
II. Imperative und objektorientierte Programmierung

- **1. Grundelemente der Programmierung**
- **2. Objekte, Klassen und Methoden**
- **3. Rekursion und dynamische Datenstrukturen**
- **4. Erweiterung von Klassen und fortgeschrittene Konzepte**

II.1. Grundelemente der Programmierung

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

1. Erste Schritte

- **Syntax von Methodendeklarationen**
- **Übersetzung und Ausführung von Java-Programmen**
- **Konstanten**
- **Eingabe und Methodenaufrufe**
- **Verkettung von Strings (& Polymorphismus)**
- **Syntax von Ausdrücken**

1. Erste Schritte

- **Syntax von Methodendeklarationen**
- Übersetzung und Ausführung von Java-Programmen
- Konstanten
- Eingabe und Methodenaufrufe
- Verkettung von Strings (& Polymorphismus)
- Syntax von Ausdrücken

Ein erstes Java-Programm

Ergebnstyp (kein Ergebnis)

Name

```
void main () {
```

Var-Dekl

```
→ int x, y;
```

An-
wei-
sun-
gen

```
x = 10;
```

```
y = -1 + 23 * 33 + 3 * 7 * (5 + 6);
```

```
IO.print ("Das Resultat ist ");
```

```
IO.println (x + y);
```

```
}
```

Zuwei-
sung

Methoden-
aufrufe

Methoden

Block

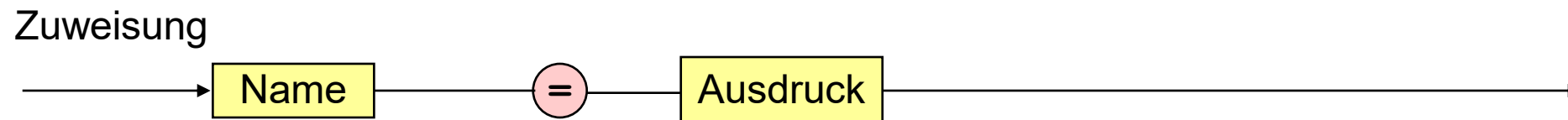
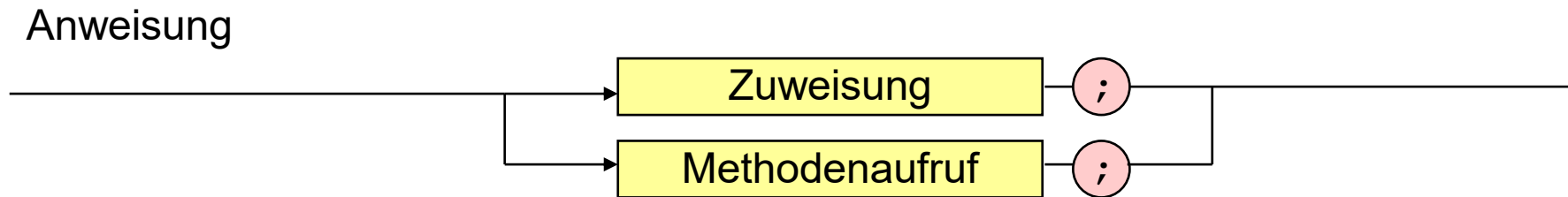
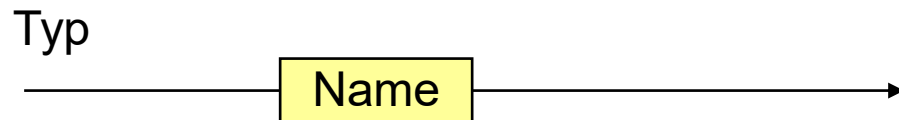
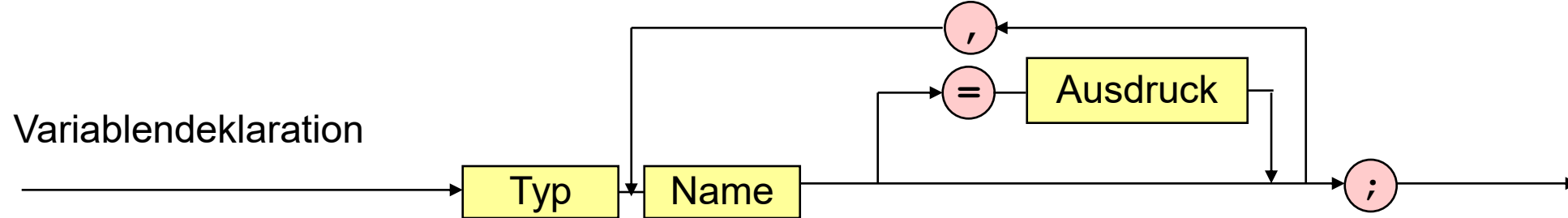
sagt, wo diese
Methode zu
finden ist

II.1.1 Erste Schritte - 7 -

The diagram illustrates the structure of a function signature node. It consists of a sequence of components connected by arrows: an arrow pointing to a pink oval containing the text 'void', followed by a yellow rectangle containing 'Name', then a pink circle containing '(', then an arrow pointing to a pink circle containing ')', followed by a yellow rectangle containing 'Block', and finally an arrow pointing to the right.

