

---

# **II.1. Grundelemente der Programmierung**

- 1. Erste Schritte**
- 2. Einfache Datentypen**
- 3. Anweisungen und Kontrollstrukturen**
- 4. Verifikation**
- 5. Arrays**

# 5. Arrays

folge:

14	1	0	8
----	---	---	---

```
folge [0] == 14, ..., folge [3] == 8
```

bestand:

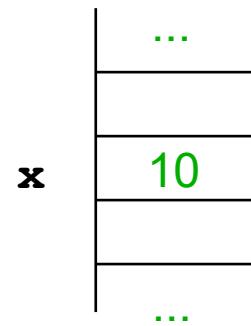
		Ort				
		0	1	2	3	...
Artikel	0	5	0	10	7	
	1	1	3	2	0	
	2	2	17	1	1	
	3	14	1	0	8	
	...					

```
bestand [0] [0] == 5, bestand [0] [1] == 0,  
bestand [0] [2] == 10, ...
```

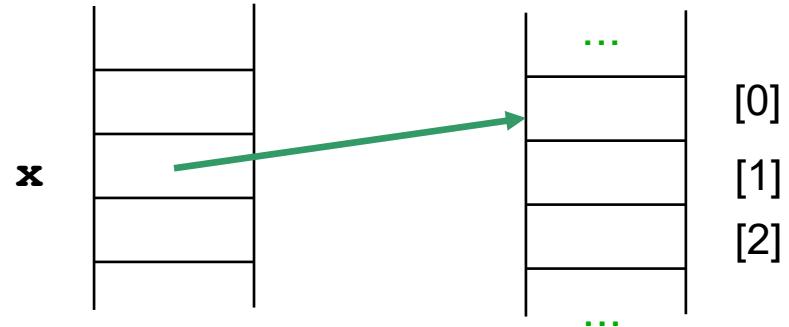
# Wert- und Referenzvariablen

---

```
int x;  
x = 10;
```



```
int [] x;  
x = new int [3];  
x [0] = 14;  
x [1] = 2;  
x [2] = 5;
```

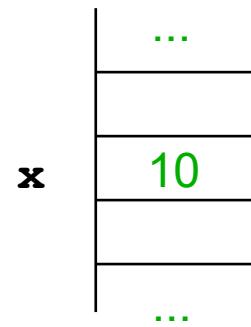


**Primitive Datentypen:** Variablen speichern Werte

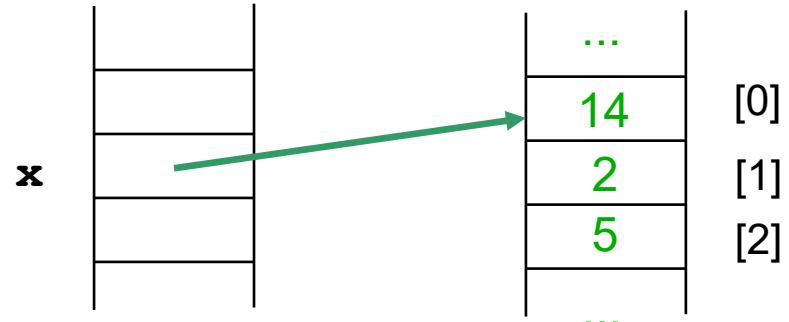
**Andere Datentypen** (Arrays, Strings, ...): Variablen speichern Verweise

# Wert- und Referenzvariablen

```
int x;  
x = 10;
```



```
int [] x;  
x = new int [3];  
x [0] = 14;  
x [1] = 2;  
x [2] = 5;
```



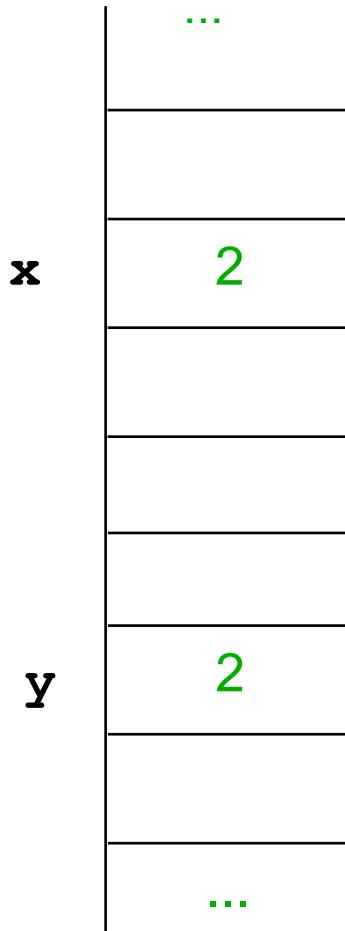
**Primitive Datentypen:** Variablen speichern Werte

