
II.1. Grundelemente der Programmierung

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

5. Reihungen (Arrays)

folge:

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

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

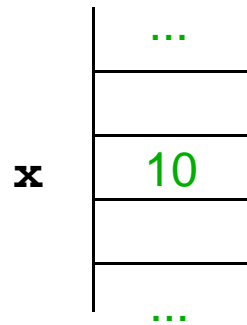
bestand:

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

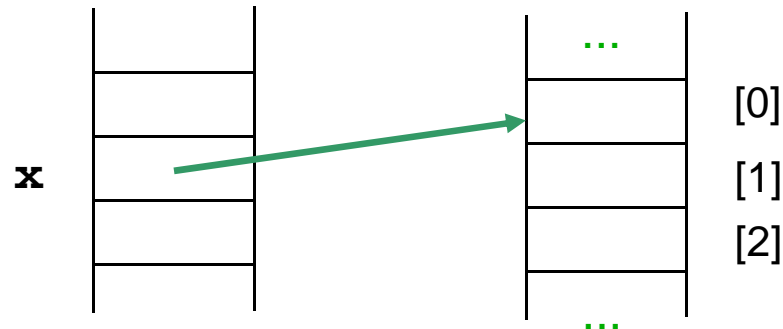
`bestand [0] [0] == 5, bestand [0] [1] == 0,
bestand [0] [2] == 10, ...`

Wert- und Referenzvariablen

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



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

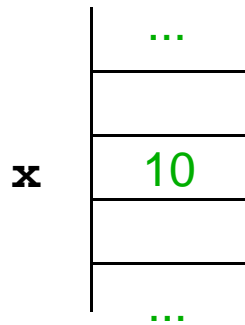


Primitive Datentypen: Variablen speichern Werte

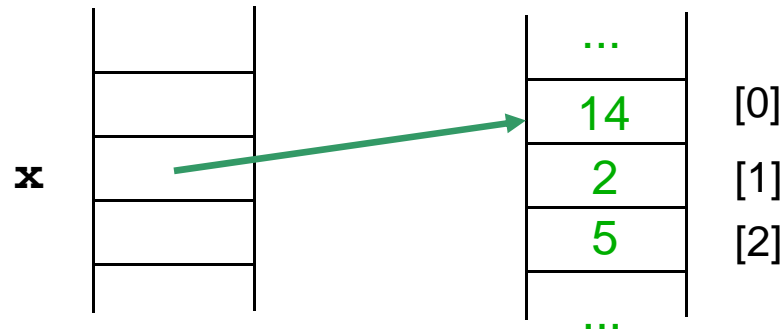
Andere Datentypen (Arrays, Strings, ...): Variablen speichern Verweise

