# Löschen

Die Ausarbeitung unseres Löschens wird beispielhaft an der folgenden Lagerstruktur erklärt.

* Lager 1
  + Lager 1.1
    - Lager 1.1.1
  + Lager 1.2
    - Lager 1.2.1
    - Lager 1.2.2
* Lager 2

# Wird ein Lager mit Unterlagern gelöscht wandern alle Lager eine Ebene nach oben. In diesem Beispiel wären das die Lager 1, 1.1 und 1.2.

Wird ein Lager, welches keine Unterlager besitzt, gelöscht, dann gibt es drei verschieden Konstellationen. Besitzt das Lager ebenfalls kein Oberlager, muss es leer sein, um gelöscht werden zu können. In diesem Beispiel wären das die Lager 2. Sind auf der Ebene des zu löschende Lagers keine anderen Lager so wird es gelöscht und die Anteile der Buchungen und der Bestand auf das Lager darüber übertragen. In diesem Beispiel wären das die Lager 1.1.1. Sind stattdessen auf der gleichen Ebene noch weitere Lager kann das Lager nur gelöscht werden wenn das Lager leer ist. Dann werden der Bestand und die Anteile ebenfalls auf das Lager darüber übertragen. In diesem Beispiel wären das die Lager 1.2.1 oder 1.2.2.

# Besonderheiten

Unsere Gruppe hat als dreier-Gruppe auch die Aufgaben für vierer-Gruppen bearbeitet. So ist es mit unserer Lagerverwaltung möglich beliebig viele Lager an jeder Stelle zu erstellen. Des Weiteren ist es möglich Lager zu löschen. Dafür sind jedoch bestimmte Voraussetzungen zu erfüllen. Außerdem ist es möglich den aktuellen Stand der Lagerverwaltung zu speichern und einen gespeicherten Stand wieder zu laden.

# Commad

Für die Realisierung eines Redo- und Undo-Mechanismus wird das Command Pattern verwendet. Der Controller verwaltet zwei Stacks. Ein Stack für die Redo-Commands und einen für die Undo-Commands. Der Command besitzt eine Execute- und eine Undo-Methode. Der Controller führt die Methoden aus, um einen Anteil zu erzeugen bzw. so einen zu löschen. Wird ein Command erzeugt wird der Redo-Stack gelöscht.

# Test

Es wurden JUnit-Tests erstellt. Diese Tests gewährleisten die Kernfunktionalitäten der Models. Viele Methoden werden implizit durch den Test von Funktionalität des Lagerverwaltungsmodels getestet.  
Das Programm wurde ausführlich getestet. Es wurden Akzeptanztests uns Oberflächentests durchgeführt.