

Lösungen 2

Patrick Bucher

06.03.2017

1

1.3

a)

- Klasse `SingleLinkedList<Allocation>`
 - Eigenschaften
 - * - `first` : `Element<Allocation>`
 - * - `last` : `Element<Allocation>`
 - Methoden
 - * + `add(element : Element<Allocation>) : void`
 - * + `remove(element : Element<Allocation>) : void`
- Klasse `Element<Allocation>`
 - Eigenschaften
 - * - `next` : `Element<Allocation>`
 - Methoden
 - * + `setNext(next : Element<Allocation>) : void`
 - * + `getNext() : Element<Allocation>`

b)

Dieses Modell erlaubt kein Iterieren über die Elemente. Dazu müsste Zugriff auf die Eigenschaft `first` gewährt werden, was nicht im Sinne von Datenkapselung und Information Hiding ist.

c)

Die Klasse `SingleLinkedList` müsste das Interface `Iterable` implementieren und die Methode `getIterator() : Iterator` anbieten.

d)

Hinweis: es werden nur Bestandteile aufgeführt, die im Modell von `SingleLinkedList` noch nicht vorhanden sind.

- Klasse `DoubleLinkedList<Allocation>`
 - identisch zu `SingleLinkedList`
- Klasse `Element<Allocation>`
 - Eigenschaften
 - * - `previous : Element<Allocation>`
 - Methoden
 - * + `setPrevious(previous : Element<Allocation>) : void`
 - * + `getPrevious() : Element<Allocation>`

e)

Siehe oben; ersetze `Allocation` durch Typparameter `T`.

2

2.3

a)

TODO