

---

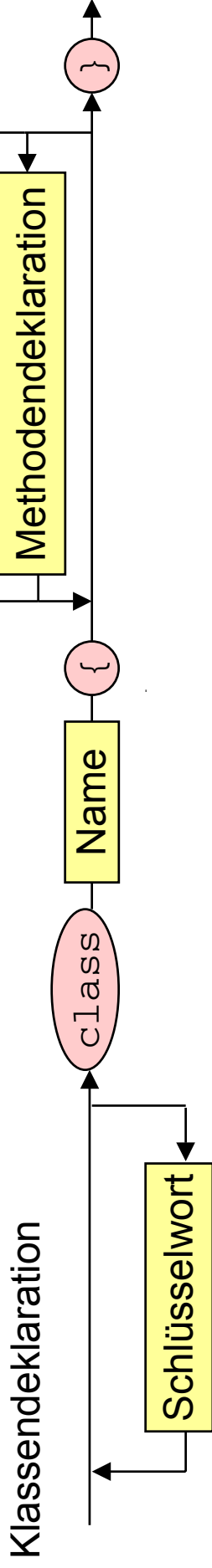
# II.2. Objekte, Klassen und Methoden

- 1. Grundzüge der Objektorientierung
- 2. Methoden, Unterprogramme und Parameter
- 3. Datenabstraktion
- 4. Konstruktoren
- 5. Vordefinierte Klassen

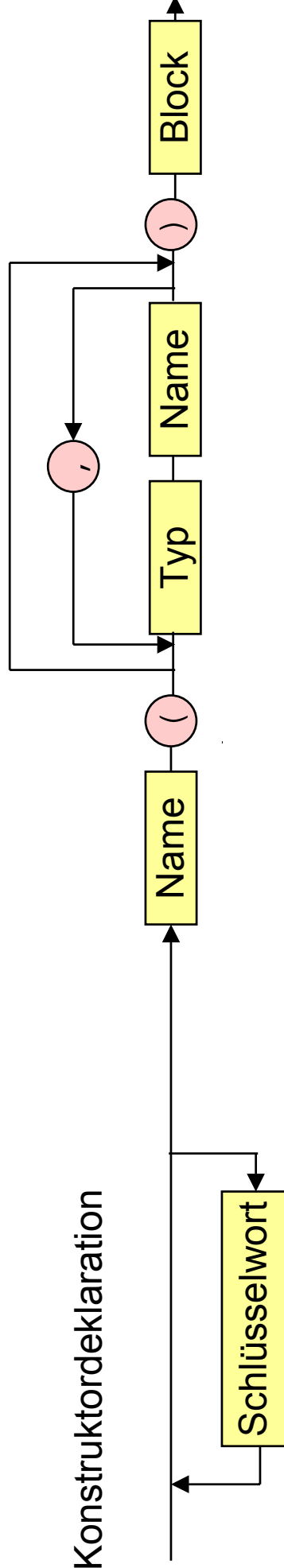
# Konstrukturen

```
public class Rechteck {  
    //Objektattribute  
    double laenge, breite;      int strichstaerke;  
  
    //Konstruktor 1  
    public Rechteck () {  
        laenge = breite = 1.0;   strichstaerke = 1;  
    }  
  
    //Konstruktor 2  
    public Rechteck (double l, double b) {  
        laenge = l;  breite = b;  strichstaerke = 1;  
    }  
    ...  
}  
  
Rechteck r = new Rechteck ();  
Rechteck s = new Rechteck (2.1,1.5);
```

