

Aufgabenblatt 8 vom 20. Juni 2021, Abgabe am 04. Juli 2021, 22:00 Uhr

Aufgabe 8.1: Hashing

Punkte siehe StudOn
Hashing

Aufgabenstellung und Abgabe (individuell, nicht als Gruppe!) im StudOn.

Aufgabe 8.2: Stacks und Queues

Punkte siehe StudOn
Stacks, Queues

Aufgabenstellung und Abgabe (individuell, nicht als Gruppe!) im StudOn.

Programmieraufgaben

Stacks (Stapel) bzw. Queues (Warteschlangen) sind zwei der elementarsten Datenstrukturen der Informatik. Beide repräsentieren eine Ansammlung von Elementen, wobei beim Herausnehmen eines Elements jeweils das zuletzt (LIFO, Last-In-First-Out) bzw. zuerst (FIFO, First-In-First-Out) hinzugefügte zurückgegeben wird.

In den folgenden beiden Aufgabe sollen Sie diese Datenstrukturen implementieren. Beide sollen Ganzzahlen speichern können. Wir stellen Ihnen dafür in der Datei `08-material.zip` im StudOn zwei Interfaces `AuD{Stack,Queue}Interface` bereit, die Sie jeweils in einer eigenen Klasse `AuD{Stack,Queue}` anhand ihrer Dokumentation implementieren sollen. Die Dokumentation ist an die der entsprechenden Datenstrukturen der Java-API angelehnt.

Achtung: Verzichten Sie in Ihrer Implementierung auf die Verwendung von Klassen aus der Java-API, mit Ausnahme von `String` und `NoSuchElementException`. Attribute primitiver Typen, die Sie zur »Verwaltung« des internen Zustands benötigen, sind natürlich erlaubt.

Aufgabe 8.3: Stapel

10 Punkte
Stacks

In dieser Aufgabe sollen Sie `AuDStack` implementieren.

1. Erstellen Sie ein neues Projekt `08-Stacks`.
2. Der Konstruktor der Klassen `AuDStack` soll als einzigen Parameter die gewünschte Kapazität erhalten.
3. Verwenden Sie zur Realisierung des Stacks intern ein `int`-Array mit einer konstanten Größe

CAPACITY (wie in Abbildung 1). Die genaue Funktionsweise der einzelnen Methoden entnehmen Sie den JavaDoc-Kommentaren im Interface. Damit wir bei der Korrektur Ihren Code korrekt prüfen können, verwenden Sie bitte die Bezeichnungen **buffer** für das Array und **top** wie in der Abbildung dargestellt. Der Index **top** gibt also die Stelle an, an welcher das nächste Element eingefügt werden soll.

4. Geben Sie die Datei **AuDStack.java** im EST ab!

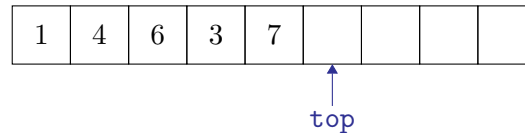


Abbildung 1: Implementierung eines Stacks als Array.

Aufgabe 8.4: Warteschlangen

12 Punkte
Queues

In dieser Aufgabe sollen Sie **AuDQueue** implementieren.

1. Erstellen Sie ein neues Projekt **08-Queues**.
2. Der Konstruktor der Klassen **AuDQueue** soll als einzigen Parameter die gewünschte Kapazität erhalten.
3. Verwenden Sie zur Realisierung der Queue intern ein **int**-Array mit einer konstanten Größe **CAPACITY**, welches Sie diesmal jedoch als Ringpuffer verwalten (wie in Abbildung 2). Die Spezifikation der Methoden entnehmen Sie wieder der Dokumentation. Ringpuffer werden in Kapitel 11 der Vorlesungsfolien beschrieben. Damit wir bei der Korrektur Ihren Code korrekt prüfen können, verwenden Sie bitte die Bezeichnungen **buffer** für das Array, sowie **first** und **last** wie in der Abbildung dargestellt. An Stelle **first** soll das nächste Element eingefügt werden; **last** gibt entsprechend den Index des ältesten Elements an (das Element, das als nächstes zurückgegeben werden soll).
4. Beachten Sie, dass der Ringpuffer erst voll ist, wenn er **CAPACITY** viele Elemente enthält und nicht, wenn **first** das Ende des Arrays erreicht hat. Sobald **first** am Ende angekommen ist, soll **first** wieder an Position 0 verschoben werden, sofern dies möglich ist (daher der Name Ringpuffer). Das gleiche Verhalten gilt für **last**.
5. Geben Sie die Datei **AuDQueue.java** im EST ab!

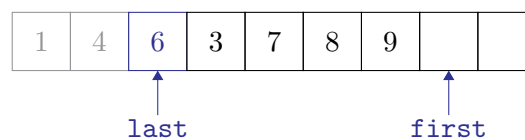


Abbildung 2: Implementierung einer Queue als Array.

Sollte Ihr Programm nicht übersetz- bzw. ausführbar sein, wird die Lösung mit 0 Punkten bewertet. Stellen Sie also sicher, dass IntelliJ IDEA keine Fehler in Ihrem Programm anzeigt, Ihr Programm übersetz- und ausführbar ist sowie die in der Aufgabenstellung vorgegebenen Namen und Schnittstellen *exakt* eingehalten werden. Geben Sie am Schluss die Dateien **AuDStack.java** und **AuDQueue.java** über die EST-Webseite ab. Wenn Sie die Aufgabe zusammen mit einem Übungspartner bearbeitet haben, geben Sie im EST unbedingt dessen Gruppenabgabe-Code an! Kontrollieren Sie, ob Ihre Namen am Anfang aller Dateien angegeben sind – schreiben Sie im Quellcode Ihre Angaben in einen Kommentar. Im EST-Abgabesystem können Sie modifizierte Dateien mehrfach abgeben. Nur die zuletzt hochgeladene Version wird bewertet.