



Prof. Dr. Fritz Jobst

Aufgaben zum Fach Programmieren in Java

Blatt 7 : Objektorientierung

1. Geben Sie eine Klasse PersonMitOrt an.

Die Klasse PersonMitOrt soll als Attribute einen Namen, einen Vornamen sowie einen Ort (jeweils als String) erhalten. Geben Sie einen Konstruktor für diese Klasse sowie eine toString()-Methode an. Der Konstruktor soll alle Attribute auf die ihm übergebenen Werte setzen. Im Beispiel soll das Objekt x den Namen "Meier", den Vornamen "Sepp" und den Ort "Mannheim" haben. Die toString()-Methode soll ihr Ergebnis mit dem + Operator oder mit einem StringBuffer-Objekt zusammenbauen.

```
PersonMitOrt x = new PersonMitOrt ("Meier", "Sepp", "Mannheim");
```

2. Verwaltung der Objekte

Die aus einer Aufgabe bekannten 5 Personen mit Ort sollen als Objekte der Klasse PersonMitOrt angelegt und in einem Feld aus Personen verwaltet werden. Hier die Personen:

```
Meier Sepp Mannheim  
Huber Anton Regensburg  
Berger Franz Passau  
Winkler Hans Regensburg  
Maler Horst Passau
```

Das Feld soll ausgegeben, mit dem Bubble-Sort sortiert und nach dem Sortieren nochmals ausgegeben werden.

Eine Möglichkeit: Sie können in der Klasse PersonMitOrt folgende Methode zum Vergleich definieren:

```
public int compareTo (PersonMitOrt other) { ...}
```

Der Aufruf x.compareTo(y) liefert für x, y als PersonMitOrt-Objekte:

```
-1 : falls x kleiner als y ist  
0 : falls x gleich y ist  
+1 : falls x größer y ist
```

Verwenden Sie zum Vergleich von Strings die compareTo-Methode der Klasse String. Vergleichen Sie zunächst die Namen. Bei Namensgleichheit entscheidet der Vorname über die Relation. Ist auch dieser gleich, so entscheidet der Ort. Wenn auch dieser gleich ist, dann sind die Objekte gleich. Damit können Sie zwei Objekte vergleichen und den Bubble-Sort anwenden.

3. Aufgabe: Stack

Geben Sie eine Klasse Stack an. Stack soll eine "Last In, First Out"-Strategie zur Verwaltung von n=20 Einträgen implementieren. Stack soll die Operationen `void push(int)`, `int pop()` und `int peek()` anbieten. `push(Zahl)` soll eine Zahl auf den Stack legen. `pop()` soll das zuletzt auf den Stack gelegte Element zurückliefern, und dieses vom Stack entfernen. `peek()` soll das zuletzt auf den Stack gelegte Element liefern, ohne den Stack zu verändern. Siehe auch: <http://de.wikipedia.org/wiki/Stack>

Achten Sie darauf, dass ein Zugriff auf einen leeren Stack nicht möglich ist. Auch ein `push()` auf einen vollen Stack ist nicht erlaubt.