Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. R. Hennicker, Dr. A. Klarl

WS 2016/17
Übungsblatt 9

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 in grafischer Repräsentation an.

```
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 — Erzeugen von Arrays — Hausaufgabe

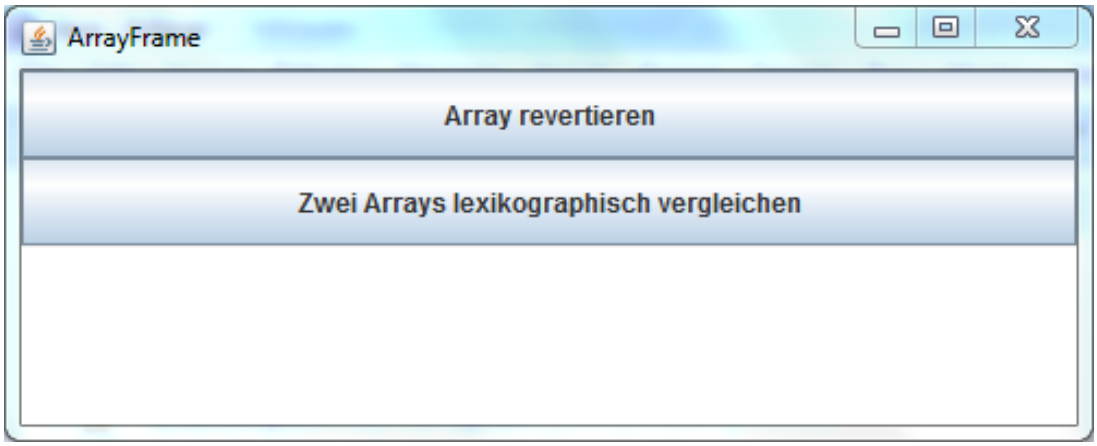
Gegeben ist folgendes Programm, welches nach verschiedenen Prinzipien Arrays erzeugt. Geben Sie für dieses Programm den Zustand des Speichers (Stack und Heap) nach den Zeilen 3, 5, 10, und 12 in grafischer Repräsentation an.

