

**Name:**

**Matrikelnummer:**

**Bearbeitungszeit:** 90 min.

**Aufgabe 1.** Schreiben Sie eine Klasse `Routensuche` mit einem parameterlosen Konstruktor und der Methode

```
public boolean existsRoute(ArrayList<Integer>[] nachbarn).
```

Dabei ist `nachbarn` ein Array von `ArrayList`s, sodass `nachbarn[i]` die Liste der Nachbarn des Orts `i` angibt (Orte sind jeweils mit einer Nummer aus  $0, \dots, \text{nachbarn.length}-1$  bezeichnet).

Die Methode `existsRoute()` soll überprüfen, ob es eine Route gibt, die alle Orte  $0, \dots, \text{nachbarn.length}-1$  genau einmal besucht. Die Route kann in einem beliebigen Ort starten und in einem beliebigen Ort enden. (Eine Route ist eine Folge von Orten  $o_1, \dots, o_n$  wobei jeweils  $o_{i+1}$  Nachbar von  $o_i$  ist.)

**Aufgabe 2.** Gegeben ist die Klasse `Produkt` mit den Methoden

```
public boolean zweifach(),  
public int gibFertigungszeit().
```

Die Methode `zweifach()` gibt `true` zurück, wenn das Produkt aus zwei Teilprodukten besteht, und ansonsten `false`. Die Methode `gibFertigungszeit()` liefert die Zeit, die zur Fertigung des Produkts nötig ist.

Schreiben Sie eine Klasse `Planung` mit einem parameterlosen Konstruktor und der Methode

```
List<Produkt> waehle(List<Produkt> dieProdukte, int zeit),
```

die Produkte aus `dieProdukte` auswählt und in einer `List` zurückgibt, die in Summe innerhalb von `zeit` Zeiteinheiten gefertigt werden können. Dabei sollen die Produkte so gewählt werden, dass insgesamt möglichst viele (Teil-)Produkte gefertigt werden: die Fertigung von "zweifachen" Produkten zählt bei der Bewertung einer Auswahl doppelt.

*Beispiel:* Bei zweifachen Produkten mit Fertigungszeiten 13,20,17,12 und vier weiteren Produkten mit Fertigungszeiten 11,5,4,8 enthält die korrekte Lösung für `zeit=52` die Aufträge mit den Zeiten 13,17,12,5,4 (in beliebiger Reihenfolge).

*Hinweis:* Dieses Beispiel kann ohne Rekursion gelöst werden. Es ist (relativ) einfach, eine Auswahl zu erzeugen, die höchstens ein (Teil-)Produkt weniger als die optimale Lösung enthält. Für eine solche Lösung erhalten Sie 8 von 10 Punkten.