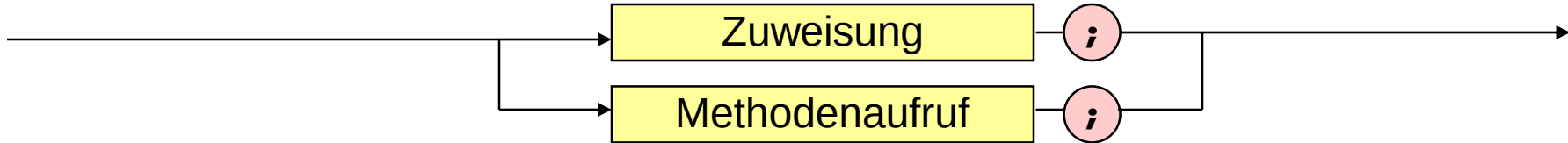

II.1. Grundelemente der Programmierung

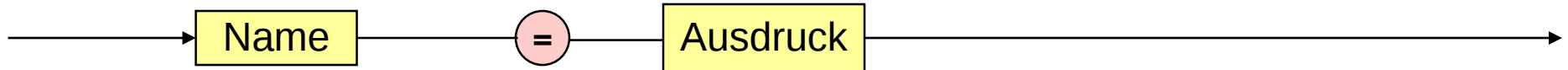
- 1. Erste Schritte
- 2. Einfache Datentypen
- 3. Anweisungen und Kontrollstrukturen
- 4. Verifikation
- 5. Reihungen (Arrays)

Anweisung

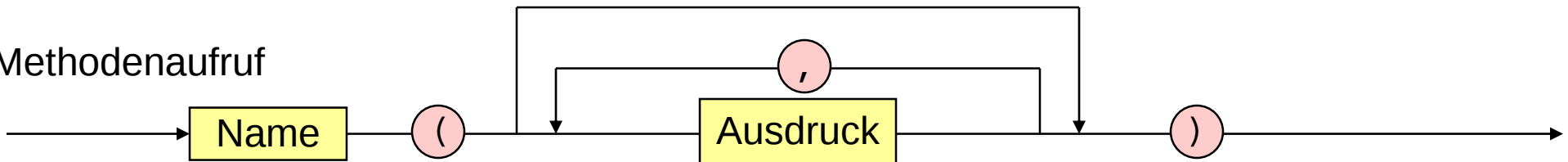
Anweisung



Zuweisung



Methodenaufruf



3. Anweisungen und Kontrollstrukturen

- **Methodenaufruf**
- **Zuweisung**
- **Bedingte Anweisungen (`if`, `switch`)**
- **Schleifen (`while`, `do`, `for`)**
- **Sprunganweisungen (`break`, `continue`)**

3. Anweisungen und Kontrollstrukturen

- **Methodenaufruf**
- Zuweisung
- Bedingte Anweisungen (`if`, `switch`)
- Schleifen (`while`, `do`, `for`)
- Sprunganweisungen (`break`, `continue`)

Methodenaufruf

```
public class Rechnung {  
  
    public static void main (String [] arguments) {  
  
        int x = 10;  
        int y = -1 + 23 * 33 + 3 * 7 * (5 + 6);  
  
        System.out.print ("Das Resultat ist ");  
        System.out.println (x + y);  
  
    }  
  
}
```

