

**Name:**

**Matrikelnummer:**

### **Aufgabe 1.**

Schreiben Sie eine Klasse Lagerplanung mit der Methode

```
public int [] planen(int anzBereiche, int anzPlaetze, int maxStuecke,
                    int [] stueckArtikel)
```

die die Belegung eines Lagers berechnet. Das Lager besteht aus `anzBereiche` vielen identischen Bereichen, die jeweils über `anzPlaetze` viele Lagerplätze verfügen.

Die einzulagernden Artikel sind im Array `stueckArtikel` angegeben, wobei `stueckArtikel[i]` angibt, wie viele Stück vom Artikel mit Nummer `i` eingelagert werden sollen. Jedem Artikel soll ein Lagerbereich zugeordnet werden, also eine Nummer `0... anzBereiche-1`. Einem Lagerbereich dürfen nicht mehr Artikel zugeordnet werden als Lagerplätze vorhanden sind. Weiters dürfen in einem Lagerbereich insgesamt (summiert über alle Lagerplätze) nicht mehr als `maxStuecke` viele Stücke gelagert werden.

Die Methode `planen` soll eine zulässige Zuordnung von Artikeln zu Lagerbereichen zurückliefern, sodass `zuordnung[i]` den Lagerbereich angibt, der dem Artikel `i` zugeordnet ist. Gibt es keine zulässige Zuordnung, so soll die Methode `null` zurückliefern.

## Aufgabe 2.

Gegeben seien die Klassen *Lieferant* und *Produkt*. Die Klasse *Lieferant* verfügt über die folgenden Methoden:

Die Methode

```
public void bestellen(Produkt dasProdukt, int menge)
```

bestellt *menge* Stück von *dasProdukt* beim Lieferanten.

Die Methode

```
public int kannLiefern(Produkt dasProdukt)
```

gibt die Menge zurück, die der Lieferant von *dasProdukt* liefern kann.

Die Methode

```
public double getPreis(Produkt dasProdukt)
```

gibt den Preis des Lieferanten für ein Stück von *dasProdukt* zurück.

Schreiben Sie eine statische Methode

```
public static boolean findeBestenLieferanten(  
    ArrayList<Lieferant> dieLieferanten,  
    Produkt dasProdukt, int menge),
```

die *menge* Stück von *dasProdukt* bei jenem Lieferanten aus *dieLieferanten* bestellt, der *menge* Stück von *dasProdukt* zum günstigsten Preis liefern kann. (Genauer: Es soll unter all jenen Lieferanten in *dieLieferanten*, die *menge* Stück von *dasProdukt* liefern können, bei jenem bestellt werden, der das Produkt zum günstigsten Preis anbietet.) Sollte kein Lieferant in *dieLieferanten* die gewünschte Menge liefern können, so soll die Methode *false* zurückgeben, ansonsten *true*.

### Aufgabe 3.

Gegeben seien zwei Klassen *Webseite* und *Suchmaschine*. Die Klasse *Suchmaschine* verfügt über einen Konstruktor

```
public Suchmaschine(ArrayList<Webseite> dieWebseiten)
```

sowie über eine Methode

```
public ArrayList<Webseite> finde(String wort),
```

die eine Liste von allen (durch den Konstruktor bekannten) Webseiten zurückgibt, die *wort* enthalten.

Schreiben Sie eine Klasse *ErweiterteSuchmaschine*, die von *Suchmaschine* erbt und das folgende Interface *ErweiterteSuche* implementiert:

```
interface ErweiterteSuche
{
    /**
     * gibt eine Liste von allen Webseiten zurueck,
     * die posWort enthalten und negWort nicht enthalten
     */
    public ArrayList<Webseite> findeOhne(
        String posWort, String negWort)
}
```

Die Klasse *ErweiterteSuchmaschine* soll auch über den Konstruktor

```
public ErweiterteSuchmaschine(ArrayList<Webseite> dieWebseiten)
```

verfügen.