

Einstieg in Java und OOP

Christian Silberbauer

Übungsblatt 7

Die folgenden Aufgaben beziehen sich auf die Lohn- und Gehaltsabrechnung aus dem Übungsblatt 3.

Kopieren Sie bitte zunächst das vorhandene Projekt und speichern Sie es unter *PersonalVerwaltung2*, bevor Sie Änderungen anstellen.

Aufgabe 1

Ändern Sie die `add()`-Methode der Klasse `MitarbeiterListe` aus dem Übungsblatt 3 Aufgabe 2 so ab, dass der interne Array in einen doppelt so großen Array umkopiert wird, sobald er durch Hinzufügen eines Mitarbeiters droht *überzulaufen*.

Aufgabe 2

Verallgemeinern Sie die Klasse `MitarbeiterListe` aus dem Übungsblatt 3 Aufgabe 2 zu einer generischen Klasse `MyArrayList<T>`. Lösen Sie mit der neuen Klasse auch die Klasse `AbrechnungsListe` aus dem Übungsblatt 3 Aufgabe 6 ab. Sie können die Klassen `MitarbeiterListe` und `AbrechnungsListe` anschließend aus dem Projekt löschen.

Aufgabe 3

Sorgen Sie dafür, dass die Elemente von `MyArrayList<T>` mit der erweiterten for-Schleife durchlaufen werden können. Dazu muss diese das `Iterable<T>`-Interface implementieren. Die Iterator-Klasse sollte als private, interne Klasse namens `MyArrayListIterator` implementiert werden. Ihre `next()`-Methode wirft eine `NoSuchElementException`, wenn sie am Ende der Liste angelangt ist. Ein Entfernen von Elementen muss der Iterator nicht unterstützen, daher wirft die `remove()`-Methode eine `UnsupportedOperationException`. Wenden Sie die erweiterte for-Schleife in `listMitarbeiter()` und `listAbrechnungen()` in der Klasse `PersonalVerwaltung` an.

Kopieren Sie bitte nochmals das vorhandene Projekt und speichern Sie es unter *PersonalVerwaltung3*, bevor Sie weitere Änderungen anstellen.

Aufgabe 4

Verwenden Sie nun statt der eigenen `MyArrayList<T>` die Standardklasse `ArrayList<T>`. `MyArrayList<T>` können Sie daraufhin aus dem Projekt entfernen.

Aufgabe 5

Java bietet zum Sortieren die statische Methode `Collections.sort()`. Verwenden Sie diese zum Sortieren der Mitarbeiterliste, sodass Sie auf Ihre eigene Bubblesort-Implementierung verzichten können. Damit dies funktioniert, muss die Klasse `Mitarbeiter` das generische Interface `Comparable<Mitarbeiter>` implementieren. Es ist demnach eine Methode `int compareTo(Mitarbeiter m)` erforderlich, deren Rückgabewert sich im Prinzip verhält wie die `compareTo()`-Methode der Klasse `String`. Arbeiten Sie ggf. mit Hilfe der Java-Dokumentation. Im Anschluss können Sie die `istKleiner()`-Methode löschen, da sie quasi durch die `compareTo()`-Methode ersetzt wird.

Aufgabe 6

Welches Design-Patttern wird in Aufgabe 5 angewandt? Begründen Sie.