Einfache Arrays

Annabelle Klarl

Zentralübung zur Vorlesung
„Einführung in die Informatik: Programmierung und Softwareentwicklung“

`http://www.pst.ifi.lmu.de/Lehre/wise-12-13/infoeinf`

WS12/13

Arrays: Wiederholung

Ein Array ist ein Tupel von Elementen gleichen Typs

$$a = [w_1, w_2, w_3, \ldots, w_n]$$

- Reihenfolge relevant: $[w_1, w_2] != [w_2, w_1]$
- Zugriff auf ein bestimmtes Element möglich z.B. `a[0]` $= w_1$

  Achtung! Array: $[\ w_1, w_2, w_3, \ldots, w_n]$

  Position: 0 1 2 … n−1
- Elemente müssen den gleichen Typ haben
  - Grunddatentyp z.B. `[1,2,3,4]`, aber nicht `[1,1.0,2]`
  - Klassentyp (später)
  - Arraytyp z.B. `[ [1,2],[3,4] ]`, aber nicht `[ [1,2],[1.0,2.0] ]`

Aufgabe 1: Arithmetisches Mittel

Schreibe eine Methode, die das arithmetische Mittel aller Zahlen in einem **`int`**-Array berechnet.

Algorithmusidee:

- Summiere alle Elemente des Arrays auf
- Teile die Summe durch die Anzahl der Elemente

```
public static double durchschnitt(int[] daten) {
    int sum = 0;
    for (int i = 0; i < daten.length; i++)
        sum = sum + daten[i];
    return (double)sum / daten.length;
}
```

Array `daten` muss mind. ein Element enthalten, sonst Division durch 0!

Aufgabe 2: Sortierung

Schreibe eine Methode, die überprüft, ob ein `double`-Array aufsteigend sortiert ist.

Algorithmusidee: Durchlaufe das Eingabe-Array elementweise

- Falls **ein** Element größer als sein rechter Nachbar ist, gib `false` zurück
- Falls **alle** Elemente kleiner als ihr rechter Nachbar sind, gib `true` zurück

```
public static boolean istSortiert(double[] arr) {
    for (int i = 1; i < arr.length; i++) {
        if (arr[i - 1] > arr[i])
            return false;
    }
    return true;
}
```

Achtung: Beginne mit dem Index `1`, sonst ist `arr[i-1]` nicht definiert.

Aufgabe 3a: Kleines Einmaleins mit Arrays

Gib das kleine Einmaleins auf der Konsole mit Hilfe von Arrays aus.

Algorithmusidee: (siehe ZÜ5)

```
x * y = produkt
```

- Erzeuge ein leeres doppeltes Array (=> Matrix)
- Für `x`: gehe die Zahlen von 1 bis 10 durch
- Für `y`: gehe für jede Möglichkeit der Zahl `x` nochmal die Zahlen 1 bis 10 durch
- Berechne das Produkt aus den Zahlen `x` und `y` und speichere den Wert in der entsprechenden Zelle des doppelten Arrays.

Aufgabe 3a: Kleines Einmaleins mit Arrays

```
public class EinmaleinsArrays {
    public static void main(String[] args) {

        int[][] elements = new int[10][10];

        for (int i = 1; i <= 10; i++) {

            for (int j = 1; j <= 10; j++) {

                elements[i-1][j-1] = i*j;

            }
        }
    }
}
```

