

Aufgabe 11.1

Implementieren Sie eine Klasse `Kran.java`. Diese soll direkt aus der Klasse `Fahrzeug.java` abgeleitet werden. Die Klasse `Kran.java` soll eine zusätzliche Eigenschaft `maxTraglast` vom Typ `int` haben. Außerdem soll der Konstruktor von `Kran.java` einen vierten Parameter `t1` vom Typ `int` enthalten. Mit diesem Wert soll `maxTraglast` initialisiert werden. Überschreiben Sie die Methode `printData` so, dass die neue Eigenschaft mit ausgegeben wird.

(4 Punkte)

Aufgabe 11.2

Die Klasse `TreeSet` der Java-Klassenbibliothek bietet eine Methode

```
public Object last()
```

an. Versuchen Sie der Dokumentation zu entnehmen, was diese Methode im Wesentlichen leistet.

(1 Punkte)

Aufgabe 11.3

Gegeben sind k aufsteigend sortierte Folgen $F_1, F_2, \dots, F_k$ von ganzen Zahlen. Für alle $i \in \{1, 2, \dots, k\}$ bezeichnen wir mit n_i die Anzahl der Elemente in der Folge F_i und wir setzen $n = n_1 + n_2 + \dots + n_k$. Die gegebenen k Folgen sollen in eine gemeinsame Folge G gemischt werden, die auch wieder aufsteigend sortiert ist. Dazu soll der in der Vorlesung behandelte Algorithmus so verallgemeinert werden, dass in jedem Schritt aus den noch verbliebenen Elementen in $F_1, F_2, \dots, F_k$ das kleinste Element herausgesucht und nach G kopiert wird. Wie kann man diese Idee so umsetzen, dass die sich ergebende Laufzeit im worst case in $O(n \log_2 k)$ ist?

(4 Punkte)

Aufgabe 11.4

Mit Hilfe welcher in der Vorlesung bereits behandelten Datenstruktur kann man leicht eine nicht rekursive Variante des Sortierverfahrens QuickSort implementieren? Beschreiben Sie kurz, wie die Datenstruktur dabei verwendet wird.

(2 Punkte)
