

## Übung Java Generics & Collections

Blatt 1 – Ausgabe am 16.05.2017, Bearbeitung bis 30.05.2017

**Hinweise:** Verwenden Sie für die Lösung der Aufgaben das *Eclipse-Projekt* `pi.informatik.pep.jgc`, welches im Repository oder auf der Lehre-Seite verfügbar ist. Eine aktuelle Eclipse Version erhalten Sie unter <http://www.eclipse.org/downloads/>.

Die Abgabe erfolgt über Ihr Repository. Nutzen Sie das Repository auch dazu, Zwischenstände Ihrer Arbeit zu sichern.

### Aufgabe 1.1 *Allgemeinwissen*

Machen Sie sich mit Java Generics vertraut, indem Sie die Aufgaben und Fragen unter <https://docs.oracle.com/javase/tutorial/java/generics/QandE/generics-questions.html> bearbeiten. Ihre Lösungen können Sie anschließend auch anhand der dortigen Musterlösungen vergleichen.

### Aufgabe 1.2 *Grundlagen*

Das Projekt beinhaltet einige Klassen rund um das Unversitätsleben. Um Studenten zu verwalten, werden je nach Bedarf Mengen oder Listen verwendet. Um ein sinnvolles Ergebnis zu erhalten, müssen sowohl Gleichheit zwischen Studenten als auch deren Ordnung definiert werden. Gehen Sie folgendermaßen vor:

- a) Implementieren sie die `hashCode` und `equals` Methoden von Studenten. Überlegen Sie sich genau, wie Sie die Gleichheit definieren und erläutern Sie in einem Kommentar, warum Sie diese so definieren.
- b) Implementieren Sie **entweder** eine `Comparator` Klasse für Studenten, **oder** erweitern Sie die Studentenkategorie um die `Comparable` Schnittstelle. Erläutern Sie in einem Kommentar, warum Sie sich für diese Lösung entschieden haben und welche Unterschiede zwischen den beiden Ansätzen besteht.

Sie können Ihre Lösungen testen, indem sie die `main` Methode in der Klasse `Main` anpassen und ausführen.

### Aufgabe 1.3 *Verwendung von Generics*

Java bringt selbst keine Implementierung einer Baumstruktur mit, da spezielle Anforderungen an diese unterschiedliche Lösungen erfordern. Meist wird ein Baum verwendet, der zusätzliche Informationen

(Daten) an den jeweiligen Knoten ablegen kann. Implementieren Sie dazu die im Projekt vorhandene Klasse `GenericTreeNode`, welche mit verschiedenen Typen verwendet werden kann. Die Funktionsbeschreibung soll als Anhaltspunkt dienen:

- Ein `GenericTreeNode` ist ein Knoten, der selbst als Baum fungieren kann.
- Jeder Knoten kann ein Datenobjekt beliebigen Typs speichern
- Der gesamte Baum speichert jedoch nur dieselben Typen von Datenobjekten, so wie in der Praxis üblich.
- Knoten verwalten ihre (direkten) Kinder sowie Ihren Elternknoten
- Knoten sind geordnet, d.h. die Reihenfolge der Kinder spielt eine Rolle