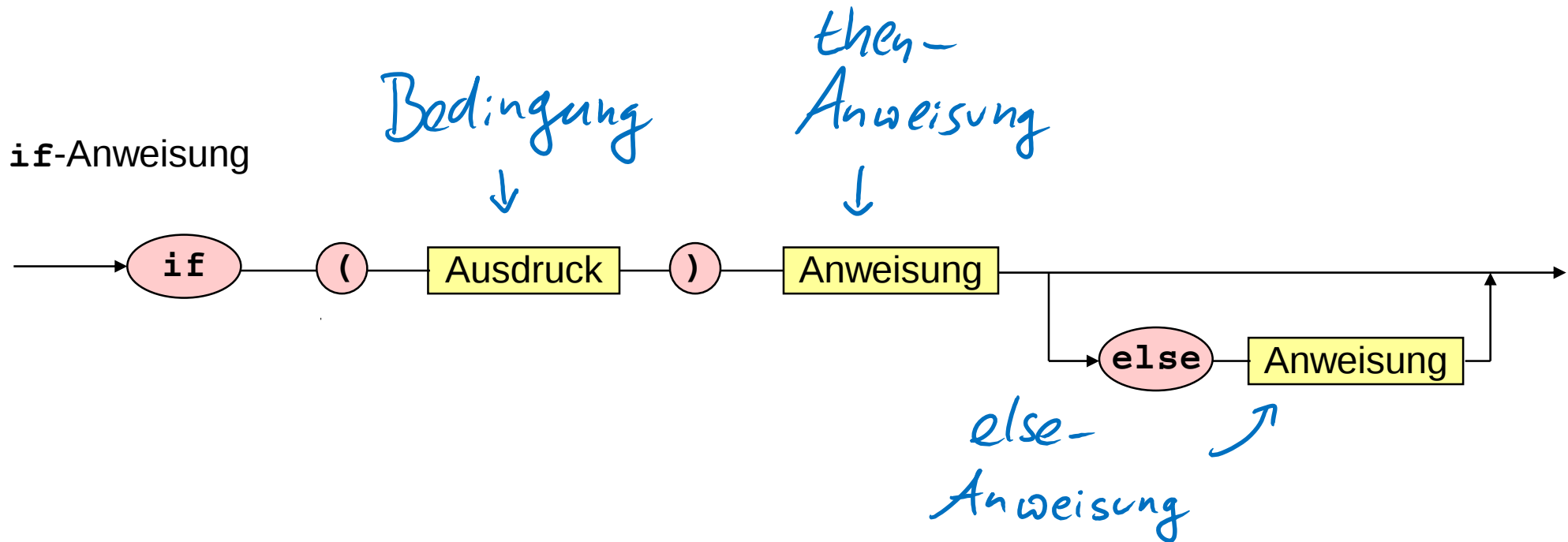
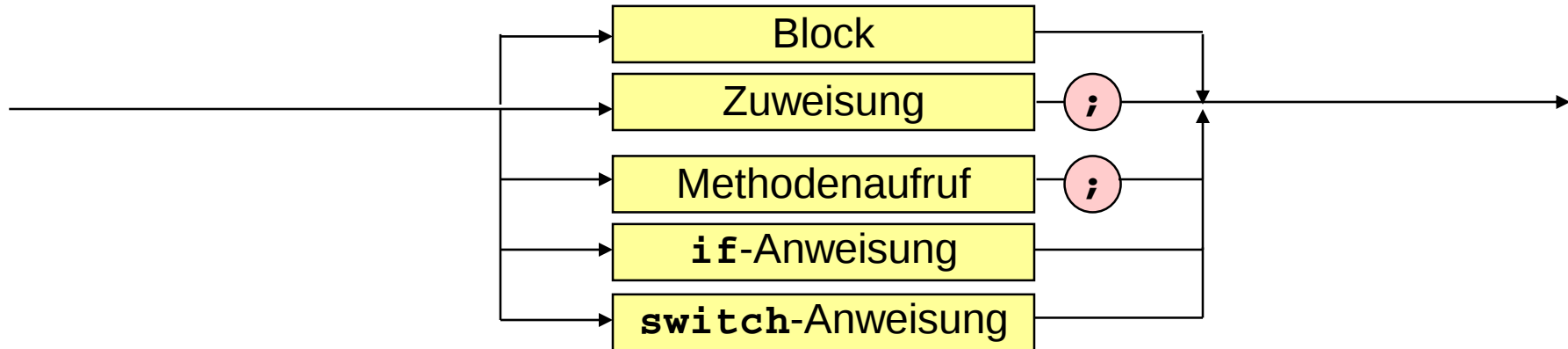
3. Anweisungen und Kontrollstrukturen

- Methodenaufruf
- Zuweisung
- Bedingte Anweisungen (`if`, `switch`)
- Schleifen (`while`, `do`, `for`)
- Sprunganweisungen (`break`, `continue`)