[illegible]

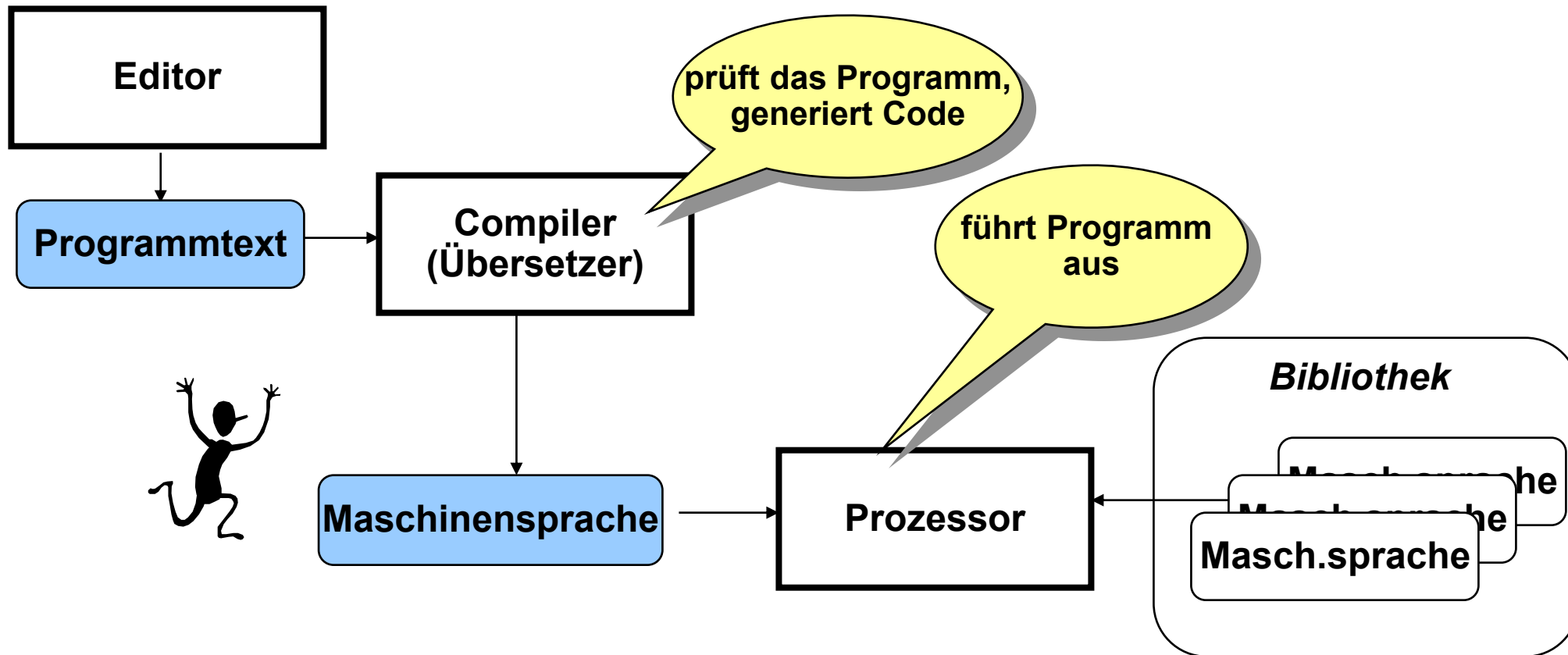
Variablendeklaration, Anweisung



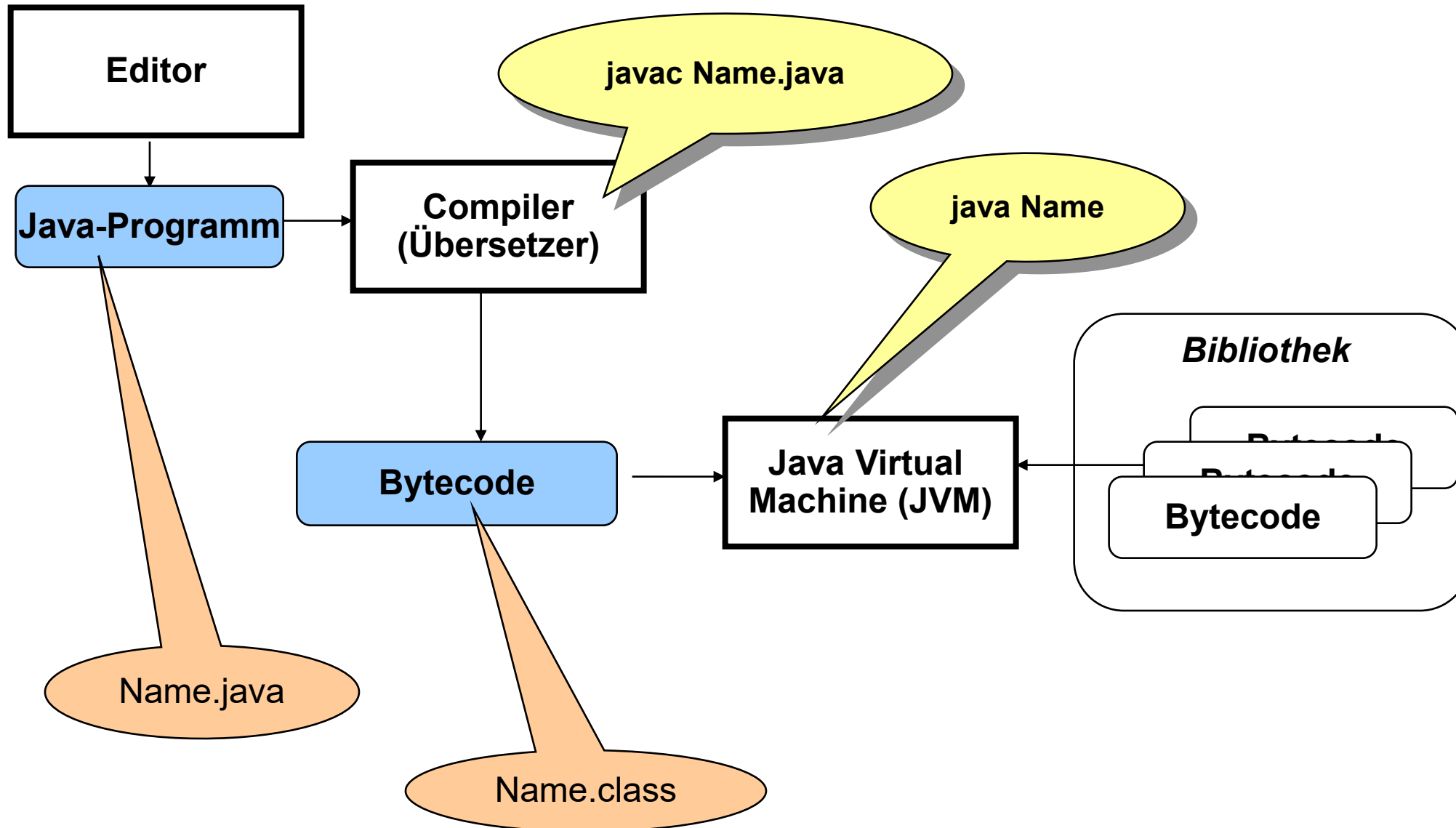
1. Erste Schritte

- Syntax von Methodendeklarationen
- Übersetzung und Ausführung von Java-Programmen
- Konstanten
- Eingabe und Methodenaufrufe
- Verkettung von Strings (& Polymorphismus)
- Syntax von Ausdrücken

Vom Programmtext zum ausf. Programm



Ausführen von Java-Programmen (JDK)



1. Erste Schritte

- Syntax von Methodendeklarationen
- Übersetzung und Ausführung von Java-Programmen
