

Name:

Matrikelnummer:

Bearbeitungszeit: 90 min.

Aufgabe 1. An einem runden Tisch sollen mehrere Personen Platz nehmen. Dabei sollen die Personen so gesetzt werden, dass sie mit ihren rechten und linken Sitznachbarn möglichst zufrieden sind: Die Summe der Zufriedenheiten aller Personen mit ihrem linken und rechten Nachbarn soll maximiert werden.

Gegeben ist die Klasse `Person` mit der Methode

```
int getZufriedenheit(Person anderePerson),
```

die den Zufriedenheitswert einer Person mit einer anderen Person als Sitznachbarn zurückgibt.

Schreiben Sie die Klasse `Sitzordnung` mit einem parameterlosen Konstruktor und der Methode

```
Person[] getSitzordnung(List<Person> personen),
```

die eine optimale Sitzordnung für die übergebenen Personen zurückgibt. Das zurückgegebene Array soll die Sitzordnung am runden Tisch im Uhrzeigersinn enthalten. Die erste Person der Sitzordnung kann beliebig gewählt werden.

Aufgabe 2.

Für einen sportlichen Bewerb sollen die Teilnehmer in Paare mit möglichst ähnlichem Gewicht eingeteilt werden: Für alle Paare werden die Gewichtsunterschiede im Paar aufsummiert, und diese Summe soll minimal sein.

Gegeben ist die Klasse `Teilnehmer` mit der Methode

```
double getGewicht(),
```

die das Gewicht des Teilnehmers zurückgibt.

Schreiben Sie die Klasse `Paarung` mit einem parameterlosen Konstruktor und der Methode

```
Teilnehmer[][] getPaare(Teilnehmer[] dieTeilnehmer),
```

die die optimalen Paare für die Teilnehmer zurückgibt (Sie können annehmen, dass die Anzahl der Teilnehmer gerade ist). Das zurückgegebene doppelt indizierte Array `paare` soll die Paare so enthalten, dass `paare[i][0]` und `paare[i][1]` die Teilnehmer im Paar `i` bezeichnet.

Beispiel: Für

```
paare = getPaare({tn1, tn2, tn3, tn4, tn5, tn6});
```

könnte `paare` die Werte `{{tn1, tn4}, {tn2, tn3}, {tn6, tn5}}` enthalten.