Erzeuge ein doppeltes leeres Array

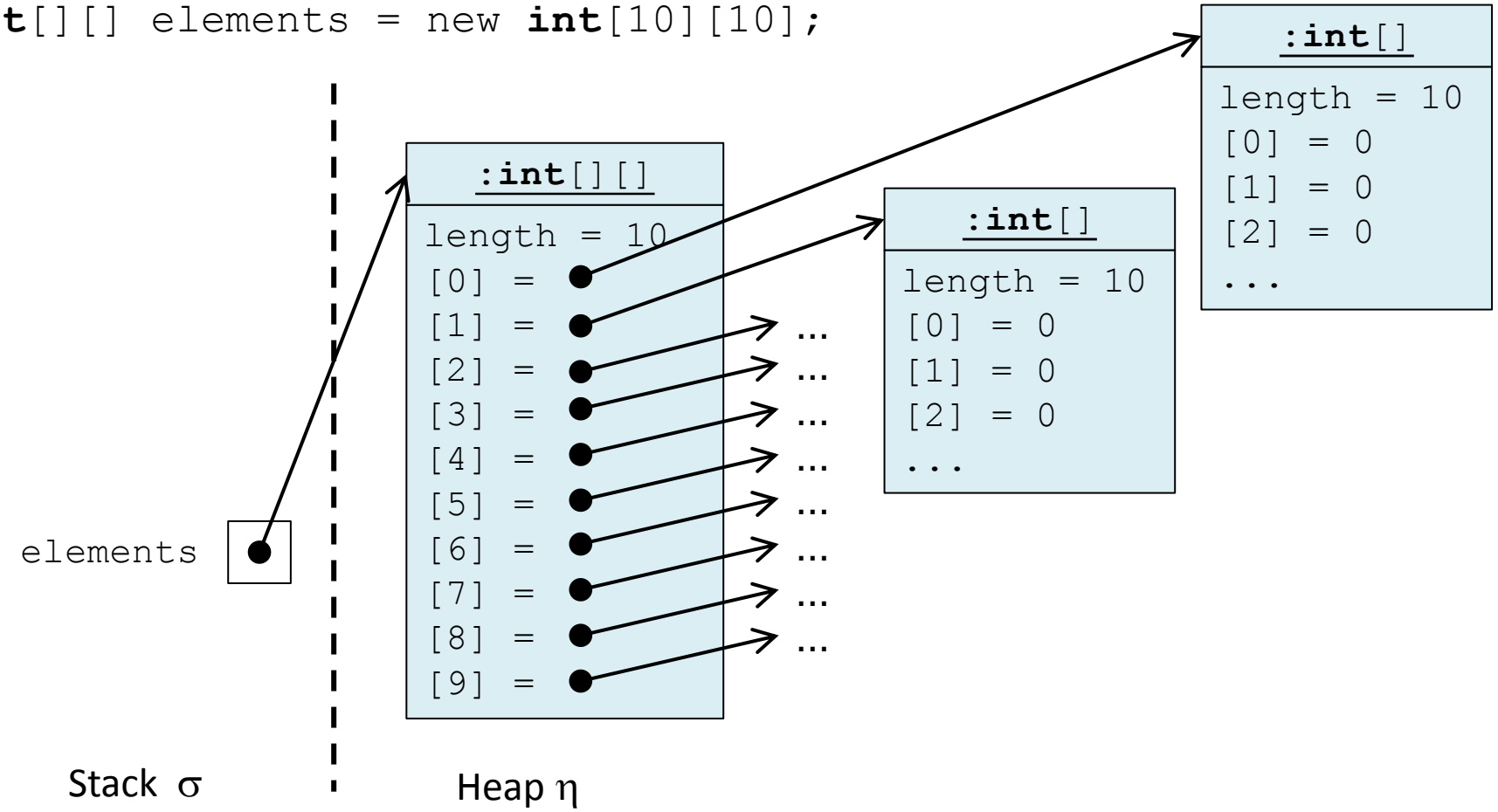
Für `x`: gehe die Zahlen von 1 bis 10 durch

Für `y`: gehe für jede Möglichkeit der Zahl `x` nochmal die Zahlen 1 bis 10 durch

Berechne das Produkt aus den Zahlen `x` und `y` und speichere den Wert.

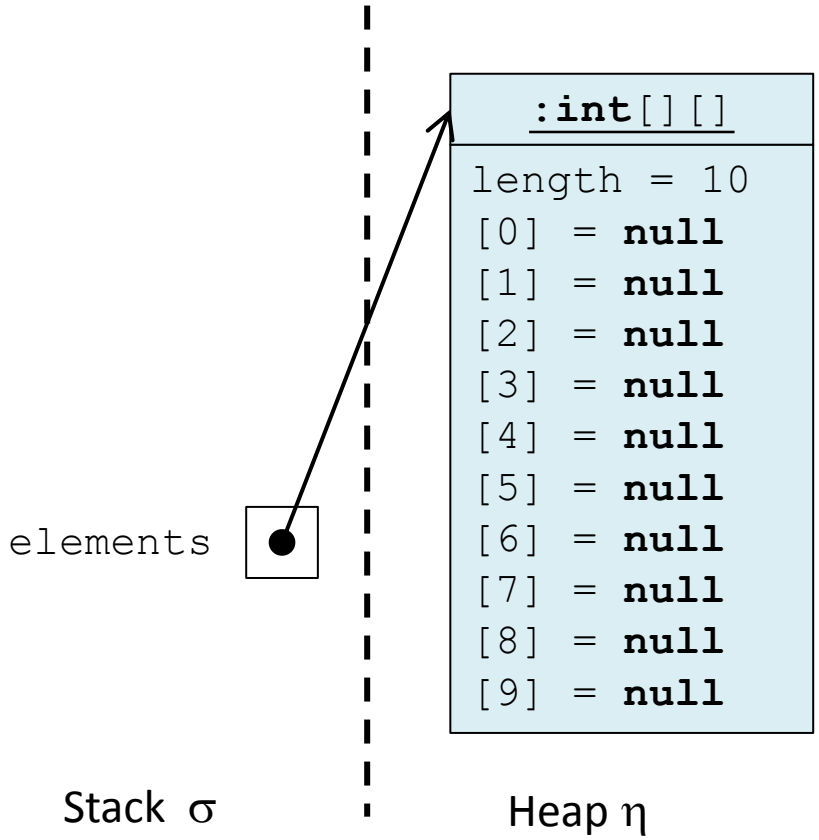
Aufgabe 3b: Speicherdarstellung

```
int[][] elements = new int[10][10];
```