3. Anweisungen und Kontrollstrukturen

- Methodenaufruf
- Zuweisung
- Bedingte Anweisungen (`if`, `switch`)
- Schleifen (`while`, `do`, `for`)
- Sprunganweisungen (`break`, `continue`)

Bedingte Anweisung



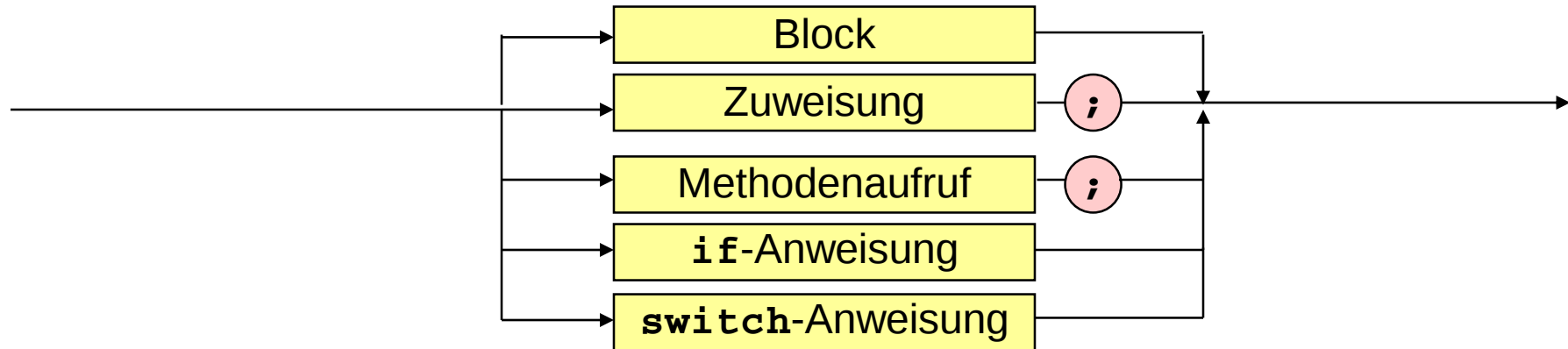
Methoden mit if-Anweisung

```
if (i == 5)
    if (j == 5)
        System.out.println ("i und j sind 5.");
    else System.out.println ("i ist 5, j ist nicht 5.");
```

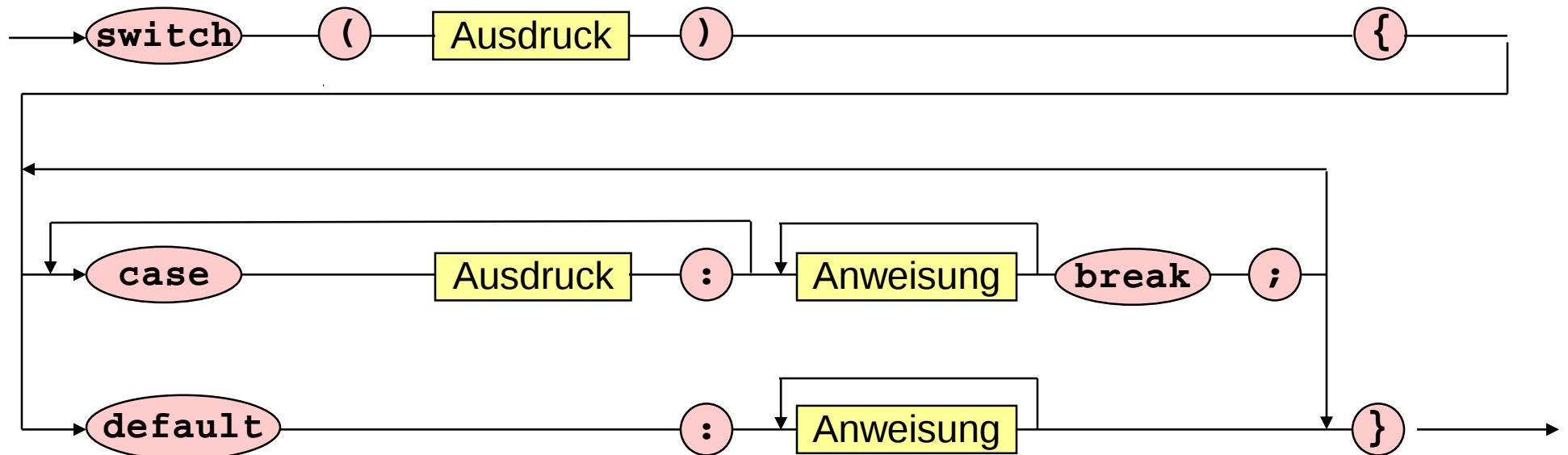
```
if (i == 5) {
    if (j == 5)
        System.out.println ("i und j sind 5.");
}

else System.out.println ("i ist nicht 5, " +
    "keine Aussage fuer j.");
```

