

Name:

Matrikelnummer:

Bearbeitungszeit: 60 min.

Gegeben sind die Klassen `fahrten.geg.Ort` und `fahrten.geg.LKW`. Die Klasse `LKW` verfügt über den Konstruktor

```
public LKW(Ort einOrt),
```

der einen `LKW` in `Ort einOrt` erzeugt, und über die Methoden

```
public double fahreZu(Ort andererOrt),
public Ort getStandort().
```

Die Methode `fahreZu(andererOrt)` lässt den `LKW` vom derzeitigen Standort zum `Ort andererOrt` fahren. Der neue Standort des `LKW` ist dann `andererOrt`. Der Rückgabewert der Methode ist die für diese Fahrt benötigte Fahrzeit. Die Methode `getStandort()` gibt den derzeitigen Standort des `LKW` zurück.

Schreiben Sie die Unterklasse `fahrten.LKWmitAuswertung` von `LKW` mit der zusätzlichen Methode

```
public double getMittlereFahrzeit(Ort start, Ort ziel),
```

die die mittlere Fahrzeit aller Fahrten des `LKW` von `Ort start` zu `Ort ziel` zurückgibt (die Fahrzeit kann zufällig schwanken). Wenn `start==ziel` ist, dann soll die Methode den Wert `0.0` zurückgeben. Hat der `LKW` noch keine Fahrten von `start` zu `ziel` durchgeführt (`start != ziel`), dann soll die Methode `Double.POSITIVE_INFINITY` zurückgeben.

Der Konstruktor von `LKWmitAuswertung` soll dem Konstruktor der Klasse `LKW` entsprechen. Das Verhalten der Methoden `fahreZu(andererOrt)` und `getStandort()` in der Klasse `LKWmitAuswertung` sollen dem Verhalten der Methoden in der Klasse `LKW` entsprechen. Das Paket `fahrten.geg` darf nicht verändert werden.

Hinweis: In der Klasse `LKWmitAuswertung` müssen die Zeiten der durchgeführten Fahrten in geeigneter Weise gespeichert werden.

Beispiel: Es wird ein `LKW` mit dem Aufruf `new LKWmitAuswertung(ort0)` erzeugt. In der untenstehenden Tabelle sind die darauffolgenden Aufrufe von `fahreZu(andererOrt)` angegeben.

<i>andererOrt</i>	Fahrzeit
ort1	10.5
ort2	8.0
ort0	16.0
ort1	9.5
ort0	12.0
ort1	13.0

Die Aufrufe der Methode `getMittlereFahrzeit(start,ziel)` müssen dann folgende Werte ergeben:

<i>start\ziel</i>	ort0	ort1	ort2
ort0	0.0	11.0	$+\infty$
ort1	12.0	0.0	8.0
ort2	16.0	$+\infty$	0.0

Testen: Zum Testen Ihrer Lösung stellt die Klasse `Ort` die Methode

```
public static Ort [] getBeispiel()
```

zur Verfügung, die die drei obigen Orte als Array zurückgibt. Weiters kann mit der Methode

```
public static Ort [] getZufaelligeOrte(int anzahl)
```

ein Array mit `anzahl` zufälligen Orten erzeugt werden.