Wert- und Referenzvariablen

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



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

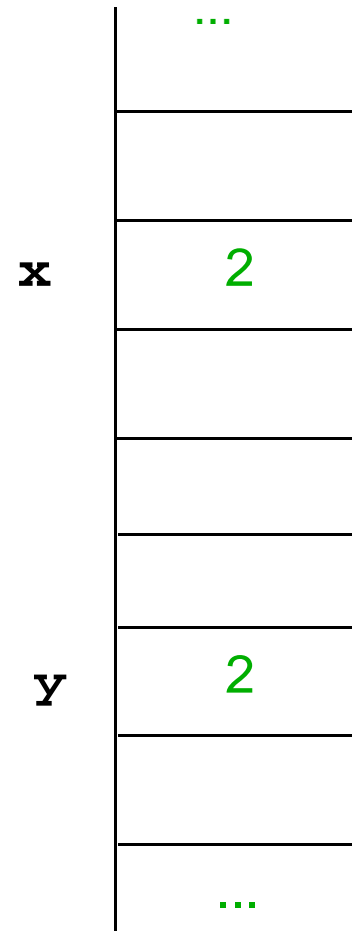


Primitive Datentypen: Variablen speichern Werte

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

Zuweisung bei Wertvariablen

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

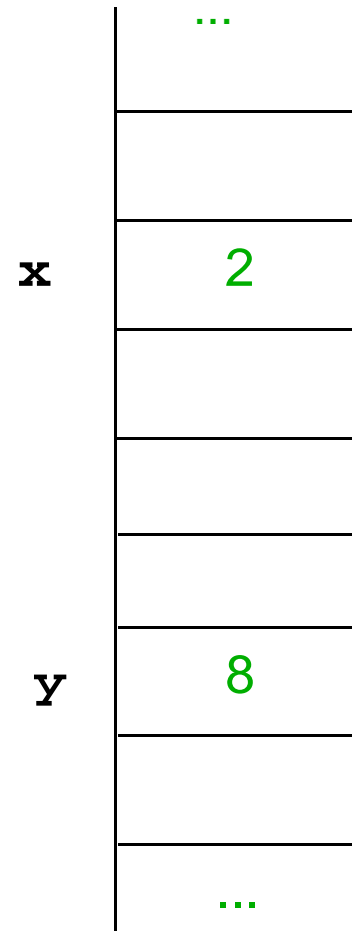


Zuweisung bei Wertvariablen

```
int x = 2;
```

```
int y = x;
```

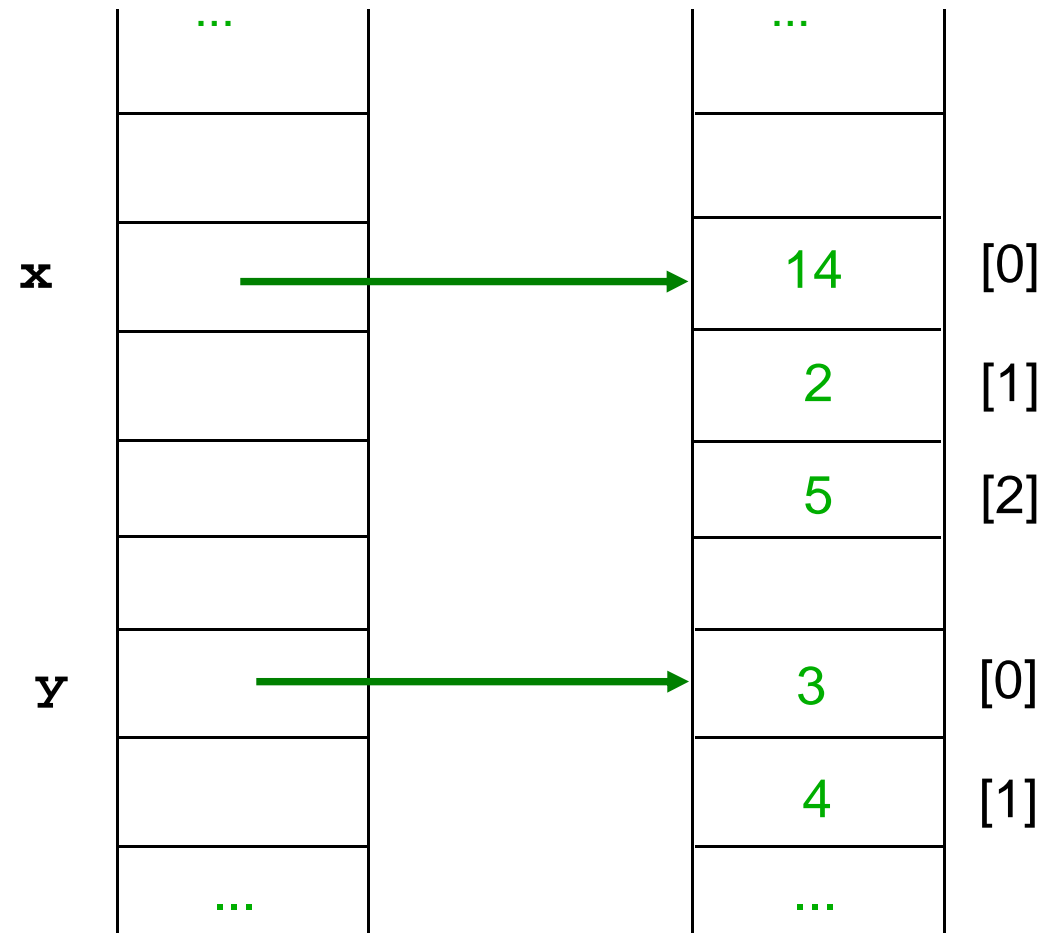
```
y = 8;
```



Zum Schluss: $x == 2$

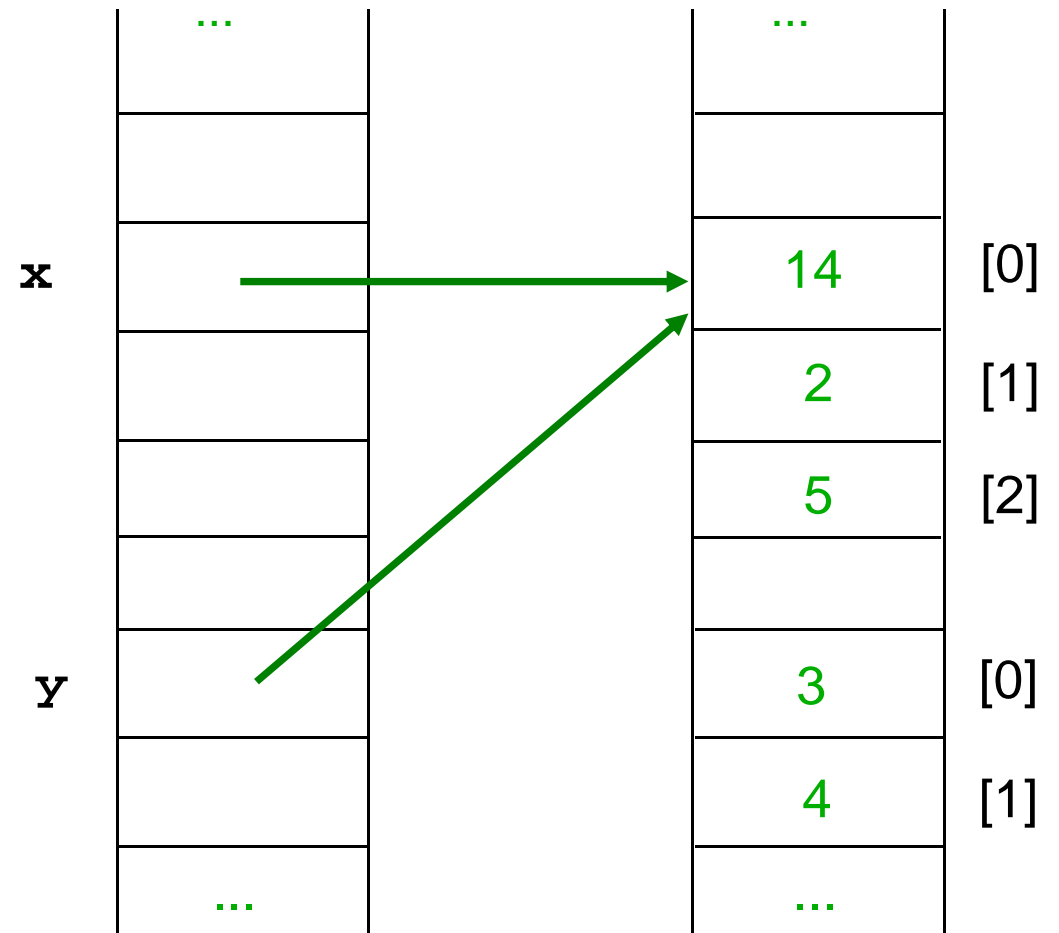
Zuweisung bei Referenzvariablen

```
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;
```



