

Übungen zu Einführung in die Informatik: Programmierung und Software-Entwicklung

Aufgabe 9-1

Erzeugen von Arrays

Präsenz

Gegeben ist folgendes Programm, welches nach verschiedenen Prinzipien Arrays erzeugt. Geben Sie für dieses Programm den Zustand des Speichers (Stack und Heap) nach den Zeilen 3, 5, 7, 9 und 15 an in grafischer Repräsentation.

```
1 public class CreateArrays {
2     public static void main(String[] args) {
3         int[] a = new int[3];
4         for (int i = 0; i < a.length; i++)
5             a[i] = 2 * i;
6
7         int[] b = { 1, 3 };
8
9         int[] c = new int[a.length + b.length];
10        for (int i = 0; i < c.length; i++) {
11            if (i < a.length)
12                c[i] = a[i];
13            else
14                c[i] = b[i - a.length];
15        }
16    }
17 }
```

Aufgabe 9-2

Operationen auf Arrays

Präsenz

In dieser Aufgabe sollen Sie ein Programm mit einer grafischen Benutzeroberfläche implementieren, welches verschiedene Operationen auf Arrays zur Verfügung stellt.

a) Die grafische Benutzeroberfläche soll wie folgt aussehen:



Es soll zwei Buttons, einen zum Revertieren eines Arrays und einen zum lexikographischen Vergleich zweier Arrays, geben. Darunter soll ein Ausgabebereich platziert werden, in dem Rückmeldung über die Ergebnisse gegeben wird.

Schreiben Sie eine Klasse `ArrayFrame`, die die Hauptklasse dieser grafischen Benutzeroberfläche sein soll und das Fenster erzeugt. Um Ihr Programm ausführen zu können, schreiben Sie eine weitere Klasse `ArrayFrameMain`, die Sie wie gewohnt im gleichen Ordner wie Ihre Klasse `ArrayFrame` abspeichern.

- b) Erweitern Sie Ihre Klasse `ArrayFrame` um eine Ereignisbehandlung für den Button zum Revertieren eines Arrays. Wird der Button zum Revertieren eines Arrays gedrückt, soll der Benutzer zunächst mit Hilfe der Klasse `JOptionPane` nach dem zu revertierenden `int`-Array gefragt werden und anschließend dieses `int`-Array revertiert werden, d.h. die Reihenfolge der Einträge umgedreht werden. Nach dem Revertieren soll der Benutzer im Ausgabebereich darüber informiert werden, was seine Eingabe war und was die Ausgabe (d.h. das revertierte Array) ist.

Hinweis: Auf der Vorlesungswebseite finden Sie eine Klasse `Konverter.java`, mit der Sie in der Methode `konvertiereZuIntArray` einen Komma-separierten String in ein `int`-Array konvertieren können (d.h. die Eingabe `1,2,3` wird konvertiert in ein Array `[1,2,3]`). Umgekehrt konvertiert die Methode `konvertiereZuString` ein `int`-Array in einen Komma-separierten String.

- c) Erweitern Sie Ihre Klasse `ArrayFrame` um eine Ereignisbehandlung für den Button zum lexikografischen Vergleich zweier Arrays, die nach Knopfdruck einzulesen sind. Ein Array $a = [a_1, \dots, a_m]$ ist genau dann kleiner als ein Array $b = [b_1, \dots, b_n]$ falls gilt:

- Entweder $m < n$ und für alle $i = 1, \dots, m$ gilt $a_i = b_i$,
- oder es gibt ein $k \in \mathbb{N}$ mit $k \leq m$ und $k \leq n$, so dass $a_k < b_k$ gilt und für alle $i \in \mathbb{N}$ mit $i < k$ gilt $a_i = b_i$ (also z. B. $[1, 2, 3] < [1, 3, 1]$, hier ist $k = 2$).