Bedingte Anweisung



switch-Anweisung



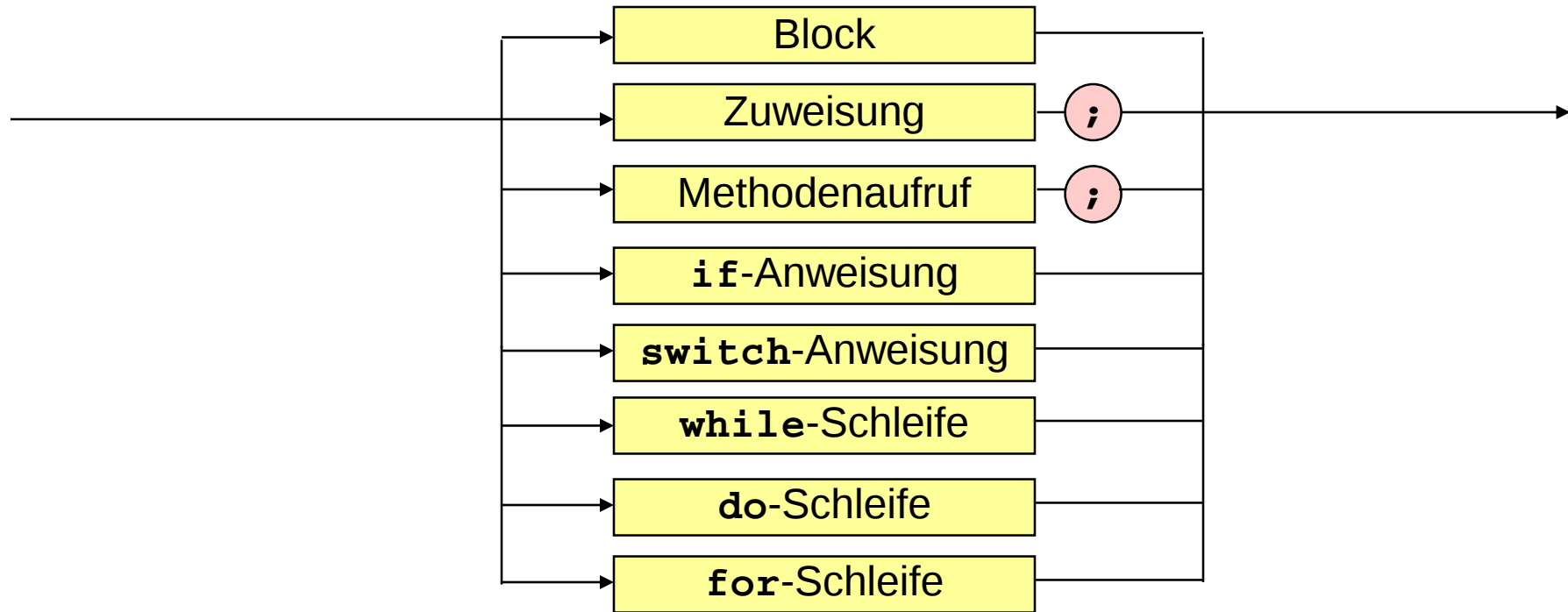
Methode mit switch-Anweisung

```
switch (i) {  
  
    case 0: case 1: case 2: case 3: case 4:  
        System.out.println ("i kleiner 5."); break;  
  
    case 5:  
        System.out.println ("i gleich 5."); break;  
  
    default:  
        System.out.println ("i groesser 5.");  
  
}
```