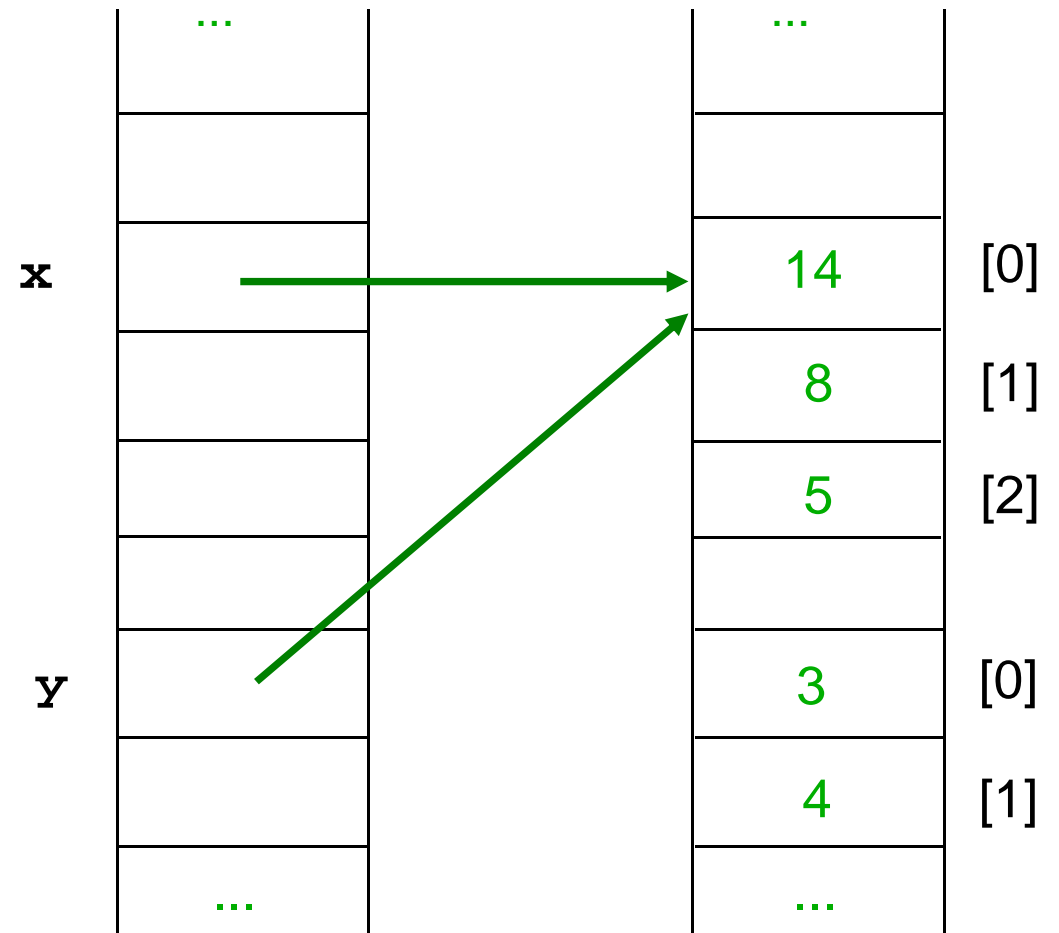
Zuweisung bei Referenzvariablen

```
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

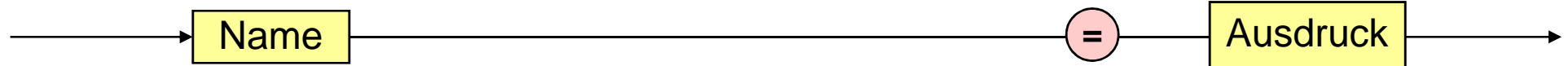
```
int [] x = new int [3];  
  
x [0] = 14;  
x [1] = 2;  
x [2] = 5;  
  
int [] y = new int [2];  
  
y [0] = 3;  
y [1] = 4;  
  
y = x;  
  
