

Übungen zu Informatik B

Sommersemester 2018

Blatt 5

Aufgabe 5.1: Sequenzdiagramme (25 Punkte)

Betrachten Sie das Programm `UseLibrary`, das die Implementation des Klassendiagrammes vom vorherigen Aufgabenblatt benutzen soll. Geben Sie den Programmablauf in einem Sequenzdiagramm wieder. Beschränken Sie sich dabei auf die Darstellung der aufrufenden Klasse sowie der Instanzen `lib`, `li`, `searchResult` und der `List inventory` aus `lib`. Zum Zeichnen können Sie ein beliebiges Programm, wie beispielsweise das auf den CIP-Pool-Rechnern installierte DIA (<http://www.dia-installer.de>) oder Visual Paradigm (<https://ap.visual-paradigm.com/university-of-osnabrueck>) verwenden. Geben Sie Ihre Lösung auch schriftlich ab.

Aufgabe 5.2: Cloneable und clone (30 Punkte)

Betrachten Sie die Klassen `List` und `Entry`, mit denen der ADT Liste implementiert wurde.

Erzeugen Sie auf Basis dieser Klassen eine typsichere Liste. Begründen Sie Ihrem Tutor, ob es sinnvoller ist, von der bestehenden Klasse zu erben und nur die nötigsten Methoden zu überschreiben, oder die Liste vollständig neu zu implementieren.

Lesen Sie außerdem die Java - Dokumentation zu dem Interface `Cloneable` und der Methode `Object.clone()`. Implementieren Sie das Interface `Cloneable` in Ihrer generischen Liste.

Testen Sie anschließend automatisiert in einer eigenen Testklasse, ob Sie die Methode `clone()` korrekt implementiert haben. Achten Sie darauf, auch auf die nicht absoluten Anforderungen an eine `clone()` - Implementation zu testen. Begründen Sie, wenn Ihre Implementation bewusst einige dieser Anforderungen nicht erfüllt.

Aufgabe 5.3: Offenes Hashing (45 Punkte)

Implementieren Sie eine Klasse in der Objekte typsicher nach dem Prinzip des *offenen* Hashings abgelegt werden können. Die Klasse soll das gegebene Interface `HashSet` implementieren und alle vorgegeben Methoden vollständig umsetzen. Der Hash-Wert und die Gleichheit von Objekten soll entweder über die Methode `equals(Object)` und `hashCode()` aus der Klasse `Object` oder über die Methoden `equals(Object, Object)` und `hashCode(Object)` einer bei der Instanziierung des `HashSet` gegebenen Instanz vom Typ `HashFunction` erfolgen.

Beim offenen Hashing werden alle Objekte in ein Array mit vorher fest definierter Länge bezüglich Ihres Hash-Wertes abgelegt. Ein Objekt mit einem Hash-Wert von 5633 würde also an Position 3 eines Arrays mit der Länge 10 abgelegt. An jeder Stelle des Arrays befindet sich somit eine einfach verzeigte Liste in der alle Objekte mit der gleichen Position bezüglich ihres Hash-Wertes abgelegt werden.

Beim Suchen und Löschen werden alle Einträge in dem entsprechenden Feld des Array durchlaufen, bis das gesuchte, bzw. das zu löschende Objekt gefunden oder die Liste vollständig durchlaufen wurde.

Schreiben Sie eine Testklasse, die die grundlegenden Funktionen des `HashSet` automatisch testet. Sie können sich sehr viel Arbeit ersparen, wenn Sie zuerst die Testklasse implementieren und dann Ihr `HashSet` solange verändern, bis die Testklasse nicht mehr fehlschlägt. Sie können die Testklasse besser kontrollieren, wenn Sie einen extra Typen implementieren, bei dem Sie den Hash-Wert festlegen können.