Die erweiterte for-Schleife

- Mit Java 5 wurde die `for`-Schleife um eine syntaktische Variante erweitert, die den Umgang mit Feldern und Collections vereinfachen soll.
- Man nennt diese Variante auch die "**`foreach`**"-Anweisung.
- Hauptanwendung bei den sogenannten Collections.

Die erweiterte for-Schleife

Syntax:

```
for (FormalerParameter : Ausdruck)
    Anweisung;
```

- ⇨ `FormalerParameter:` eine Parameterdeklaration bestehend aus Datentyp und Variablenname
- ⇨ `Ausdruck:` ist ein Array oder ein Objekt des Typs **`java.lang.Iterable`**, z. B. ein Collection-Objekt

Wirkungsweise:

- ⇨ Die in **`FormalerParameter`** definierte Variable durchläuft nacheinander alle Elemente des durch **`Ausdruck`** definierten Objekts und kann im Schleifenrumpf angesprochen werden.

Die erweiterte for-Schleife

```
int [] tab1 = { 12, 34, 67, 2, 4, 9, 17};

// "normale" for-Schleife
for (int i = 0; i < tab1.length; i++)
    System.out.printf("%4d", tab1[i]);

System.out.printf("\n");

// neue Syntax: lies: "for each zahl in tab1"
for (int zahl : tab1)
    System.out.printf("%4d", zahl);

System.out.printf("\n");
```


Die erweiterte for-Schleife

```
/**
 * Minimum in einem Feld bestimmen
 */
public int min(int[] t) {
    int minimum = t[0];
    for (int zahl : t)
        if (zahl < minimum)
            minimum = zahl;
    return minimum;
}
```

Die erweiterte For-Schleife

```
/**
 * Summe über ein Feld berechnen
 */
public int summe(int[] t) {
    int summe = 0;
    for (int zahl : t)
        summe += zahl;
    return summe;
}
```


Beispiel: 2-dimensionale Arrays (1)

```
public class ForEachTest2 {

   /** Allgemeine Funktion zum Ausgeben von beliebigen
    *  int-Matrizen
    *  Anwendung der for-each-Syntax
    */
    public void matrixAusgabe(double[][] mat) {
        for (double[] zeile : mat) {
            for (double element : zeile)
                System.out.printf("%5.1f", element);
            System.out.println();
        }
        System.out.println();
    }
```

Beispiel: 2-dimensionale Arrays (2)

```
/** Allgemeine Funktion zum eingeben von beliebigen
  * int-Matrizen
  *
  */
 public void matrixEingabe(double[][] mat) {
     int i = 0, j = 0;
     for (double[] zeile : mat) {
         System.out.print("mat[" + i++ + "]: "
                          +  zeile.length
                          + " Werte eingeben: ");
         for (j = 0; j < zeile.length; j++) {
             zeile[j] = Stdin.readDouble();
         }
     }
 }
```


Beispiel: 2-dimensionale Arrays (3)

```
    public void start() {
        double[][] matrix1 = new double[2][3];
        matrixEingabe(matrix1);
        matrixAusgabe(matrix1);

        double[][] matrix2 = new double[4][5];
        matrixEingabe(matrix2);
        matrixAusgabe(matrix2);
    }

    public static void main(String[] args) {
        new ForEachTest2().start();
    }
}
```