```
1  public class CreateArrays2 {
2    public static void main(String[] args) {
3      char[] a = {'A','L','L','E','N','E','U','N','E'};
4
5      char[] b = new char[a.length/2 + 1];
6      for(int i = 0; i < a.length/2+1; i++) {
7        b[i] = a[i+a.length/2];
8      }
9      b[3] = 'E';
10     b[4] = 'S';
11
12     char[] c = {'G', 'U', 'T', 'E', 'S'};
13   }
14 }
```

Aufgabe 9-3 **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 lexikografischen 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 `public static int[] konvertiereZuIntArray(String string)` einen Komma-separierten String (ohne Leerzeichen) 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 `public static String konvertiereZuString(int[] intArray)` ein `int`-Array in einen Komma-separierten String.

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

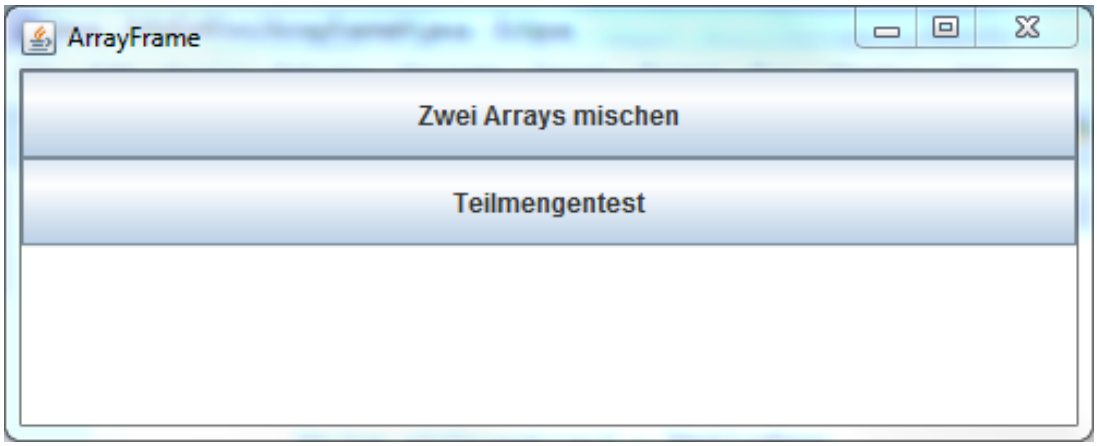
- Entweder $m < n$ und für alle $i = 1, ..., m$ gilt $a_i = b_i$,
- oder es gibt ein $k \in N$ mit $k \leq m$ und $k \leq n$, so dass $a_k < b_k$ gilt und für alle $i \in N$ mit $i < k$ gilt $a_i = b_i$ (also z. B. $[1, 2, 3] < [1, 3, 1]$, hier ist $k = 2$).

Beispielsweise ist das `int`-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 9-4 **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 `int`-Arrays und einen zum Teilmengentest zweier `int`-Arrays, geben. Darunter soll ein Ausgabebereich platziert werden, in dem Rückmeldung über das Mischen und den Teilmengentest 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 `int`-Arrays. Wird der Button zum Mischen zweier `int`-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) Ein Array `a` ist eine Teilmenge eines Arrays `b`, wenn jedes Element von Array `a` auch im Array `b` vorkommt. Erweitern Sie Ihre Klasse `ArrayFrameH` um eine Ereignisbehandlung für den Button zum Teilmengentest von `int`-Arrays. Wird der Button gedrückt, soll der Benutzer zunächst mit Hilfe der Klasse `JOptionPane` nach dem `int`-Array `a` und anschließend nach dem `int`-Array `b` gefragt werden. Nach dem Teilmengentest sollen im Ausgabebereich nochmal die beiden Eingabe-Arrays `a` und `b` gezeigt werden sowie eine Information, ob das `int`-Array `a` Teilmenge des `int`-Arrays `b` ist.

Aufgabe 9-5 **Weihnachtsknobelei** *Hausaufgabe*

Das Christkind macht an Weihnachten eine Rundreise um Geschenke zu verteilen. Auf der Rundreise liegen 5 Lagerorte, die von 0 bis 4 durchnummeriert sind. An jedem Lagerort $i \in \{0, \ldots, 4\}$ liegen g_i Geschenke bereit, die das Christkind alle abholt, wenn es den Ort besucht. Auf dem Weg von einem Lagerort i zum nächsten Lagerort ($i + 1$ falls $i < 4$ bzw. 0 falls $i = 4$) besucht das Christkind k_i Kinder, von denen jedes genau ein Geschenk erhalten soll. Prinzipiell kann

das Christkind an jedem Lagerort seine Rundreise beginnen. Das Christkind muss jedoch darauf achten, dass es den Beginn so wählt, dass es auf seiner Rundreise immer genügend Geschenke zum Verteilen dabei hat.

Ihre Aufgabe ist es, dem Christkind bei der Planung zu helfen und ihm alle Lagerorte zu berechnen, von denen aus es seine Rundreise erfolgreich durchführen kann. Dazu soll ein Java-Programm erstellt werden, das zunächst über eine grafische Benutzeroberfläche ein Array einliest, das für alle Lagerorte i die Anzahl g_i der dort vorhandenen Geschenke angibt. Danach soll ein Array eingelesen werden, das für alle Lagerorte i die Anzahl k_i der auf dem Weg zum nächsten Lagerort zu beschenkenden Kinder angibt. Das Programm soll die Nummern aller Lagerorte ausgeben, von denen aus das Christkind mit einem anfangs leeren (aber genügend großem) Schlitten die Rundreise antreten kann, so dass jedes Kind genau ein Geschenk erhält. Testen Sie Ihr Programm mit einem Array $[3, 6, 8, 7, 4]$ von bereit liegenden Geschenken an den Lagerorten $i = 0, \ldots, 4$ und einem Array $[2, 3, 4, 10, 8]$, das die Anzahl der Kinder zwischen zwei aufeinanderfolgenden Lagerorten angibt.

Besprechung der Präsenzaufgaben in den Übungen ab 14.12.2016. Abgabe der Hausaufgaben bis Mittwoch, 11.01.2017, 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.*