**Andere Datentypen** (Arrays, Strings, ...): Variablen speichern Verweise

# Zuweisung bei Wertvariablen

---

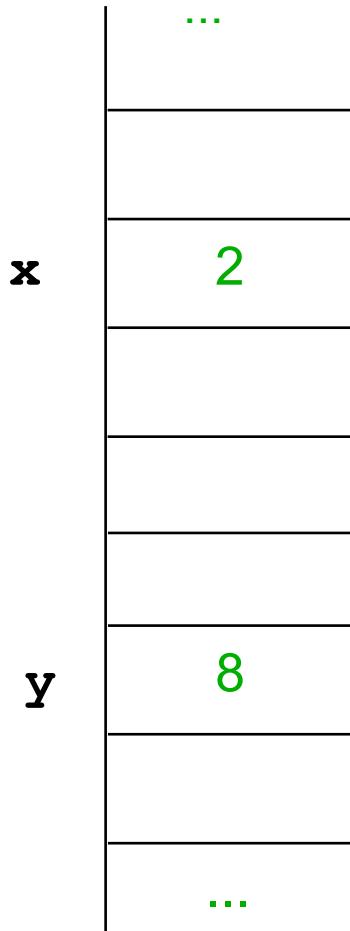
```
int x = 2;  
  
int y = x;  
  
y = 8;
```



# Zuweisung bei Wertvariablen

---

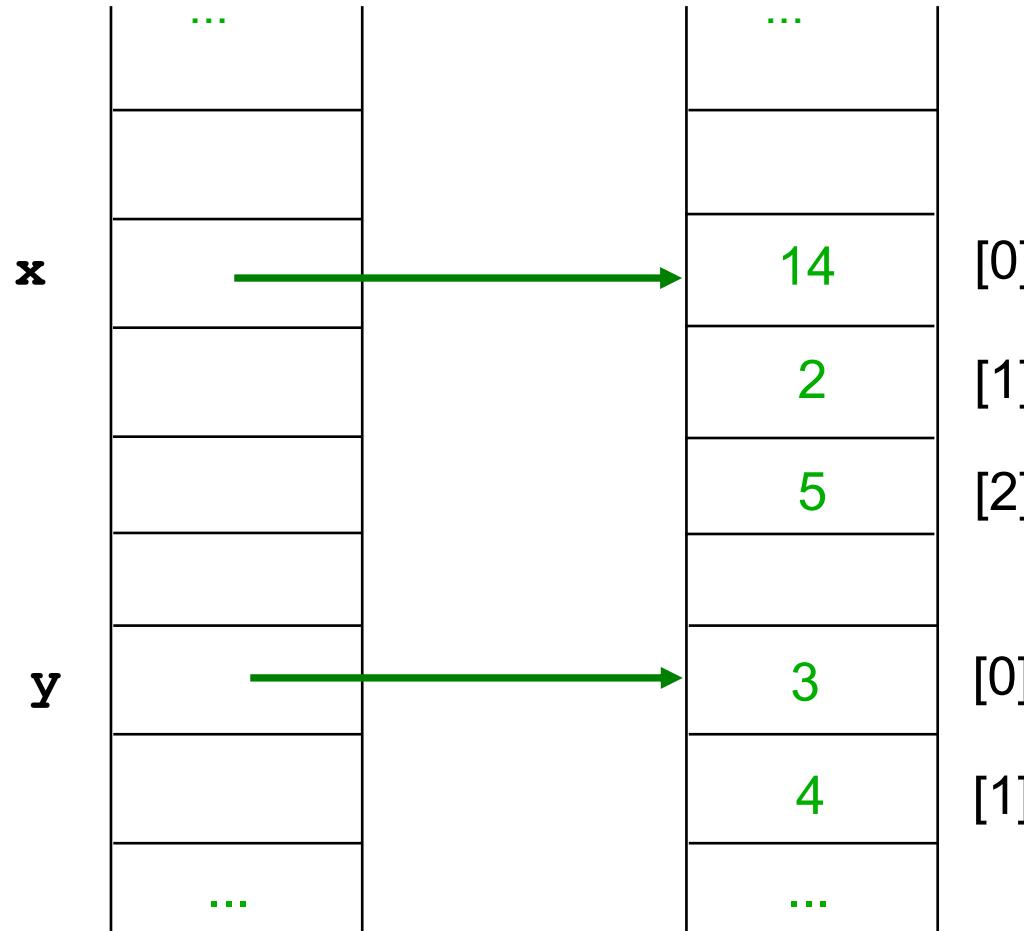
```
int x = 2;  
  
int y = x;  
  
y = 8;
```



