

Name:

Matrikelnummer:

Bearbeitungszeit: 90 min.

Aufgabe 1. Gegeben ist die Klasse `Robot` mit dem Konstruktor

```
Robot(String bezeichnung),
```

der einen Roboter mit der angegebenen Bezeichnung anlegt, und mit der Methode

```
void move(),
```

die den Roboter einen Schritt machen läßt.

Schreiben Sie eine Klasse `Roboterverwaltung` mit einem parameterlosen Konstruktor und den Methoden

```
Robot createRobot(String bezeichnung),  
int getMoves(String bezeichnung).
```

Der Aufruf `createRobot(bezeichnung)` soll einen `Robot` mit der angegebenen Bezeichnung zurückgeben, wenn in der `Roboterverwaltung` noch kein Roboter mit der gleichen Bezeichnung existiert. Ansonsten soll die Methode `null` zurückgeben.

Der Aufruf `getMoves(bezeichnung)` soll die Anzahl der bisherigen Schritte des Roboters mit der angegebenen Bezeichnung zurückgeben. Existiert ein solcher Roboter nicht, dann soll der Wert `-1` zurückgegeben werden.

Hinweis: Für die Realisierung der `Roboterverwaltung` empfiehlt sich die Definition einer Unterklasse von `Robot`, um das Zählen der Schritte zu ermöglichen. Die `Roboterverwaltung` muss die erzeugten Roboter in geeigneter Weise speichern.

Aufgabe 2.

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

```
boolean isClusterable(int[] a, int k, int maxSum),
```

die überprüft, ob das Array `a` so in `k` **zusammenhängende** Teile zerlegt werden kann, dass die Summe der Elemente in jedem Teil höchstens `maxSum` ist. (Ein Teil kann auch leer sein, wenn `k > a.length`.) Sie können annehmen, dass alle Werte in `a` positiv sind.

Zum Beispiel ergibt `isClusterable({11,16,5,5,12,10},3,26)` den Wert `true`, da die Aufteilung `{11},{16,5,5},{12,10}` möglich ist. Ebenso ergibt `isClusterable({11,16,5,5,12,10},4,21)` den Wert `true`, da die Aufteilung `{11},{16,5},{5,12},{10}` möglich ist. Hingegen ergeben `isClusterable({11,16,5,5,12,10},3,25)` und `isClusterable({11,16,5,5,12,10},4,20)` den Wert `false`, da eine entsprechende Aufteilung nicht möglich ist.