Beispielsweise ist das Array `[1, 2, 8, 7, 3]` kleiner als `[1, 2, 8, 7, 3, 1]`, kleiner als `[1, 2, 8, 7, 4]` sowie kleiner als `[1, 2, 9]`. Nach dem lexikografischen Vergleich soll der Benutzer im Ausgabebereich darüber informiert werden, was seine Eingabe war und ob das erste Array lexikografisch kleiner als das zweite Array ist.

Aufgabe 9-3

Operationen auf Arrays

Hausaufgabe

In dieser Aufgabe sollen Sie ein Programm mit einer grafischen Benutzeroberfläche implementieren, welches die unten beschriebenen Operationen auf Arrays zur Verfügung stellt.

- a) Die grafische Benutzeroberfläche soll wie folgt aussehen:



Es soll zwei Buttons, einen zum Mischen zweier Arrays und einen zum Mengenvergleich zweier Arrays, geben. Darunter soll ein Ausgabebereich platziert werden, in dem Rückmeldung über das Mischen und den Mengenvergleich gegeben wird.

Schreiben Sie eine Klasse `ArrayFrameH`, die die Hauptklasse dieser grafischen Benutzeroberfläche sein soll und das Fenster erzeugt. Um Ihr Programm ausführen zu können, schreiben Sie eine weitere Klasse `ArrayFrameHMain`, die Sie wie gewohnt im gleichen Ordner wie Ihre Klasse `ArrayFrameH` abspeichern.

- b) Erweitern Sie Ihre Klasse `ArrayFrameH` um eine Ereignisbehandlung für den Button zum Mischen zweier Arrays. Wird der Button zum Mischen zweier Arrays gedrückt, soll der Benutzer zunächst mit Hilfe der Klasse `JOptionPane` nach den zu mischenden `int`-Arrays gefragt werden und anschließend diese `int`-Arrays gemischt werden, d.h. die Einträge der beiden Arrays Schritt für Schritt durchgegangen und nach dem Reißverschlussverfahren in einem neuen Array mit der Gesamtlänge der beiden Arrays zusammengefügt werden. Es kann davon ausgegangen werden, dass die beiden Arrays gleich lang sind. Das Mischen der Arrays `[0, 2, 4, 6, 8]` und `[1, 3, 5, 7, 9]` ergibt also das Ergebnis `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`. Nach dem Mischen soll der Benutzer im Ausgabebereich über seine Eingabe und Ausgabe informiert werden.

Hinweis: Verwenden Sie wieder die Klasse `Konverter.java` mit den beiden Methoden zum Konvertieren.

- c) Erweitern Sie Ihre Klasse `ArrayFrameH` um eine Ereignisbehandlung für den Button zum Mengenvergleich zweier Arrays, die nach Knopfdruck einzulesen sind. Anders als bei Arrays spielen bei mathematischen Mengen die Reihenfolge der Elemente und Duplikate von Elementen keine Rolle. Obwohl es für Mengen in Java ein eigenes Konstrukt gibt (die Klasse `Set`), kann man sie natürlich auch durch Arrays abbilden, indem man beim Vergleichen von Arrays die Elementreihenfolge und Duplikate außer Acht lässt. Gehen Sie im Folgenden von zwei beliebigen einzugebenden Arrays `arr1` und `arr2` aus. Beachten Sie, dass zwei Mengen A und B genau dann gleich sind, wenn gilt: $A \subseteq B$ und $B \subseteq A$. Nach dem Mengenvergleich soll der Benutzer im Ausgabebereich darüber informiert werden, was seine Eingabe war und ob beide durch die Arrays abgebildeten Mengen gleich sind.

*Besprechung der Präsenzaufgaben in den Übungen ab 15.12.2011. Abgabe der Hausaufgaben bis Mittwoch, 04.01.2012, 14:00 Uhr über UniworX (siehe Folien der ersten Zentralübung). Erstellen Sie zu jeder Aufgabe Klassen, die die Namen tragen, die in der Aufgabe gefordert sind. Geben Sie nur die entsprechenden .java-Dateien ab. Wir benötigen **nicht** Ihre .class-Dateien.*