

Einführung in die Programmierung
WS 2009/10

Übungsblatt 12: Typparameter, Datenstrukturen

Besprechung: 08./10./11./12.02.2010

Hinweise:

Die Abgabe der Aufgaben auf diesem Blatt ist freiwillig. Die behandelten Inhalte sind Stoff der Nachholklausur. Um Ihre individuelle Lösung einschätzen zu können, bieten wir Ihnen auch weiterhin die Möglichkeit der Korrektur an. Weitere Bonuspunkte können nicht erzielt werden.

Aufgabe 12-1 *Typparameter*

Die Klasse `MinMax` enthält zwei Methoden, die das Maximum bzw. das Minimum aus einer Menge von `Comparable`-Objekten bestimmen. Die Schreibweise `Comparable... cs` in der Parameter-Liste steht für ein oder mehrere Objekte vom Typ `Comparable`. Man kann also auf diese Weise eine Methode definieren, die eine beliebige, nicht-leere Menge von Objekten eines bestimmten Typs als Parameter erhält. Diese Schreibweise kann allerdings nur beim letzten Parameter einer Methode auftreten. Im Methodenrumpf kann der entsprechende Parameter behandelt werden wie ein Array, er kann also auch in Schleifen verwendet werden. Hier wird eine erweiterte Form der `for`-Schleife verwendet, die nützlich ist, wenn ein Array lediglich durchlaufen werden soll, ohne dass eine Verwendung des Schleifenindex nötig ist.

Die `main`-Methode enthält nach zwei richtigen Verwendungen der Methoden (für `Integer`- und `String`-Objekte) auch eine Verwendung, in der beide Objekt-Typen gemischt vorkommen. Dieser Fehler wird aber nicht beim Übersetzen erkannt. Korrigieren Sie die Klasse `MinMax`, so dass eine solche gemischte Verwendung bereits beim Übersetzen als Fehler erkannt wird. Damit das korrigierte Programm übersetzbar ist, müssen Sie dann natürlich die letzte Anweisung in der `main`-Methode auskommentieren. Begründen Sie Ihre Korrektur der Klasse.

Aufgabe 12-2 *Datenstrukturen*

Auf der Vorlesungs-Homepage ist die Klasse `Queue` gegeben, die eine Warteschlange mit den Mitteln einer doppelt verankerten Liste implementiert. Ergänzen Sie diese Klasse um folgende Methoden, wobei Sie die Eigenschaft der `Queue` als Datenstruktur nach dem FIFO (First-in-first-out)-Prinzip beachten müssen:

- Die Methode `public void put(E element)` hängt das gegebene Element `element` an das Ende der Liste an.
- Die Methode `public E get()` gibt das erste Element (d.h. das Element, das momentan von allen Elementen am längsten in der Liste gespeichert wird) der Liste aus und löscht dieses Element aus der Liste.
- Die Methode `public boolean contains(E element)` gibt `true` zurück, wenn das gegebene Element `element` in der Liste vorhanden ist, ansonsten `false`. Die Elemente sollen dabei auf Gleichheit getestet werden.