

Aufgabe 4.1

Schreiben Sie ein Java-Programm `Palindrom.java`, welches als Kommandozeilenparameter eine Zeichenkette (String) übergeben bekommt und für diese Zeichenkette prüft, ob es sich um ein Palindrom handelt und entsprechend `Ist ein Palindrom` bzw. `Ist kein Palindrom` auf dem Bildschirm ausgibt. Beschreiben Sie kurz die grundlegende Vorgehensweise Ihres Algorithmus zum Testen, ob eine Zeichenkette ein Palindrom ist.

Zur Erläuterung: Bei einem *Palindrom* ist die Folge der Zeichen in der Zeichenkette in der Reihenfolge vom ersten bis zum letzten Zeichen gleich der Folge der Zeichen in der Zeichenkette vom letzten bis zum ersten Zeichen (man erhält also immer die gleiche Folge von Zeichen, egal ob man vorwärts oder rückwärts liest). Beispiele für Palindrome sind `aababaa`, `acbdeffedbc` und `ababababab`. Dagegen sind `abc` und `ababab` keine Palindrome.

(6 Punkte)

Aufgabe 4.2

Erweitern Sie die Klasse `MyLinkedList.java` aus der Vorlesung um eine Methode

```
public void deleteElementsGreaterThan(int x),
```

die eine ganze Zahl `x` als Parameter übergeben bekommt und alle Elemente aus der Liste entfernt, in denen eine Zahl gespeichert ist, die größer ist als `x`. Für die Liste

(3, 5, 2, 10, 7, 8, 20, 17)

und `x` gleich 9 soll zum Beispiel als Ergebnis die Liste

(3, 5, 2, 7, 8)

entstehen.

Was würden Sie anders machen, wenn Sie davon ausgehen könnten, dass die in der verketteten Liste gespeicherten ganzen Zahlen aufsteigend sortiert sind?

(6 Punkte)

Aufgabe 4.3

Erweitern Sie die Klasse `MyStack` aus der Vorlesung um folgende Methode:

```
public int[] MultiPop(int n)
```

Diese Methode bekommt eine natürliche Zahl `n` übergeben und soll diese Anzahl von Elementen aus dem Stack mit `Pop()` entfernen, sofern überhaupt so viele Elemente im Stack enthalten sind. Sind weniger als `n` Elemente im Stack, dann sollen alle Elemente aus dem Stack mit `Pop()` entfernt werden.

Der Rückgabewert der Methode soll ein Array von ganzen Zahlen sein, in dem die aus dem Stack entfernten Elemente in der Reihenfolge, wie diese entfernt wurden, abgelegt sind. Die Länge des Arrays soll gleich der Anzahl der tatsächlich entfernten Elemente aus dem Stack sein.

Sei zum Beispiel folgender Stack gegeben, der gegenwärtig die Elemente

(3, 6, 1, 7, 9, 10, 14)

enthält (wie in der Vorlesung ist das erste Element, also die 3, das Element, was zuletzt in den Stack eingefügt wurde). In dieser Situation soll der Aufruf

MultiPop(4)

ein Array der Länge 4 zurückgeben, in dem die Zahlen 3, 6, 1, 7 in dieser Reihenfolge abgelegt sind, und der Stack enthält danach nur noch die Elemente

(9, 10, 14).

Dagegen soll der Aufruf

MultiPop(10)

ein Array der Länge 7 zurückgeben, in dem die Zahlen 3, 6, 1, 7, 9, 10, 14 in dieser Reihenfolge abgelegt sind, und der Stack ist danach leer. Implementieren Sie Ihren Algorithmus in einer Methode **MultiPop** und speichern Sie die so erweiterte Klasse **MyStack** in einer Datei **MyStack.java**.

(6 Punkte)