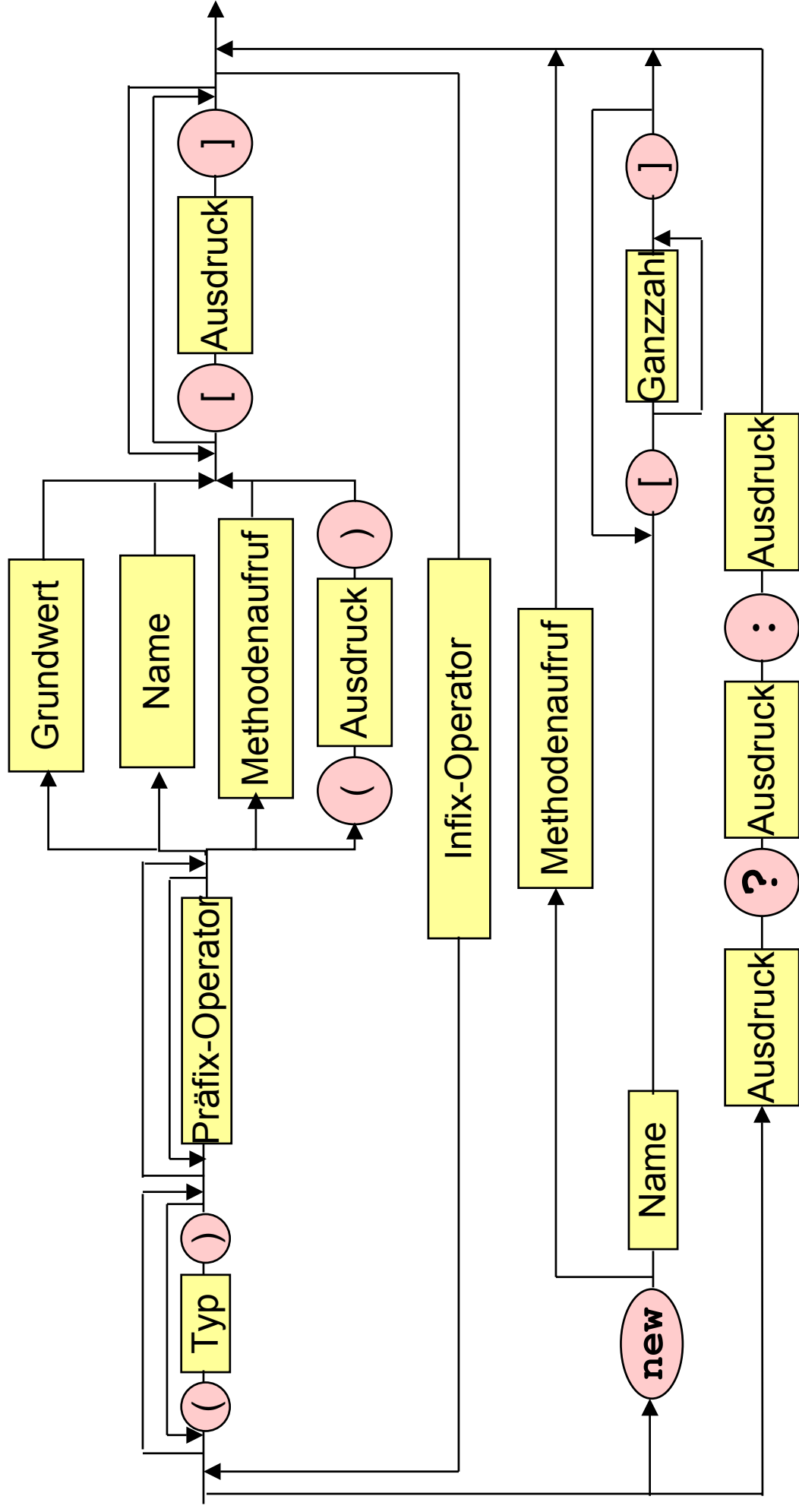
# Konstruktordeklaration



Konstruktordeklaration



# Ausdruck



# Überladene Methoden

---

- **Verschiedene Methoden können den gleichen Namen haben, falls ihre Parameterlisten verschieden sind**
- **Parameterlisten gelten als *verschieden*, falls:**
  - unterschiedliche *Anzahl* von Parametern oder
  - unterschiedliche *Datentypen* bei den Parametern
- **Unterschiedliche *Parameterbezeichner* oder unterschiedliche *Resultattypen* reichen nicht!**
- **Bsp:**
  - Rechteck ( )
  - Rechteck (double l, double b)
  - Rechteck (double kantenlaenge)
  - Rechteck (int seitenlaenge)
- **Es wird die *speziellste* passende Methode ausgeführt.**

# Überladene Konstruktoren

```
public Rechteck () {  
    laenge = breite = 1.0;           strichstaerke = 1;  
}  
  
public Rechteck (double l, double b) {  
    laenge = l;  breite = b;        strichstaerke = 1;  
}  
  
public Rechteck (double kantenlaenge) {  
    laenge = breite = kantenlaenge;  strichstaerke = 1;  
}  
  
public Rechteck (int s) {  
    laenge = breite = 1.0;           strichstaerke = s;  
}
```

```
Rechteck r = new Rechteck ();  
Rechteck s = new Rechteck (2.1, 1.5);  
Rechteck t = new Rechteck (3.0);  
Rechteck u = new Rechteck (3);
```

# Konstruktor mit Selbstverweis

```
public Rechteck () {
    laenge = breite = 1.0;           strichstaerke = 1;
}

public Rechteck (double laenge, double breite) {
    this.laenge = laenge;
    this.breite = breite;
    strichstaerke = 1;
}

public Rechteck (double laenge) {
    this.laenge = breite = laenge;
    strichstaerke = 1;
}

public Rechteck (int strichstaerke) {
    laenge = breite = 1.0;
    this.strichstaerke = strichstaerke;
}
```

# Kopier - Konstruktor

---

```
public Rechteck (Rechteck original) {  
  
    if (original != null) {  
        laenge = original.laenge;  
        breite = original.breite;  
        strichstaerke = original.strichstaerke;  
    }  
    else { // kein Original vorhanden  
        laenge = breite = 1.0;  
        strichstaerke = 1;  
    }  
  
}
```