Zum Schluss: **x == 2**

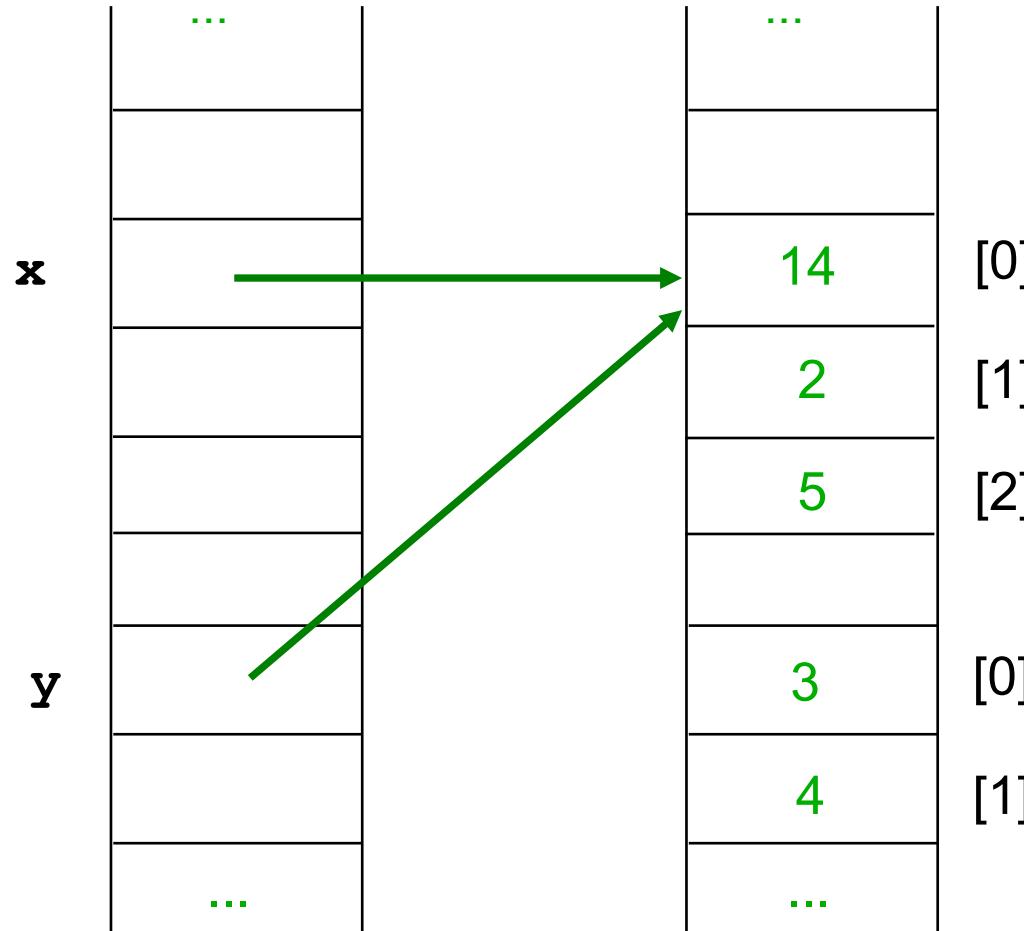
# Zuweisung bei Referenzvariablen

```
int [] x = new int [3];  
  
x [0] = 14;  
x [1] = 2;  
x [2] = 5;  
  
int [] y = new int [2];  
  
y [0] = 3;  
y [1] = 4;  
  
y = x;  
  
y [1] = 8;
```



