

Name:

Matrikelnummer:

Bearbeitungszeit: 60 min.

Gegeben ist die Klasse `post.gcg.Ort`, die Orte mit den Entfernungen zu anderen Orten darstellt. Dazu enthält die Klasse die Methode

```
public double getEntfernungZu(Ort andererOrt),
```

die die Entfernung zu einem anderen Ort zurückgibt. Zum Beispiel liefert für zwei Orte `ort1` und `ort2` der Aufruf `ort1.getEntfernungZu(ort2)` die Entfernung von `ort1` zu `ort2` zurück. Sie können weiters annehmen, dass die Entfernungen symmetrisch sind, also

```
ort1.getEntfernungZu(ort2) == ort2.getEntfernungZu(ort1)
```

ist.

Schreiben Sie die Klasse `post.Filialauswahl` mit einem parameterlosen Konstruktor und der Methode

```
public Set<Ort> waehleAus(List<Ort> alleOrte, double radius),
```

die aus der Liste `alleOrte` einige Orte auswählt und als Set zurückgibt, sodass folgende zwei Bedingungen erfüllt sind:

1. Für jeden Ort `ort1` in `alleOrte` gibt es eine ausgewählten Ort `ort2`, dessen Entfernung von `ort1` höchstens `radius` ist.
2. Die Entfernung zwischen zwei ausgewählten Orten ist größer als `radius`.

Hinweis: Der Set der ausgewählten Ort kann evtl. auch nur einen Ort enthalten, und er ist leer, wenn `alleOrte` leer ist. Im allgemeinen gibt es mehrere Möglichkeiten, Orte so auszuwählen, dass die Bedingungen 1 und 2 erfüllt sind.

Beispiel: Für die Orte A, . . . ,E sind die Entfernungen in der folgenden Tabelle angegeben:

	A	B	C	D	E
A	0	4	3	6	4
B	4	0	4	6	7
C	3	4	0	3	5
D	6	6	3	0	6
E	4	7	5	6	0

Für `radius==5` sind die korrekten Auswahlen `{C}`, `{A,D}` und `{B,D,E}`.

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

```
public static List<Ort> getBeispiel()
```

zur Verfügung, die eine Liste mit Orten zurückgibt, die dem obigen Beispiel entspricht. Weiters kann mit der Methode

```
public static List<Ort> getAlleOrte(double[][] entfernung)
```

eine Liste mit Orten erzeugt werden, die den angegebenen Entfernungen entsprechen. Die Methode

```
public String toString()
```

der Klasse Ort gibt den Namen des Ortes zurück.