

Name:

Matrikelnummer:

Aufgabe 1. Gegeben sind die Klassen Rettungsboot und Schiff. Die Klasse Rettungsboot verfügt über die Methode

```
1 public void reparieren(),
```

sowie die Methode

```
public boolean istOK(),
```

die true zurückliefert, wenn das Rettungsboot funktionstüchtig ist, und die ansonsten false zurückliefert.

Die Klasse Schiff verfügt über einen Konstruktor

```
public Schiff(String bezeichnung),
```

der ein Schiff mit entsprechender Bezeichnung erzeugt. Weiters ist in Schiff die Methode String toString() überschrieben.

Schreiben Sie eine Klasse Passagierschiff, die von Schiff erbt, und über einen Konstruktor

```
public Passagierschiff(Rettungsboot[] dieBoote, String bez)
```

verfügt, der ein Passagierschiff mit den übergebenen Rettungsbooten und der Bezeichnung bez erzeugt.

Überschreiben Sie die Methode String toString() so, dass dem Rückgabewert der Methode aus der Oberklasse noch ein String der Form "Es sind n Rettungsboote funktionstüchtig." angehängt wird, wobei n die Anzahl der funktionstüchtigen Rettungsboote angibt.

Weiters soll die Klasse Passagierschiff über die Methode

```
public boolean bootePruefen(int minAnzahl)
```

verfügen. Für den Fall, dass weniger als minAnzahl Rettungsboote funktionstüchtig sind und das Passagierschiff über mindestens minAnzahl Rettungsboote verfügt, soll diese Methode so viele Rettungsboote reparieren (mittels der Methode reparieren()), dass insgesamt genau minAnzahl Rettungsboote funktionstüchtig sind, und dann true zurückliefern. Wenn minAnzahl größer als die Anzahl der vorhandenen Rettungsboote ist, soll kein Rettungsboot repariert und false zurückgeliefert werden.

Aufgabe 2. Schreiben Sie eine Methode

```
static boolean istPackbar(int [] gewicht, int [] maxLast),
```

die überprüft, ob Gegenstände mit den in `gewicht` angegebenen Gewichten in vorgegebene Kisten gepackt werden können, wobei das von den Kisten jeweils aufnehmbare Gesamtgewicht in `maxLast` angegeben ist.

Beispiele:

```
istPackbar({5,2,7},{4,12}) = true
```

```
istPackbar({5,2,7},{5,9}) = true
```

```
istPackbar({5,2,7},{6,8}) = false
```

Aufgabe 3. Schreiben Sie eine Methode

```
static int getMaxAufsteigend(int [] liste),
```

die die Länge der längsten (streng) aufsteigenden Folge von direkt aufeinander folgenden Werten in `liste` zurückgibt.

Beispiele:

```
getMaxAufsteigend({1,2,1,2,1,2,1}) = 2
```

```
getMaxAufsteigend({1,5,3,4,8,9,4,3,5}) = 4
```

```
getMaxAufsteigend({9,8,7,6,5,4,3,2,1}) = 1
```

```
getMaxAufsteigend({0,0,1,1,2,2}) = 2
```