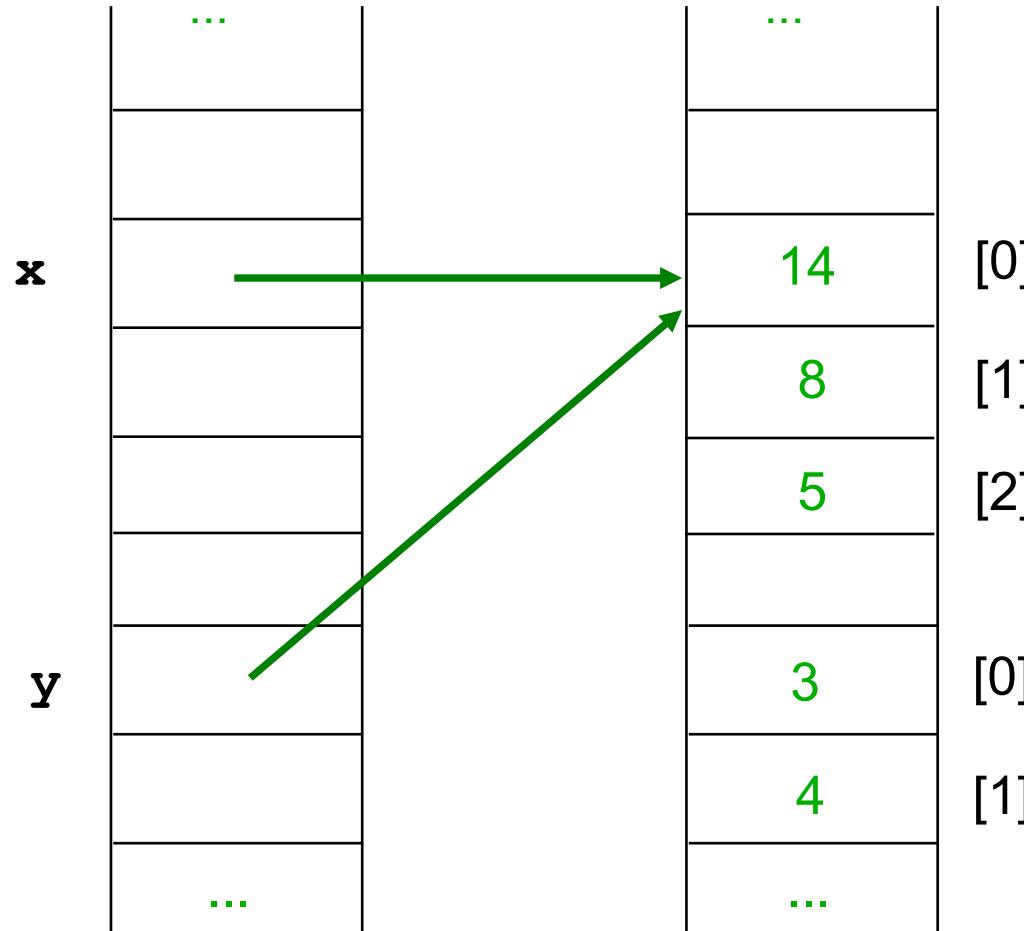
# Zuweisung bei Referenzvariablen

```
int [] x = new int [3];  
  
x [0] = 14;  
x [1] = 2;  
x [2] = 5;  
  
int [] y = new int [2];  
  
y [0] = 3;  
y [1] = 4;  
  
y = x;  
  
y [1] = 8;
```



# Zuweisung bei Referenzvariablen

```
int [] x = new int [3];  
  
x [0] = 14;  
x [1] = 2;  
x [2] = 5;  
  
int [] y = new int [2];  
  
y [0] = 3;  
y [1] = 4;  
  
y = x;  
  
y [1] = 8;
```

