

Selbstverweise

- Beim Erzeugen eines Objektes wird automatisch die Referenzvariable `this` angelegt.
- `this` verweist auf das eigene Objekt.
- Mittels `this.<Instanzvariable>` kann auf Datenelemente des aktuellen Objekts zugegriffen werden.
- Mittels `this(<Parameterliste>)` können bereits vereinbarte Konstruktoren (mit passender Parameterliste) aufgerufen werden.

Klassenvariablen und -methoden

- werden mit dem Schlüsselwort `static` vereinbart
- sind der Klasse, nicht den Objekten zugeordnet; d.h.
 - **Klassenmethoden** können auch aufgerufen werden, ohne dass ein Objekt der Klasse erzeugt wurde
Beispiele: `main` in Applikationen
`double wurzel = Math.sqrt(25); // ergibt 5`
 - **Klassenvariablen** sind (mit ihren Werten) allen Exemplaren der Klasse gemeinsam, sind also *klassenglobal*
Beispiel: `Math.PI`
- werden mit dem Klassennamen (statt mit dem Objektname) qualifiziert
- Klassenmethoden dürfen direkt nur auf Datenelemente und Methoden zugreifen, die ebenfalls `static` sind!

Arrays (1)

- Arrays sind Verweisdatentypen.
- Arrays stehen alle Methoden zur Verfügung, die von allen Objekten (Instanzen beliebiger Java-Klassen) benutzt werden können.
(Arrays erben von der Klasse `java.lang.Object`.)
- Array-Elemente werden (wie die Datenelemente von Objekten) automatisch initialisiert.

Aber:

- Es gibt keine Klasse, von der Arrays Instanzen sind.
- Arrays haben keine Konstruktoren. Statt dessen gibt es eine spezielle Syntax des `new`-Operators.

Arrays (2)

- Arrays sind immer **eindimensional**, können aber geschachtelt werden (d.h. Arrays als Elemente enthalten).
- Deklaration:

```
int[] bsp;
```

```
int bsp[];
```

```
String[][] aStr;
```

```
String aStr[][];
```

Die Anzahl der Elemente wird erst bei der Initialisierung angegeben.

Initialisierung von Arrays

- direkte Initialisierung (gleichzeitig mit der Deklaration):

```
int[] bsp = {5, 4, 3, 2, 1};
```

```
String[][] aStr = {"ja", "nein"}, {"yes", "no"}, {"?"};
```

- nachträgliche Initialisierung mit dem new-Operator:

```
bsp = new int[5]; // Standardinitialisierung der Elemente
```

```
aStr = new String[3][2];
```

Danach ist z.B. möglich:

```
bsp[0] = 5; bsp[1] = 4; bsp[2] = 3; bsp[3] = 2; bsp[4] = 1;
```

```
int laenge = bsp.length; // ergibt 5
```

Parameterübergabe

Der Parameter der `main`-Methode

```
String[] args
```

wird beim Programmstart mit jenen Zeichenketten initialisiert, die auf der Kommandozeile als Argumente nach dem Namen der Applikation angegeben sind.

```
java BspApplikation das sind 4 Parameter
```

bewirkt die Initialisierung

```
args[0] = "das"      args[1] = "sind"  
args[2] = "4"       args[3] = "Parameter"
```