

Arbeitsblatt: Hash Tables – Open Addressing

Anhand dieses Arbeitsblatts sollen Sie die Konzepte von *Open Addressing* umsetzen und einige Feinheiten der Implementierung von *HashSets* in Java studieren.

HashSets implementieren das bekannte Java-Interface *Set* mittels Hashing. Im Wesentlichen kann man Elemente in eine solche Menge einfügen (einmalig, d.h. es muss geprüft werden, ob ein gleiches Element schon vorhanden ist), prüfen, ob ein bestimmtes Element in der Menge ist und Elemente entfernen. Im Gegensatz zu *HashMaps* werden zu den Schlüssel-Elementen keine weiteren Objekte in der Datenstruktur abgelegt.

Vorbereitetes Programmier-Projekt

Importieren Sie das Projekt *ch.fhnw.algd2.hashset* vom Datei-Server. Im Projekt enthalten ist ein vorbereitete Klasse *MyHashSet*, die Ihnen eine Reihe von Aufgaben abnimmt, weil sie nicht mit *Hashing* zu tun haben.

Sie finden die zu implementierende Klasse *MyHashSet* im *src*-Ordner. Im *test*-Ordner finden Sie eine Reihe vorbereiteter JUnit-Tests zur Überprüfung Ihrer Implementierung.

Es brauchen nicht mehr Elemente abgelegt werden als initial (im Konstruktor) angegeben wurden. Sie brauchen also *kein Rehashing* zu implementieren. Wenn ein Element eingefügt werden müsste, aber kein Platz mehr dafür vorhanden ist, soll die *add*-Methode eine *IllegalStateException* werfen (in Übereinstimmung mit der Spezifikation im *Collection*-Interface).

Es sollen keine *null*-Werte in der Datenstruktur abgelegt werden können. Das macht die Implementierung einfacher, setzt aber entsprechende Vorbedingungen-Tests voraus.

Entscheiden Sie sich für ein *Open Addressing* Kollisionsbehandlungsverfahren und begründen Sie Ihren Entscheid.

Gehen Sie wie folgt vor:

1. Implementieren Sie den Konstruktor fertig. Als Parameter erhält dieser eine Mindestgrösse der zu erzeugenden Tabelle. Je nach Sondierungs-Schema sollte aber die Tabelle grösser gewählt werden, um die entsprechenden Voraussetzungen zu erfüllen. Testen Sie Ihre Implementation mit der Unit-Test Klasse *ArrayinitializationTest* (nicht erforderlich bei *Linear Probing*).
2. Vervollständigen Sie die Methode *add(E element)* entsprechend dem von Ihnen gewählten Sondierungs-Verfahren. *null*-Werte sind nicht erlaubt und sollen zu einer Exception führen. Testen Sie Ihre Implementation mit der Unit-Test Klasse *AddElementTest*.
3. Vervollständigen Sie die *contains(Object o)* Methode. Testen Sie Ihre Implementation mit der Unit-Test Klasse *ContainsElementTest*.
4. Vervollständigen Sie die *remove(Object o)* Methode, welche es ermöglicht Elemente zu entfernen. Welche Anpassungen brauchen allenfalls die Methoden *add* und *contains*? Testen Sie Ihre Implementation mit der Testklasse *RemoveElementTest*.