

Aufgabenblatt: ADT – Verkettete Listen

Wichtig: Für die Schüler, die bereits mit dem Thema vertraut sind, eignet sich die Arbeit mit generischen Listen, dies ist der zweite Vorlagenordner!

(1.)(a.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode "listAll", die den Inhalt aller Knoten auf der Konsole ausgibt. Verwenden Sie in der Klasse System.out.print!

Im weiteren soll keine Methode eine direkte Ausgabe im Konsolenfenster machen!

(b.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode "size", die die Anzahl aller Knoten auf der Konsole ausgibt.

(c.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode "get":

```
public Object get(int index)
```

Diese Methode gibt ein in der Liste gespeichertes Objekt zurück. Die Position des Objekts wird der Methode als Index übergeben. Die Methode soll im Falle eines ungültigen Index(kleiner 1 oder größer als die Listenanzahl) null ausgeben!

(d.) Erweitern Sie die Klasse Node um eine Methode setContent!

(e.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode isEmpty(), die wiedergibt, ob die Liste leer ist!

(f.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode clear, die alle Einträge der Liste löscht, diese aber(d.h. den Listenkopf) erhält!

(g.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode add(überladen!), die ein Element an einer bestimmten Stelle einfügt(und die anderen Elemente nach hinten verschiebt)!

```
public void add(Object content, int index)
```

(h.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer privaten Variable size, die die Anzahl der Elemente speichert! Diese Variable soll geeignet beim Anlegen und Löschen(Aufgabe siehe unten) erhöht oder erniedrigt werden! Vereinfachen Sie auch die Methode size, die Ihnen die Anzahl der Listenelemente wiedergibt!

(i.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode remove, die ein Element an einer bestimmten Stelle löscht! Achten Sie auch darauf, dass die interne Variable size erniedrigt wird!

(j.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode set, die ein Element an einer bestimmten Stelle ersetzt!

(k.) Modifizieren Sie die vorhandene Klasse "SinglyLinkedList": Die Add-Methoden verwenden ähnlichen Quellcode: Verschanken Sie ihr Programm, indem Sie eine Version von add mit zwei Parametern die andere mit einem Parameter aufrufen lassen!

(l.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode contains, die feststellt, ob ein Element in der Liste vorhanden ist und true oder false ausgibt!

(m.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode indexOf, die den Index von einem Element in der Liste ermittelt und diesen zurückgibt! Ist dieser nicht vorhanden, wird -1 zurückgeben!

Fortgeschrittene Techniken(ohne Lösung):

(n.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode remove, die ein bestimmtes Objekt entfernt, falls es dieses gibt! Warum und wann funktioniert die Musterlösung nicht und wie muss der Quellcode geändert werden?

(o.) Ergänzen Sie die vorhandene Klasse "SinglyLinkedList" mit einer Methode toString, die die komplette Liste als einen String zurückgibt(Diese Lösung ist natürlich nicht als generische Klasse möglich)!

(p.) Kodieren Sie die Methoden addAll, containsAll und subList mit gleicher Funktion wie beim ArrayList-Objekt!

(q.) Leiten Sie eine Sie die neue Klasse "SinglySortedLinkedList" von der "SinglyLinkedList" ab! Welche Methoden müssen neu gestaltetet werden? Wie nennt man das Verfahren?

(r.) Kodieren Sie die Klasse "SinglySortedLinkedList" so, dass diese die Klasse Node enthält(Klasse in Klasse!)

(s.) Korrigieren Sie die Methode reverse, so dass diese korrekt funktioniert!

(2.) Erstellen Sie eine doppelt verkettete Liste! Erstellen Sie analog zu obiger Vorlage Klassen und ein Testprogramm! Ferner können Sie auch Aufgabe (a.)-(m.).... mit einer doppelt verketteten Liste lösen!