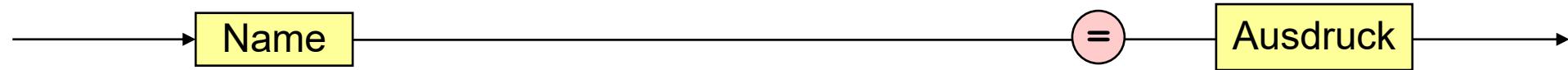


Zum Schluss: **x [1] == 8**

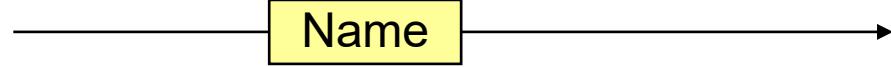
# Zuweisung, Typ

---

Zuweisung



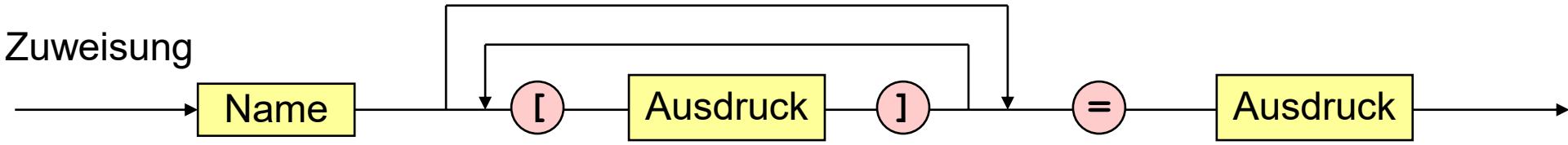
Typ



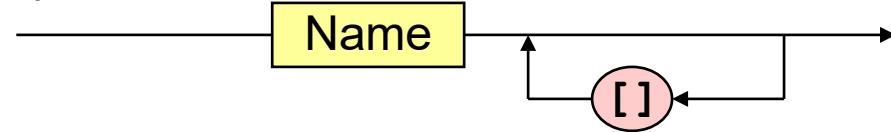
# Zuweisung, Typ

---

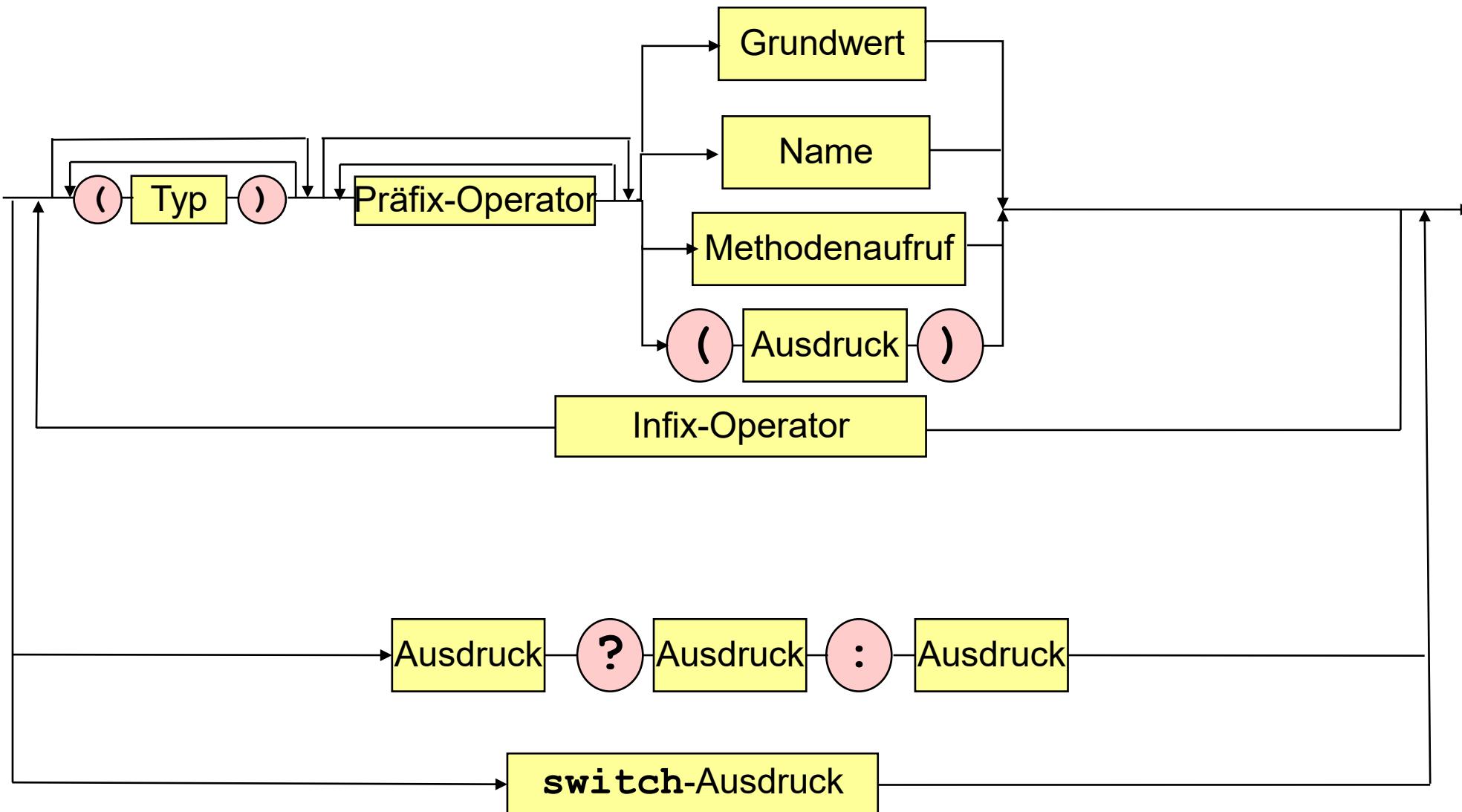
Zuweisung



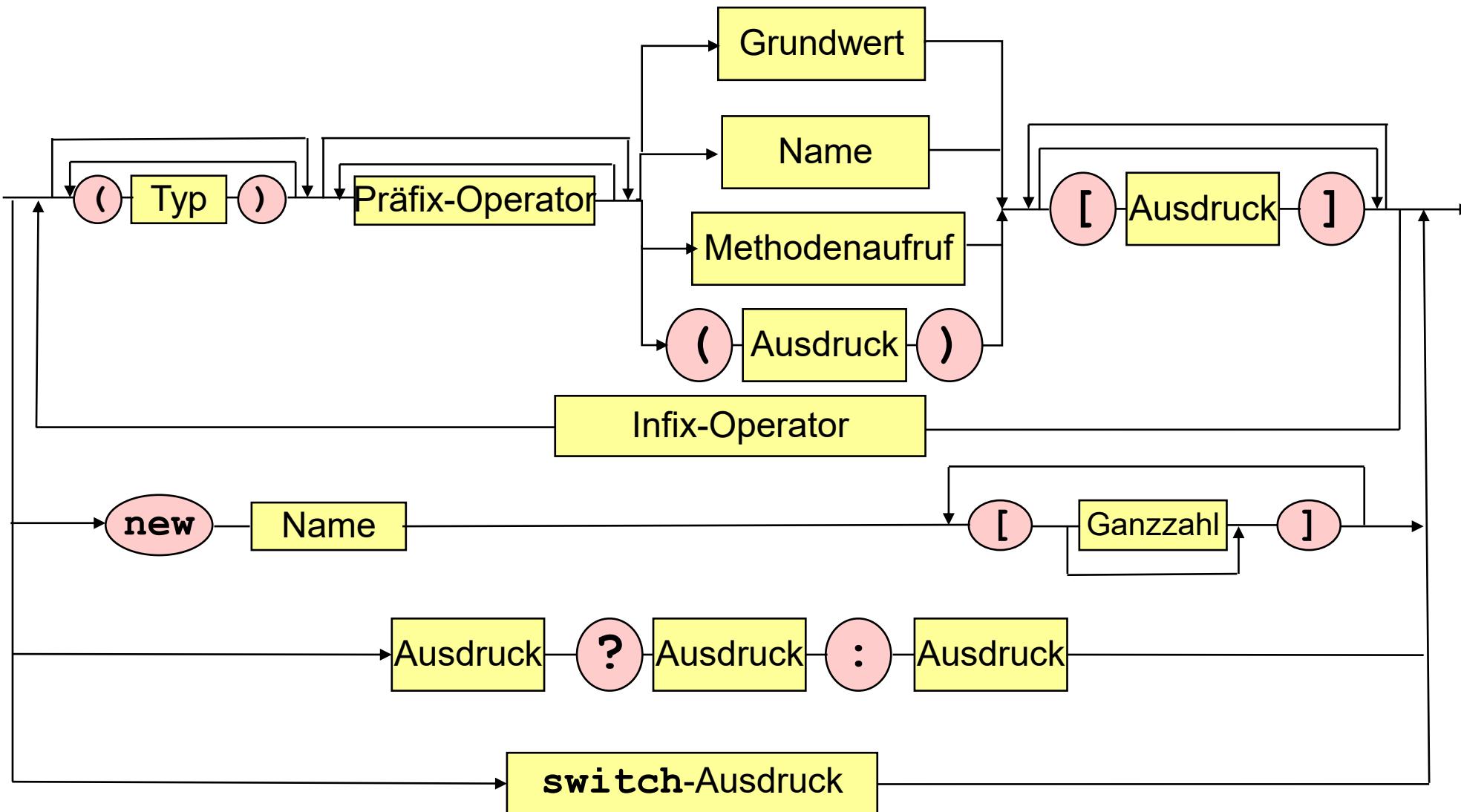
Typ



# Ausdruck



# Ausdruck



# Palindrom-Programm mit Arrays

---

```
void main (String [] args) {  
  
    char [] wort = args[0].toCharArray();  
    boolean palindrom = true;  
  
    for (int i = 0;  
         i <= (wort.length - 2) / 2 && palindrom;  
         i++)  
  
        palindrom = wort [i] == wort [wort.length - 1 - i];  
  
    IO.println(palindrom);  
}
```

# Sort-Programm mit Arrays

```
void main () {  
    int i,j,z;  
    int n = Integer.parseInt(IO.readln("Wieviele Zahlen sortieren? "));  
    int [] a = new int[n];  
  
    //Lies Elemente ein  
    for (i = 0; i < n; i++)  
        a[i] = Integer.parseInt(IO.readln("Gib eine Zahl ein: "));  
  
