

Name: _____

Mat.Nr.: _____

Aufgabe 1

Gegeben sind die Klassen *Frachtschiff* und *Fracht* mit den angegebenen Methoden:

```
class Frachtschiff
{
    /* Liefert die noch freie Ladeflaeche in qm. */
    double getFreieLadeflaeche() {...}

    /* Liefert die maximal moegliche Beladung in Tonnen. */
    double getMaxBeladung() {...}

    /* Liefert die aktuelle Beladung in Tonnen. */
    double getBeladung() {...}

    /* Fuegt die Fracht der Ladung des Frachtschiffs hinzu. */
    void add(Fracht eineFracht) {...}
}

class Fracht
{
    /* Liefert die von der Fracht benoetigten Flaeche in qm. */
    double getFlaeche() {...}

    /* Liefert das Gewicht der Fracht in Tonnen. */
    double getGewicht() {...}
}
```

Schreiben Sie eine Klasse *Frachtbrief* mit einer Methode

boolean ordneZu(Fracht eineFracht, Frachtschiff einFrachtschiff),

die die angegebene Fracht diesem Frachtschiff zuordnet. Falls dieses Frachtschiff die Fracht nicht mehr transportieren kann, soll die Fracht nicht zugeordnet werden und die Methode *false* zurückliefern. Ansonsten soll die Methode *true* zurückliefern.

Aufgabe 2

Schreiben Sie in der Klasse *Frachtbrief* aus Aufgabe 1 eine Methode

Fracht `getBesteFracht(ArrayList<Fracht> frachtliste, double freieFlaeche)`,

die aus der *frachtliste* jene Fracht zurückliefert, die am besten die *freieFlaeche* füllt, d.h. jene Fracht, deren benötigte Fläche möglichst groß ist, aber die *freieFlaeche* nicht übersteigt. Falls es keine solche Fracht gibt, soll die Methode *null* zurückliefern.

Aufgabe 3

Gegeben ist die Klasse *UniAngehoeriger* und das Interface *Lehrbeauftragter*:

```
class UniAngehoeriger
{
    /* Konstruktor, dem der Name des
       UniAngehoerigen uebergeben wird. */
    UniAngehoeriger(String name) {...}

    /* Liefert die Daten (nicht nur den Namen)
       des UniAngehoerigen in einem String zurueck. */
    public String toString() {...}
}

interface Lehrbeauftragter
{
    /* Gibt eine Liste mit den Nummern der
       Lehrveranstaltungen des Lehrbeauftragten zurueck. */
    List<Integer> getLVAs();
}
```

Schreiben Sie eine Klasse *Professor*, die Unterklasse von *UniAngehoeriger* ist und auch das Interface *Lehrbeauftragter* implementiert. Diese Klasse soll über einen Konstruktor

Professor(*String name*, *ArrayList<Integer> lvaListe*)

verfügen, dem der Name des Professors und die Liste seiner Lehrveranstaltungsnummern übergeben wird. Implementieren Sie die Methode `getLVAs()`.

Überschreiben Sie in *Professor* auch die Methode `toString()` von *UniAngehoeriger* so, dass zusätzlich zu den Daten des Uniangehörigen auch die Nummern der Lehrveranstaltungen des Professors zurückgegeben werden.