- **Konstanten**
- Eingabe und Methodenaufrufe
- Verkettung von Strings (& Polymorphismus)
- Syntax von Ausdrücken

Konstanten

```
void main () {  
  
    int x = 10;  
    int y = -1 + 23 * 33 + 3 * 7 * (5 + 6);  
  
    IO.print ("Das Resultat ist ");  
    IO.println (x + y);  
  
}
```

Konstanten

```
void main () {
```

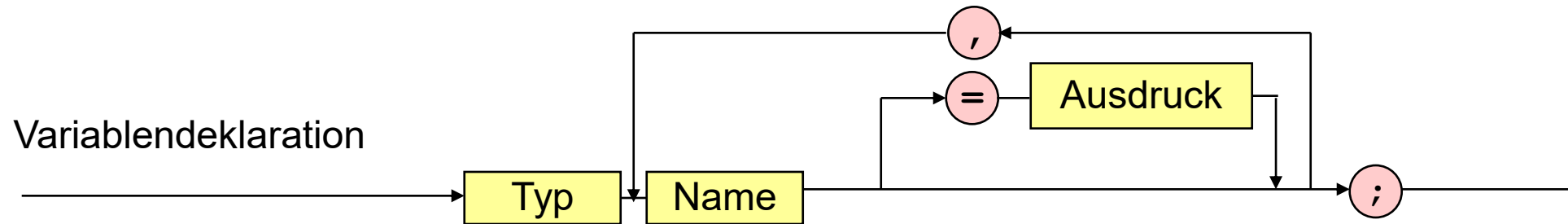
```
    final int x = 10, y = -1 + 23 * 33 + 3 * 7 * (5 + 6);
```

```
    IO.print ("Das Resultat ist ");
```