Aufgabe 3b: Speicherdarstellung (partiell initialisiert)

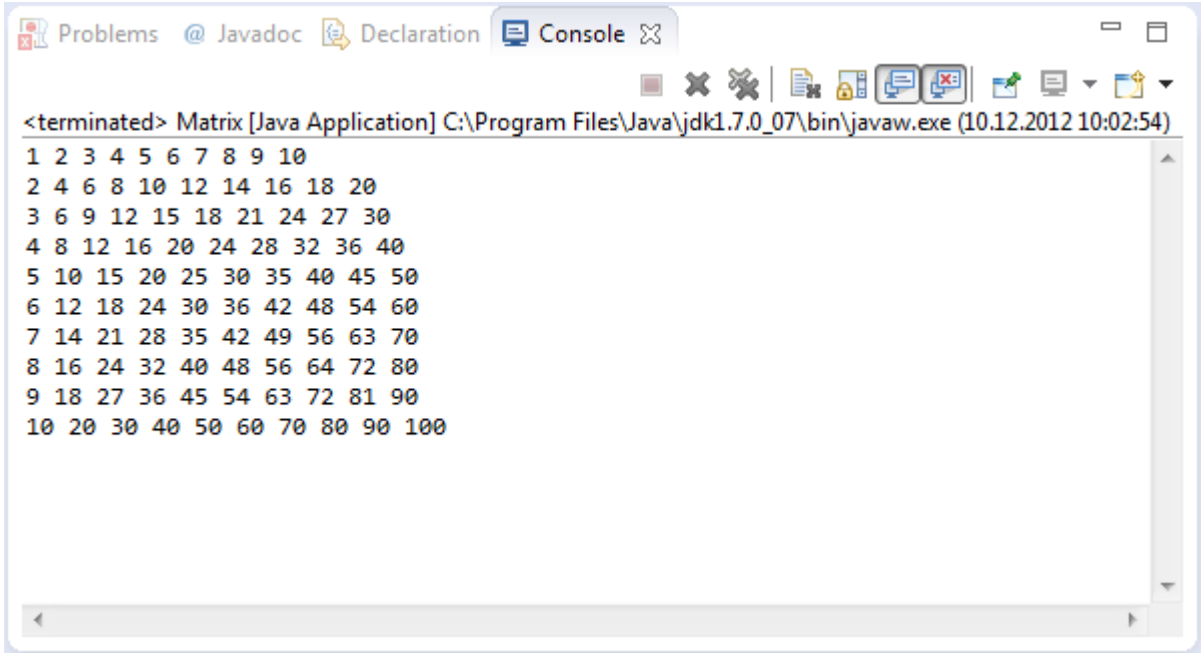
```
int[][] elements = new int[10][];
```



Aufgabe 3c: Ausgeben einer Matrix

```
public class EinmaleinsArrays {
    public static void main(String[] args) {
        int[][] elements = new int[10][10];
        for (int i = 1; i <= 10; i++) {
            for (int j = 1; j <= 10; j++) {
                elements[i-1][j-1] = i*j;
            }
        }
        for (int i = 0; i < elements.length; i++) {
            for (int j = 0; j < elements[i].length; j++) {
                System.out.print(elements[i][j] + " ");
            }
        System.out.println();
        }
    }
}
```

Aufgabe 3c: Ausgeben einer Matrix



Aufgabe 4: Kopieren von Arrays

Kopiere die Inhalte eines Arrays in ein anderes Array.

```
double[] daten = { 1, 2, 3 };
double[] kopie = new double[daten.length];
for (int i = 0; i < daten.length; i++) {
      kopie[i] = daten[i];
}
```

Hinweise: Kopieren von Arrays

Zum vollständigen Kopieren von Arrays kann man auch eine der beiden folgenden Varianten nehmen:

```
int[] daten = { 1, 2, 3 };

int[] kopie1 = daten.clone();

int[] kopie2 = new int[3];
System.arraycopy(kopie2, 0, daten, 0, daten.length);
```

Methode `clone` wird geerbt von Klasse `Object`

… kopiert das Array `daten` beginnend bei Position 0 bis zur Position `daten.length` in Array `kopie2` ab Position 0