y [1] = 8;
```



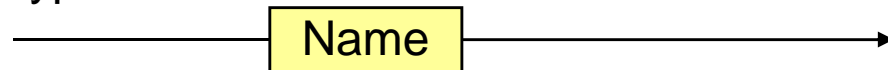
Zum Schluss: `x [1] == 8`

Zuweisung, Typ

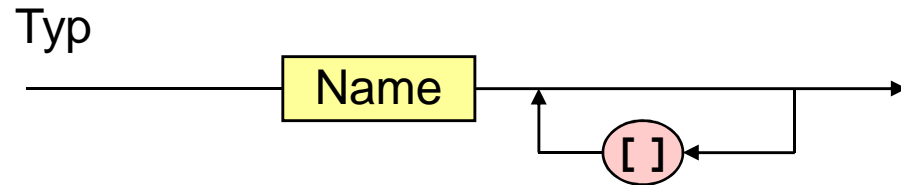
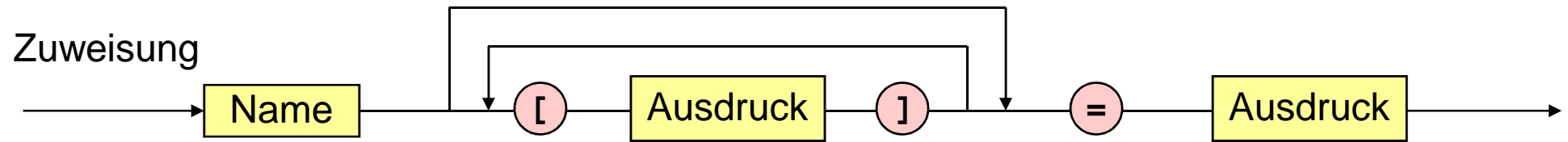
Zuweisung



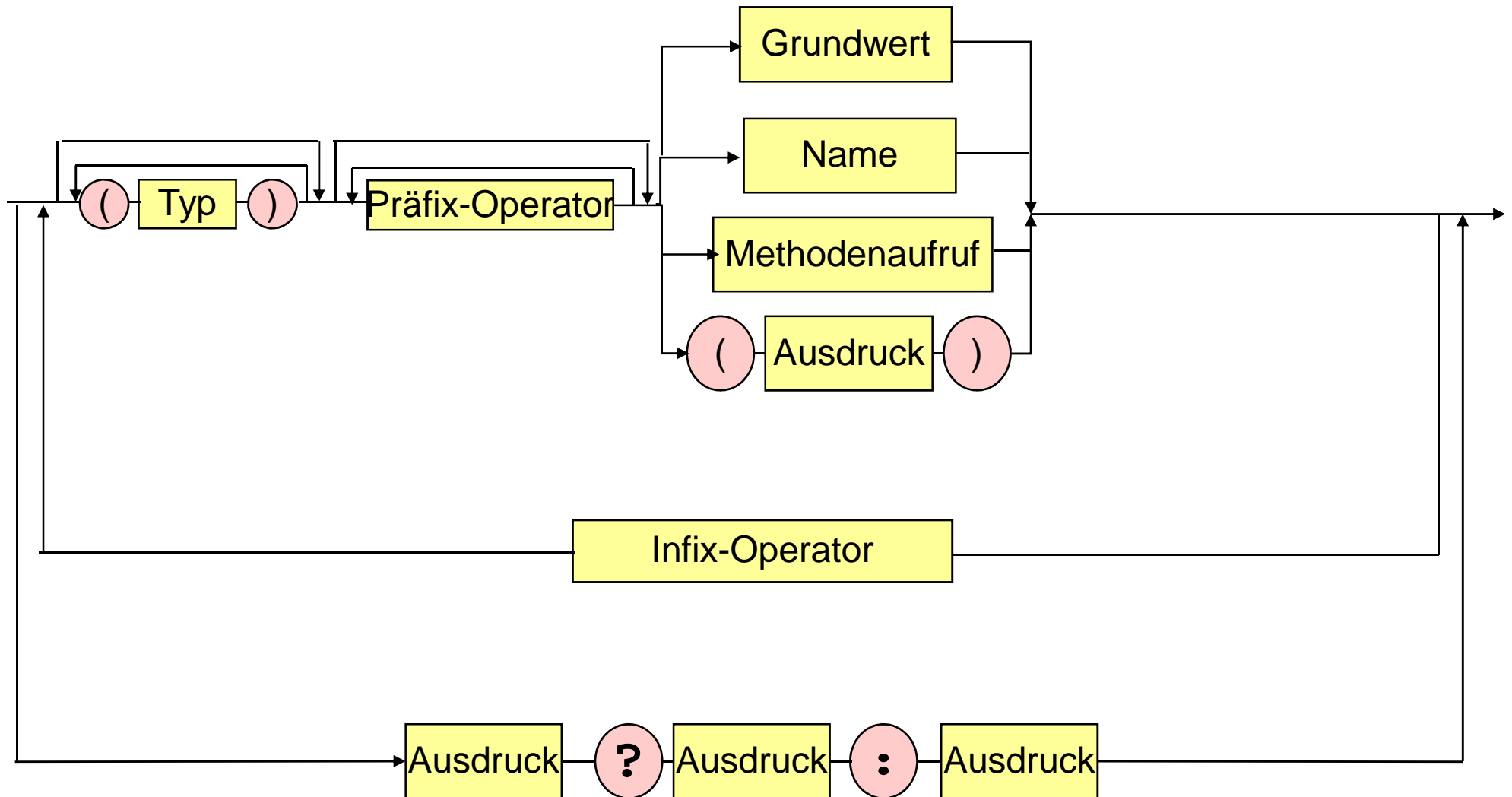
Typ



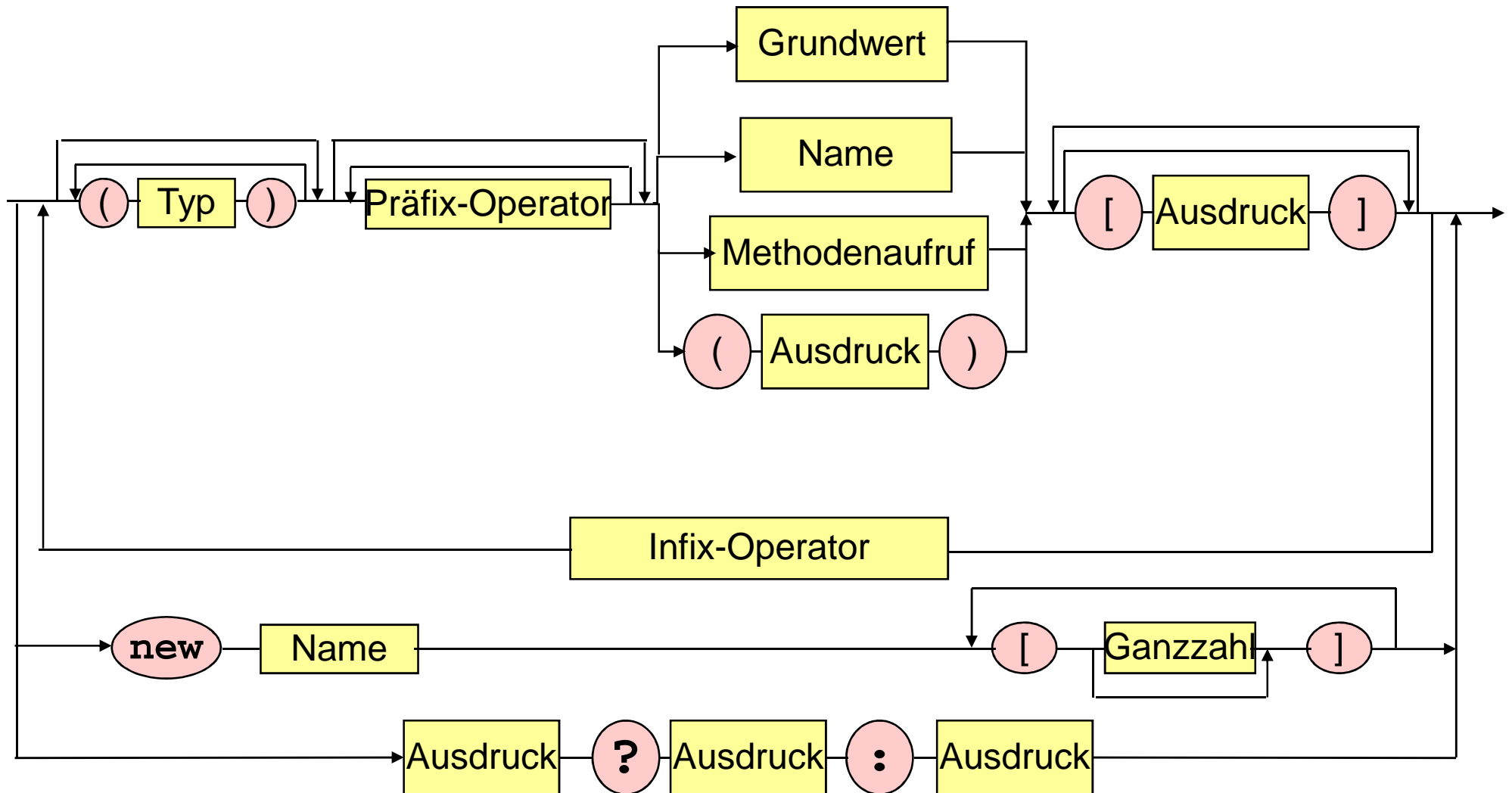
Zuweisung, Typ



Ausdruck



Ausdruck



Palindrom-Programm mit Arrays

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

Sort-Programm mit Arrays

```
public static void main (String [] args) {
    int i,j,z;
    System.out.print("Wieviele Zahlen sortieren? ");
    int n = Integer.parseInt(System.console().readLine());
    int [] a = new int[n];

    //Lies Elemente ein
    for (i = 0; i < n; i++)
        a[i] = Integer.parseInt(System.console().readLine());

    //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
    for (i = 0; i < n; i++) System.out.print(a[i] + " ");
}
```

foreach Schleife

Array a vom Typ `int []`

```
for (int i = 0; i < a.length; i++) {  
    int var = a[i];  
  
    System.out.print(var);  
  
}
```

```
for (int var : a) {  
  
    System.out.print(var);  
  
}
```


foreach Schleife

Array `a` vom Typ `type []`

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

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

Sort-Programm mit Arrays

```
public static void main (String [] args) {  
    int i,j,z;  
    System.out.print("Wieviele Zahlen sortieren? ");  
    int n = Integer.parseInt(System.console().readLine());  
    int [] a = new int[n];  
  
    //Lies Elemente ein  
    for (int x : a)  
        x = Integer.parseInt(System.console().readLine());  
  
    //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  
    for (int x : a) System.out.print(x + " ");  
}
```