    //Sortiere Elemente  
    for (i = 0; i < n-1; i++)  
  
        //Vertausche a[i] mit kleinstem Nachfolger  
        for (j = i+1; j < n; j++)  
  
            if (a[i] > a[j]) { //Nachfolger kleiner als a[i]?  
                //Vertausche a[i] und a[j]  
                z = a[i]; a[i] = a[j]; a[j] = z; }  
  
    //Gib sortierte Elemente aus  
    String result = "";  
    for (i = 0; i < n; i++) result = result + a[i] + " ";  
  
    IO.println(result); }
```

# foreach Schleife

---

Array a vom Typ int []

```
for (int i = 0; i < a.length; i++) {  
    int x = a[i];  
  
    IO.print(x);  
  
}
```

```
for (int x : a) {  
  
    IO.print(x);  
  
}
```

# foreach Schleife

---

Array a vom Typ **type** []

```
for (int i = 0; i < a.length; i++) {  
    type x = a[i];  
  
    ...  
  
}
```

```
for (type x : a) {  
  
    ...  
  
}
```

# Sort-Programm mit Arrays

```
void main () {  
    int i,j,z;  
    int n = Integer.parseInt(IO.readln("Wieviele Zahlen sortieren? "));  
    int [] a = new int[n];  
  
    //Lies Elemente ein  
    for (int x : a)  
        x = Integer.parseInt(IO.readln("Gib eine Zahl ein: "));  
  
    //Sortiere Elemente  
    for (i = 0; i < n-1; i++)  
  
    //Vertausche a[i] mit kleinstem Nachfolger  
    for (j = i+1; j < n; j++)  
  
        if (a[i] > a[j]) { //Nachfolger kleiner als a[i]?  
            //Vertausche a[i] und a[j]  
            z = a[i]; a[i] = a[j]; a[j] = z; }  
  
    //Gib sortierte Elemente aus  
    String result = "";  
    for (int x : a) result = result + x + " ";  
  
    IO.println(result); }
```

# Unbenannte Variablen

Array **a** vom Typ **int []**

```
int count = 0;
```

```
for (int element : a) {
```

```
    count++;
```

```
}
```

Auf Variable **element** wird nie zugegriffen

→ braucht keinen Namen

```
int count = 0;
```

```
for (int _ : a) {
```

```
    count++;
```

```
}
```

Auf unbenannte Variablen kann nicht zugegriffen werden

→ mehrfache Vorkommen gelten als  
verschiedene Variablen