3. Anweisungen und Kontrollstrukturen

- Methodenaufruf
- Zuweisung
- Bedingte Anweisungen (`if`, `switch`)
- Schleifen (`while`, `do`, `for`)
- Sprunganweisungen (`break`, `continue`)

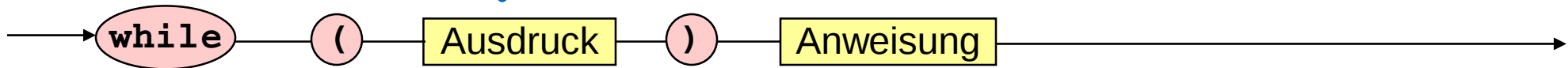
Schleifen



while-Schleife

Bedingung
↓

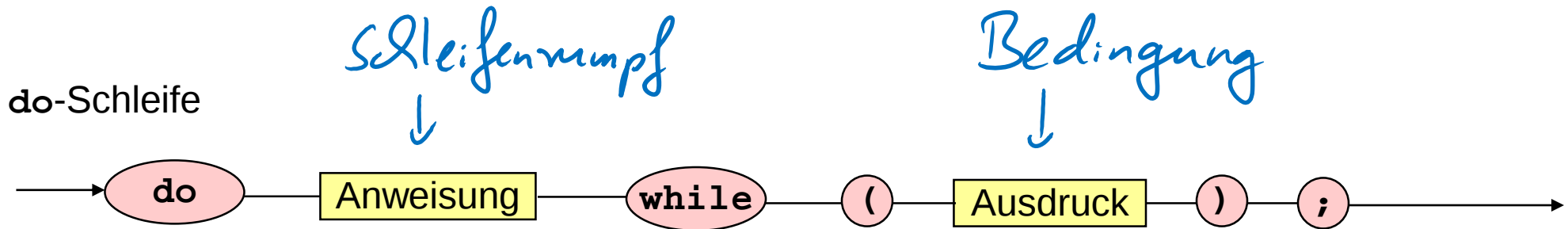
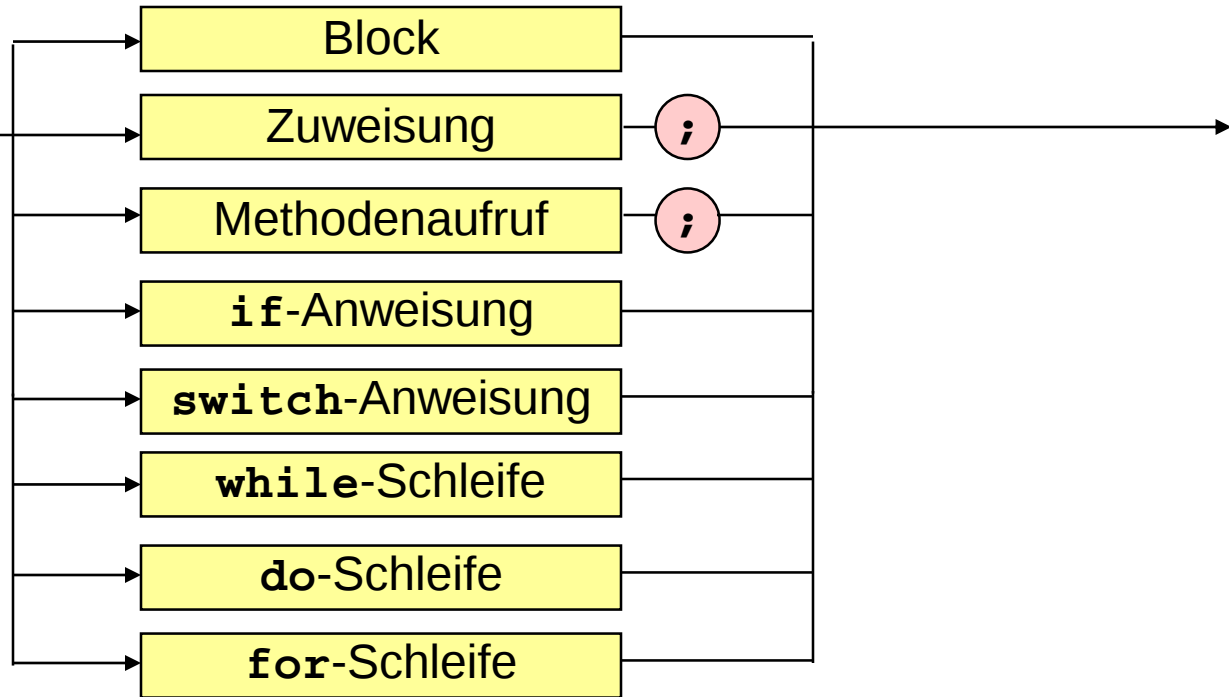
Schleifenrumpf
↓



