

# 10. Felder (Arrays)

Teil 2

Java-Beispiele:

Echo.java Primzahlen.java Monate.java

### **Techniken mit Arrays:**

- Boolean-Anzeige-Feld
- Konstante Arrays
- Parallele Arrays

## Boolean-Arrays als "Anzeige"-Feld

#### Beispiel:

#### Ermittle Primzahlen bis zu einer Grenze n

Spontaner Ansatz: "sammeln" von Primzahlen in einem Array

```
int[] prim;

0  1  2  3  4  5  6
2  3  5  7  11  13  17
```

## Boolean-Arrays als "Anzeige"-Feld

#### Beispiel:

#### Ermittle Primzahlen bis zu einer Grenze.

**Technik:** Sieb des Eratosthenes

#### **Grundidee:**

Streiche alle Vielfachen von bereits als Primzahl erkannten Zahlen.

```
boolean[] sieb:
// Position i entspricht Zahl i
// sieb[i] = true <-> i ist Primzahl
                      3
                                    5
                                              (Index)
  0
               2
                                                Anfangsbelegung
false
      false
                                  true
              true
                    true
                           true
                                        true
false
       false
              true
                     true
                           false
                                  true
                                        false
                                                    Endbelegung
```

Primzahl-Programm

```
Primzahlen.java
boolean[] sieb;
                        Deklaration
int i, j, n;
System.out.print("Primzahlgrenze: ");
n = Keyboard.readInt();
  // jetzt erst: Speicherplatz anfordern
sieb = new boolean[n]; —
                                Erzeugung
  // alle sind potentiell Primzahl:
for (i = 2; i < n; i++)
  sieb[i] = true;
for (i = 2; i < n; i++)
  // streiche alle Vielfachen von i
     for (j = i + i; j < n; j += i)
       sieb[j] = false;
```

- Aufgabe: abschließende Ausgabe der ermittelten Primzahlen
- sieb[0], sieb[1]?

## Konstante Arrays, parallele Arrays

#### Beispiel:

Verwaltung von Daten zu Monaten:

- Namen
- Anzahl der Tage



## Array-Index: ermittelt zusammengehöriges Datenpaar

```
public final static String[] MONTH_NAME =
                "Januar", "Februar", "Maerz",
                                                                    13 Monate -
                "April", "Mai", "Juni",
                "Juli", "August", "September",
                "Oktober", "November", "Dezember"};
         public final static int[] DAYS_OF_MONTH = {
                31, 28, 31, 30,
                31, 30, 31, 30,
                30, 31, 30, 31};
                                                13 Monate, damit der Mai
                                                der 5. Monat sein kann
         final static int MAI = 5;
                                                 *5
                                   3
                                           4
MONTH NAME:
               11.11
                    "Januar" | "Februar" | "Maerz" |
                                         "April"
                                                "Mai"
                                                       "Juni"
DAYS OF MONTH:
               0
                      31
                            28
                                   31
                                          30
                                                 31
                                                        30
```

## Array-Index: ermittelt zusammengehöriges Datenpaar

```
final static int MAI = 5;
public static void main (String[] args) {
   System.out.println(
     "Monat " + MONTH_NAME[MAI]
     + " hat " +
     DAYS_OF_MONTH[MAI] + " Tage");
                         derselbe Index
```

Ausgabe:

Monat Mai hat 31 Tage

## **Zweidimensionale Arrays**

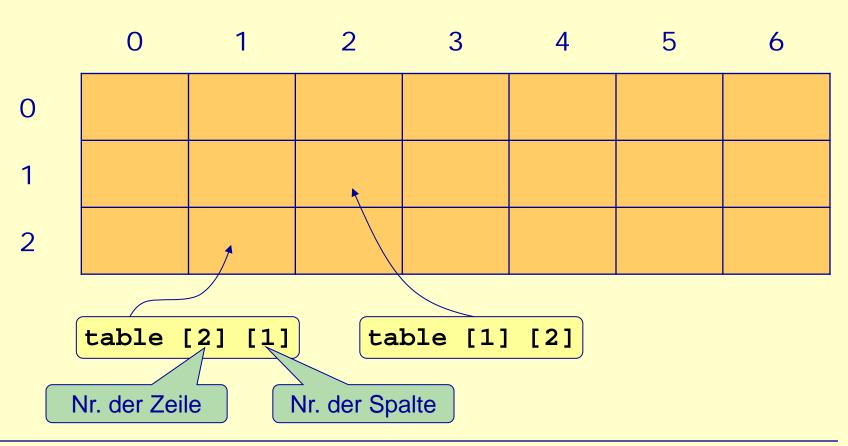
**Mehrdimensionale Arrays** 

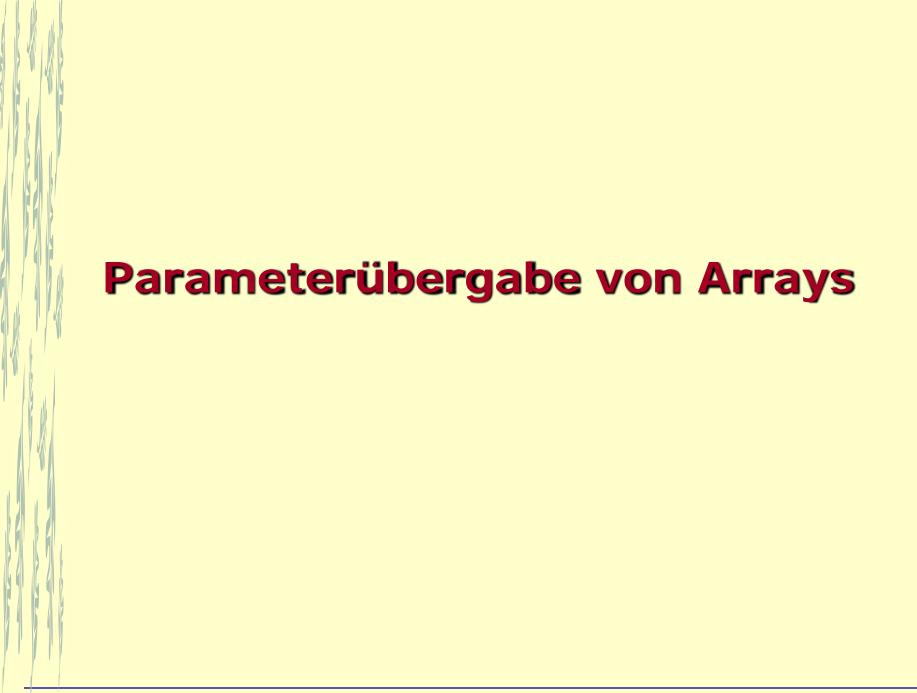
### **Zweidimensionale Arrays: Deklaration**

```
static final int
      NUM DAYS = 7;
      NUM PERS = 3;
   static double[] [] table =
      new double [NUM_PERS] [NUM_DAYS];
   3 Zeilen
                                          7 Spalten
Tabelle:
Jede Person eine Zeile (mit 7 personenbezogenen Daten)
 0
                                             5
```

## Zugriff auf Elemente: doppelte Indizes

```
static double[] [] table =
  new double [NUM_PERS] [NUM_DAYS];
```





## Parameterübergabe: Java

```
fakultaet(x);
System.out.println("Grad C");
```

- Grundsätzlich: nur Werteparameter (insb. alle elementaren Typen, String)
- Objekte (insb. Arrays):

  Referenzparameter

Aktuelle Parameter Vom Typ Array Können anderen Wert bekommen

Also: Art der Parameterübergabe in Java:

- abhängig vom Typ des Parameters
- keine Schlüsselwörter zur Unterscheidung (z. B. VAR)

## Aktuelle Parameter vom Typ Array können anderen Wert bekommen

Methode: Summe der Array-Elemente für beliebig lange Arrays, Array dabei auf 0 gesetzt

```
int summe_delete (int[] vektor, int sum) {
   int gesamt = 0;

   for (int i = 0; i < vektor.length; i++) {
      gesamt += vektor[i];
      vektor [i] = 0; // vektor auf 0 setzen
   }

   sum = 0; // Parameter auf 0 gesetzt
   return gesamt;
}</pre>
```

#### **Anwendung:**

```
int [] v1 = new int[8];
int s = 100
. . . // v1: fuellen
s1 = summe_delete(v1, s);
```

#### **Aktuelle Parameter:**

v1 auf 0 gesetzt:
0 0 0 0 0 0 0 0

s = 100 ungeändert