Hochschule
Bonn-Rhein-Sieg

Fachbereich Informatik
Prof. Dr. Peter Becker

Datenstrukturen und Algorithmen
Sommersemester 2012
19. Mai 2013

Aufgabenblatt 7

Aufgabe 1 (Typeinschränkung)

Implementieren Sie eine generische Methode `max2()`, die das zweitgrößte Element einer Liste (Datentyp `List<T>` ermittelt und als Ergebnis liefert.

- Um das zweitgrößte Element einer Liste bestimmen zu können, müssen Sie die in der Liste enthaltenen Objekte vergleichen können. Damit dies gewährleistet ist, müssen Sie eine entsprechende Typeinschränkung definieren (vgl. Folie 159 f.).
- Achten Sie darauf, dass die übergebene Liste mindestens zwei Elemente enthält. Sehen Sie für den Fall, dass die Liste weniger als zwei Elemente enthält, geeignete Maßnahmen vor. Sie könnten z.B. `null` als Ergebnis liefern oder eine `IllegalArgumentException` auslösen.
- Verwenden Sie für die Iteration über die Liste einen Iterator (siehe Folien 142 f.) oder die foreach-Schleife (siehe Folie 144 ff.).
- Schreiben Sie ein kleines Testprogramm für Ihre Methode `max2()`.

Aufgabe 2 (Comparable für generische Paarklasse implementieren)

Erweitern Sie Ihre Implementierung der generischen Klasse für Paare von Aufgabenblatt 5, Aufgabe 3 so, dass

- als Komponententyp nur Typen zugelassen sind, die die Schnittstelle `Comparable` implementieren (Achtung: Typeinschränkung notwendig) und
- die Klasse für Paare diese Schnittstelle selbst implementiert.

Der Vergleich von zwei Paaren Paar(a, b) und Paar(c, d) sei wie folgt definiert:

- Paar(a, b) = Paar(c, d), wenn $a = c$ und $b = d$ gilt,
- Paar(a, b) < Paar(c, d), wenn $a < c$ oder $a = c \wedge b < d$ gilt und
- Paar(a, b) > Paar(c, d), wenn $a > c$ oder $a = c \wedge b > d$ gilt.

Schreiben Sie ein kleines Testprogramm: Erzeugen Sie eine Liste von Paaren und ermitteln Sie das zweitgrößte Paar mit Hilfe der Methode `max2()` von Aufgabe 1.

Aufgabe 3 (Wildcards, Kovarianz und Kontravarianz)

Schreiben Sie eine generische Methode `bildePaarListe()`, die aus zwei gleichlangen Listen $A = (a_1, \ldots, a_n)$ und $B = (b_1, \ldots, b_n)$ Paare $Paar(a_1, b_1), \ldots, Paar(a_n, b_n)$ bildet und diese Paare in eine Liste C einfügt. Ihre Methode hat also drei Parameter (für die Listen A, B und C).

Die Basistypen der Listen A und B müssen dabei nicht identisch sein. Auch muss der Basistyp der Paare von C weder mit dem Basistyp von A noch mit dem Basistyp von B übereinstimmen. Beispiel:

```
class T { }
class T1 extends T { }
class T2 extends T { }
...
List<T1> a = new ArrayList<T1>();
List<T2> b = new ArrayList<T2>();
List<Paar<T>>  c = new ArrayList<Paar<T>>();
...
bildePaarListe(a,b,c);
```

Sie müssen durch Typeinschränkungen in Verbindung mit Wildcards dafür sorgen, dass solche Aufrufe wie oben dargestellt einerseits möglich sind, andererseits aber auch nur sinnvolle Kombinationen von Basistypen zugelassen sind.

Hilfe zur Lösung der Aufgaben erhalten Sie in den Übungen der KW 21.

Keine Abgabe der Lösungen, keine Korrektur, keine Punkte!