Methode mit while-Schleife

```
public class Prim {  
    public static void main (String [] args) {  
        int n = SimpleIO.getInt ("Gib eine Zahl ein"),  
            wurzel = (int) Math.sqrt (n),  
            teiler = 2;  
  
        boolean istPrimzahl = true;  
  
        while (teiler <= wurzel && istPrimzahl)  
            if (n % teiler == 0) istPrimzahl = false;  
            else teiler++;  
  
        SimpleIO.output (n + " prim: " + istPrimzahl,  
                        "Ergebnis");  
    }  
}
```

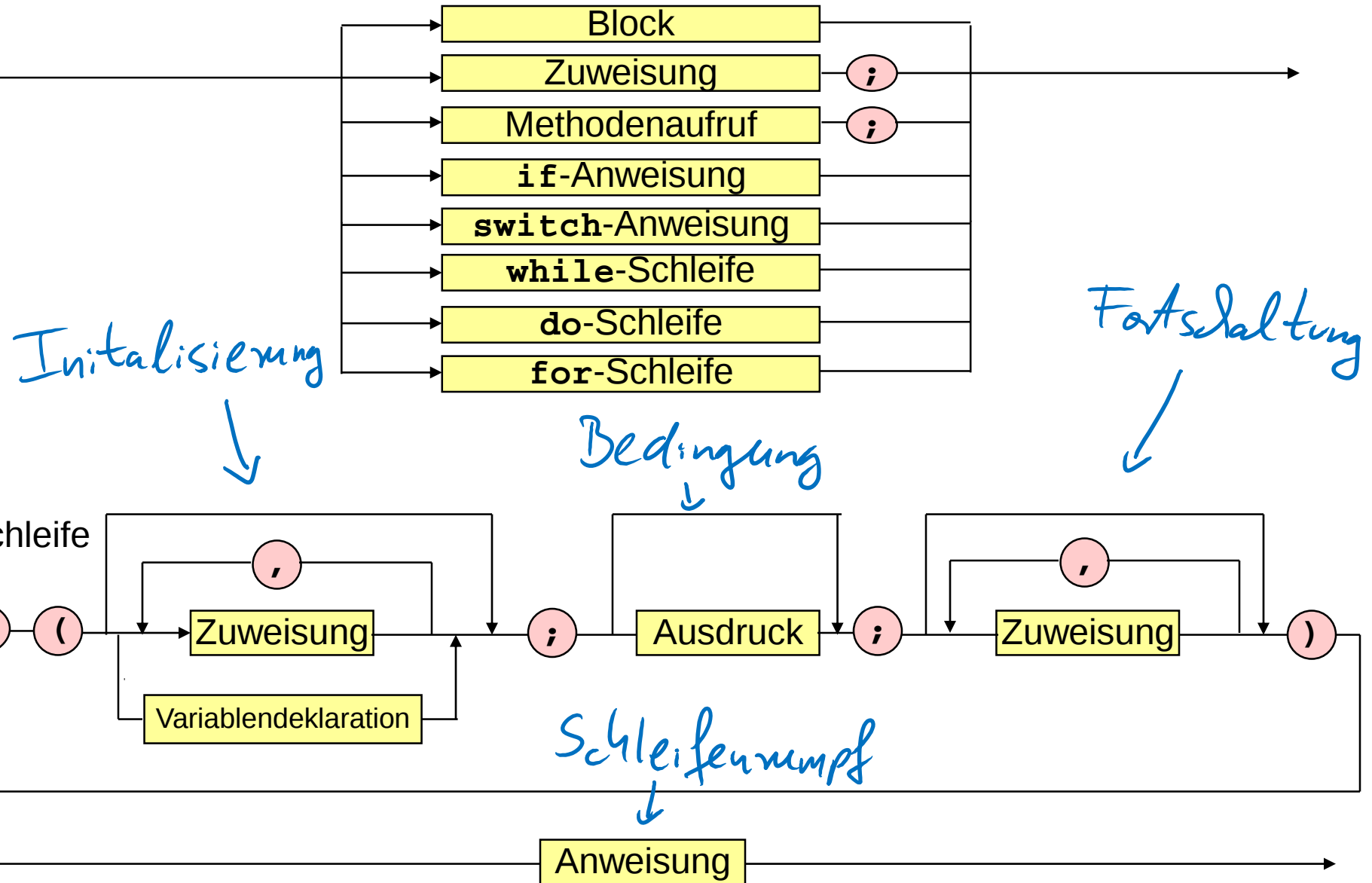
Schleifen



Methode mit do-Schleife

```
public class Wurzel {  
    public static void main (String [] args) {  
        float x = SimpleIO.getFloat ("Gib eine Zahl ein"),  
            uG = 0,  
            oG = x,  
            m,  
            epsilon = 1e-3f;  
  
        do { m = (uG + oG)/2;  
            if (m*m > x)    oG = m;  
            else           uG = m;  
        }  
        while (oG - uG > epsilon);  
  
        SimpleIO.output("Wurzel von " + x + " ist " + m,  
            "Ergebnis");  
    }  
}
```

