

Name: \_\_\_\_\_

Mat.Nr.: \_\_\_\_\_

## Aufgabe 1

Schreiben Sie die Klasse *LKWFlotte*, die LKWs verwaltet. Implementieren Sie dazu die Methode

*void add(LKW einLKW),*

die einen LKW zur Flotte hinzufügt. Implementieren Sie weiters die Methode

*boolean reserviere(int anzahlLKWs, Tag datum),*

die für den angegebenen Tag die angegebene Anzahl von LKWs reserviert und dann *true* zurückliefert. Sind für diesen Tag nicht genügend LKWs verfügbar, sollen keine Reservierungen durchgeführt und *false* zurückgeliefert werden.

In der Klasse *LKW* stehen die Methoden

*boolean istVerfuegbar(Tag datum),*

die zurückliefert, ob der LKW an diesem Tag verfuegbar ist, und

*void reserviere(Tag datum),*

die den LKW für diesen Tag reserviert, zur Verfügung.

## Aufgabe 2

Implementieren Sie die Methode

*int findeHaeufigstesElement(int[] a),*

die die Zahl zurückliefert, die im Array *a* am häufigsten vorkommt. (Kommen mehrere Zahlen gleich häufig vor, kann eine beliebige der häufigsten Zahlen zurückgeliefert werden.)

## Aufgabe 3

Von einem Zentrallager aus werden mehrere Filialen durch einen LKW beliefert. Dem LKW stehen mehrere Routen zur Auswahl, wobei auf einer Route jeweils ein Teil der Filialen beliefert wird. In der Klasse *Zentrallager* stehen folgende Methoden zur Verfügung:

*Filiale[]* *getAlleFilialen()*,

die alle Filialen zurückliefert, die beliefert werden müssen,

*Route[]* *getAlleRouten()*,

die alle möglichen Routen zurückliefert, und

*boolean* *istAufRoute(Filiale eineFiliale, Route eineRoute)*,

die zurückliefert, ob die angegebene Filiale durch die angegebenen Route beliefert wird.

Ergänzen Sie die Klasse *Zentrallager* um eine Methode

*Route[]* *getOptimaleRouten()*,

die ein Array von Routen zurückliefert, sodass (1) jede Filiale durch zumindest eine dieser Routen beliefert wird, und (2) die Anzahl der notwendigen Routen minimal ist. Ist eine Belieferung aller Filialen durch die zur Auswahl stehenden Routen nicht möglich, soll *null* zurückgeliefert werden.

*Hinweis:* Eine Möglichkeit ist, eine rekursive Hilfsmethode

*Route[]* *getOptimaleRouten(Route[] routenauswahl)*,

zu verwenden, die eine minimale Anzahl von Routen aus den ausgewählten Routen zurückliefert.