



## Objektorientierte Programmierung

Vorlesung im Wintersemester 2023/2024

Prof. Dr. habil. Christian Heinlein

### 2. Übungsblatt (26. Oktober – 1. November 2023)

#### Aufgabe 2: Reihen in Java

Definieren Sie in einer Datei mit dem Namen `Arrays.java` eine Java-Klasse `Arrays` mit den nachfolgend beschriebenen öffentlichen statischen Methoden! Übersetzen Sie die Klasse (unter Linux) mit `javac Arrays.java` und führen Sie ihre Hauptmethode mit `java Arrays` aus!

- a) Die Methode `print` erhält als Parameter eine Reihe von `double`-Werten mit beliebiger Länge und gibt die Werte nacheinander mit `System.out.print` aus; zwischen den Werten soll jeweils ein Leerzeichen und nach dem letzten Wert mit `System.out.println` ein Zeilentrenner ausgegeben werden.
- b) Die Methode `mirror` erhält als Parameter ebenfalls eine Reihe von `double`-Werten mit beliebiger Länge und dreht die Reihenfolge ihrer Elemente „an Ort und Stelle“ um, d. h. ohne hierfür eine neue Reihe zu erzeugen.
- c) Die Methode `rotate` erhält als Parameter ebenfalls eine Reihe von `double`-Werten mit beliebiger Länge sowie eine nicht-negative ganze Zahl `k` und liefert als Ergebnis eine neue Reihe, die die Elemente der ursprünglichen Reihe um `k` Positionen nach rechts „rotiert“ enthält. (Die ursprüngliche Reihe soll nicht verändert werden.)  
Beispiel: Für eine Reihe mit den Elementen 2, 8, 6, 4, 10 und `k = 2` (oder auch `k = 7`) soll `rotate` eine Reihe mit den Elementen 4, 10, 2, 8, 6 liefern.
- d) Die Methode `zip` erhält als Parameter zwei Reihen von `double`-Werten mit beliebiger Länge und liefert als Ergebnis eine neue Reihe, deren Elemente wie bei einem „Reißverschluss“ abwechselnd aus den Elementen der beiden ursprünglichen Reihen bestehen. Wenn eine Reihe länger ist als die andere, werden ihre überschüssigen Elemente einfach am Ende der Ergebnisreihe angefügt.  
Beispiel: Für Reihen mit den Elementen 3, 7, 9 und 2, 8, 6, 4, 10 soll `zip` eine Reihe mit den Elementen 3, 2, 7, 8, 9, 6, 4, 10 liefern.
- e) Die Hauptmethode `main` enthält für jede der obigen Methoden einige unterschiedliche Testaufrufe und gibt deren Ergebnisse jeweils mit `print` aus.  
Die aktuellen Aufrufparameter der Methoden sollen jeweils durch Reiheninitialisierer konstruiert werden.

Sie können davon ausgehen, dass keine der auftretenden Reihen `null` ist; Reihen der Länge 0 müssen aber korrekt behandelt werden.

Außer `System.out.print` und `System.out.println` dürfen keinerlei Java-Bibliotheksmethoden verwendet werden!

Gerüst der Klasse Arrays:

```
class Arrays {
    // Reihe a elementweise in einer Zeile ausgeben.
    public static void print (double [] a) {
        .....
    }

    // Reihenfolge der Elemente der Reihe a umdrehen.
    public static void mirror (double [] a) {
        .....
    }

    // Rotation der Reihe a um k Positionen nach rechts
    // als neue Reihe liefern.
    public static double [] rotate (double [] a, int k) {
        .....
    }

    // Elemente der Reihen a und b per "Reißverschlussverfahren"
    // zu einer neuen Reihe zusammenfügen.
    public static double [] zip (double [] a, double [] b) {
        .....
    }

    // Test.
    public static void main (String [] args) {
        // Reihen a und b durch Reiheninitialisierer konstruieren.
        // Zum Beispiel:
        double [] a = { 3, 7, 9 }, b = { 2, 8, 6, 4, 10 };

        System.out.println("mirror(a)");
        mirror(a); print(a);
        .....

        System.out.println("rotate(b, 2)");
        print(rotate(b, 2));
        .....

        System.out.println("zip(a, b)");
        print(zip(a, b));
        .....
    }
}
```