

Subtraktion von Listen (2020 W7)

Auf dem letzten Übungsblatt haben Sie eine Linked List für Integer implementiert. In dieser Aufgabe arbeiten wir an einer ähnlichen Liste `SpecialIntList`. Der Unterschied ist, dass ein Node von `SpecialIntList` kein Feld mit dem Integer Wert hat, sondern stattdessen ein Feld mit einer `IntBox`, welche dann das Feld mit dem Integer Wert hat. Diese zusätzliche Indirektion erlaubt, dass mehrere Listen die gleichen Boxen verwenden können. Fügen Sie der Klasse `SpecialIntList` eine weitere Methode `SpecialIntList subtract(SpecialIntList list)` hinzu, welche eine `SpecialIntList` als Parameter nimmt und eine `SpecialIntList` zurückgibt. Die zurückgegebene Liste enthält alle Boxen der Liste, auf welcher die Methode aufgerufen wurde, ausser der Wert der Box ist in der Liste, welche als Argument übergeben wird. Das heisst, falls eine Box einen Wert enthält, der auch in der Liste des Funktionsargumentes auftritt, dann ist diese Box nicht in der zurückgegebenen Liste enthalten. Die Boxen, der zurückgegebenen Liste sollen dabei die gleiche Reihenfolge der ursprünglichen Liste beibehalten. Die Liste, auf welcher die Methode aufgerufen wird, und die Liste, welche als Argument übergeben wird, dürfen dabei nicht verändert werden. Beachten Sie, dass die Boxen aus der zurückgegebenen Liste wirklich die gleichen Boxen sind wie aus der Liste, auf welcher die Methode aufgerufen wurde. Das bedeutet, dass Änderungen der einen Liste auch die andere Liste betreffen können, wie wir an folgendem Beispiel illustrieren:

```
SpecialLinkedIntList listA = ... // Liste mit den Werten 3, 7, 5, 5, 2
SpecialLinkedIntList listB = ... // Liste mit den Werten 5, 3, 8

// Resultiert in einer Liste mit den Werten 7, 2 (diese Reihenfolge!)
SpecialLinkedIntList result = listA.subtract(listB);

// Updated den Wert 2 zu dem Wert 8
result.set(1, 8);

// Weil result und listA die gleichen Nodes haben,
// ist auch listA von dem Update des Wertes betroffen.
assert(listA.get(4) == 8);

// Das Gleiche gilt auch umgekehrt.
listA.set(4, 52);
assert(result.get(1) == 52);
```

In der Datei `"SpecialIntListTest.java"` finden Sie den obigen Test. Wie immer, Sie können weitere Methoden und Felder hinzufügen. Ändern Sie aber nicht die anderen bestehenden Methoden und Felder. Bei der Bewertung können wir Tests hinzufügen, um zu prüfen, dass die anderen Methoden sich verhalten wie erwartet.