Schleifen



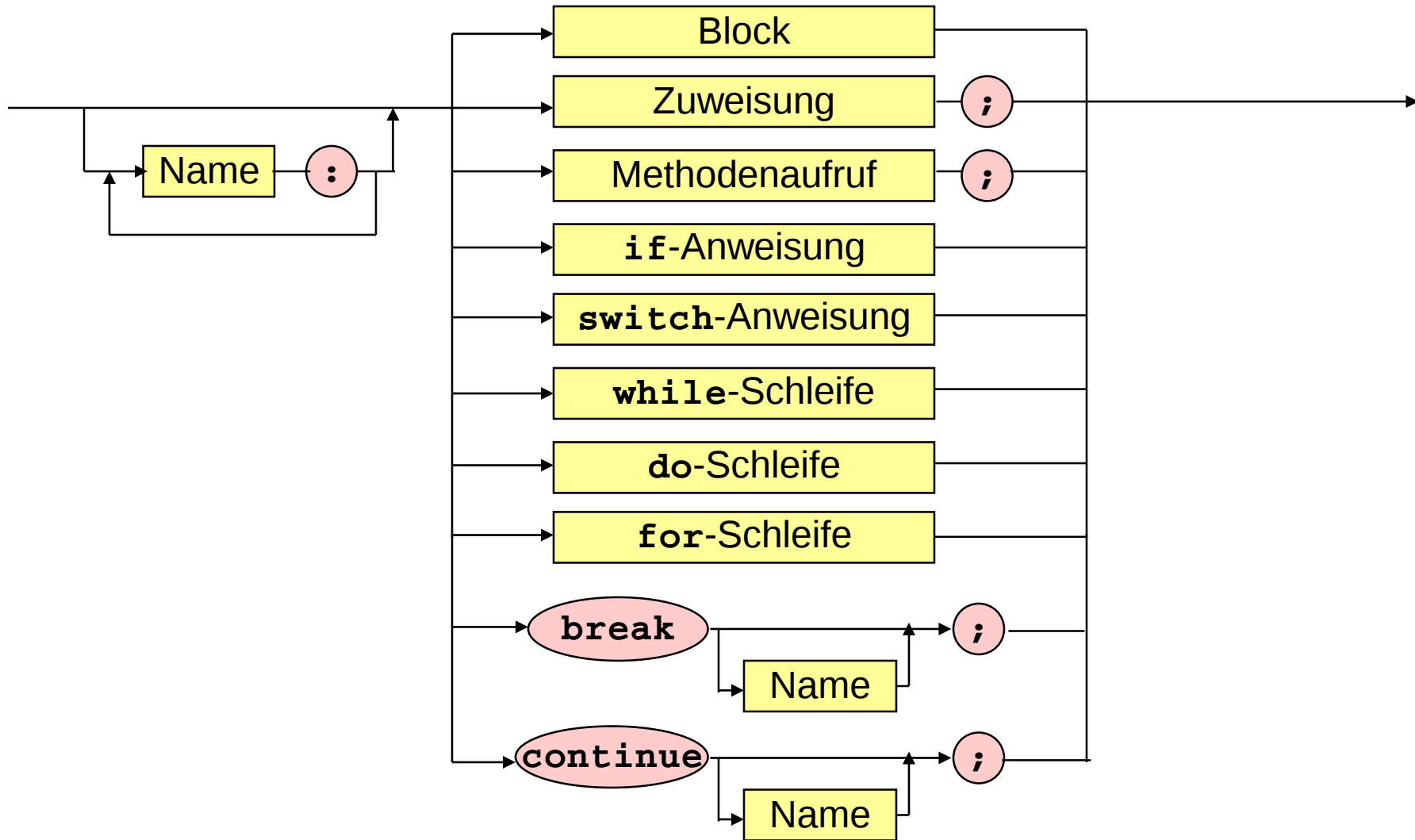
Methode mit for-Schleife

```
public class ForSchleife {  
  
    public static void main (String [] args) {  
  
        for (int i = 1; i <= 3; i++)  
  
            for (int j = 1; j <= i; j++)  
                System.out.println (i + " " + j + ", ");  
    }  
  
}
```

3. Anweisungen und Kontrollstrukturen

- Methodenaufruf
- Zuweisung
- Bedingte Anweisungen (`if`, `switch`)
- Schleifen (`while`, `do`, `for`)
- Sprunganweisungen (`break`, `continue`)

Sprunganweisungen



Methode mit Sprunganweisungen

```
public class Freitag {
    public static void main (String[] args) {

        int wochentag = SimpleIO.getInt ("Gib letzten Wochentag des Vorjahres ein");
        int schalttag = SimpleIO.getInt ("Gib 1 ein, falls Schaltjahr, sonst 0");

        monatsschleife: for (int monat = 1; monat <= 12; monat++) {

            tagesschleife: for (int tag = 1; tag <= 31; tag++) {

                //Wenn es tag nicht gibt, dann breche tagesschleife ab
                switch (monat) {
                    case 2 : if (tag > 28 + schalttag) continue monatsschleife;
                    case 4: case 6: case 9: case 11:
                        if (tag > 30) continue monatsschleife; }

                wochentag = wochentag % 7 + 1; // naechster wochentag

                if (tag != 13) continue tagesschleife; // Wenn tag kein 13., dann neuer tag
                // Wenn es ein Freitag ist, dann gib das Datum aus
                if (wochentag == 5) System.out.println ("Freitag, der 13. " + monat + ".");
                if (monat == 12) break monatsschleife; // Abbruch nach dem 13. 12.
            }
        }
    }
}
```

```
graph TD
    Start(( )) --> MainMain[main]
    MainMain --> Loop1[for monat]
    Loop1 --> Loop2[for tag]
    Loop2 --> Switch[switch]
    Switch --> If[if tag != 13]
    If --> Print[println]
    Print --> Break[break]
    Break --> EndLoop2[} tag loop]
    EndLoop2 --> EndLoop1[} monat loop]
    EndLoop1 --> EndMain[} main]
    EndMain --> End(( ))
```