammersetzung korrekt ist. Implementieren Sie diesen in der Methode `checkParentheses` der Klasse *Parentheses*. Sie können Ihren Algorithmus testen, indem Sie die dazugehörigen Testfälle ausführen, die sie in der Klasse `ParenthesesTests` finden.

Aufgabe 4.3 (Verkettete Listen (4 Punkte))

Vervollständigen Sie die Implementation der Klasse LinkedList. Die fehlenden Implementationen erkennen Sie daran, dass diese eine *NotImplementedException* werfen. Sie können Ihre Implementation testen, indem Sie die dazugehörigen Testfälle ausführen, die sie in der Klasse `LinkedListTests` finden.

Aufgabe 4.4 (Stack und Queue Implementation mittels Verketteter Listen (1 + 1 Punkte))

Implementieren Sie die fehlenden Methoden in der Stack und Queue Klasse mithilfe einer Linked List. Die fehlenden Implementationen erkennen Sie daran, dass sie eine *NotImplementedException* werfen. Verwenden Sie dazu ihre eigene Implementation (von Aufgabe 4). Falls Sie diese Aufgabe nicht lösen konnten, können Sie stattdessen die Klasse `java.util.LinkedList` der Java Standardbibliothek verwenden. Sie können Ihre Implementation testen, indem Sie die dazugehörigen Testfälle ausführen, die sie in der Klasse `QueueTests` beziehungsweise `StackTests` finden.