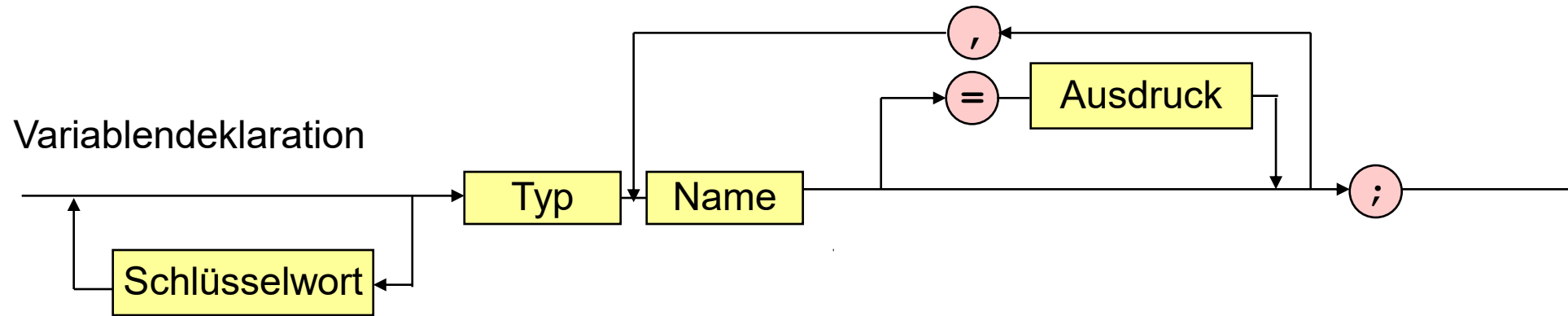
```
    IO.println (x + y);
```

```
}
```

Variablendeklaration, Anweisung



Variablendeklaration, Anweisung



Schlüsselwort: **final**, ...

1. Erste Schritte

- Syntax von Methodendeklarationen
- Übersetzung und Ausführung von Java-Programmen
- Konstanten
- **Eingabe und Methodenaufrufe**
- Verkettung von Strings (& Polymorphismus)
- Syntax von Ausdrücken

Eingabe und Methodenaufrufe

```
void main () {
```

```
/*  Verwende vordefinierte Klasse IO  
    zur Aus- und Eingabe von Werten  
*/
```

```
String x = IO.readLine("Gib ein Wort ein: ");
```

```
// Jetzt hat x den eingegebenen Wert.
```

```
IO.print ("Das eingegebene Wort ist: ");
```

```
IO.println (x);
```

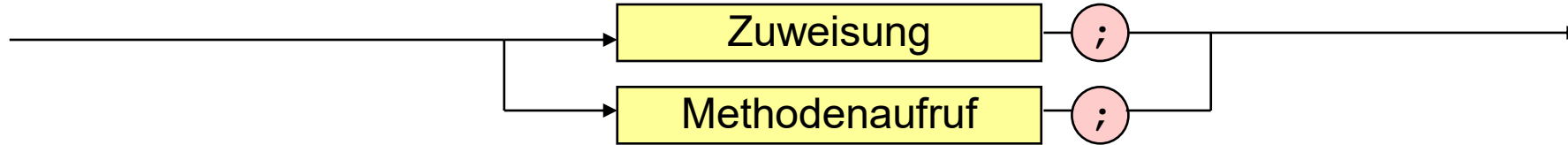
```
}
```

Eingabe und Methodenaufrufe

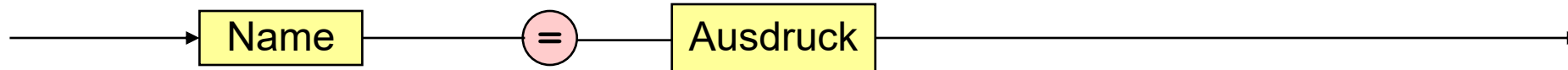
```
void main () {  
  
    String xString = IO.readLine("Bitte erste Zahl eingeben: ");  
    String yString = IO.readLine("Bitte zweite Zahl eingeben: ");  
  
    int x = Integer.parseInt(xString);  
    int y = Integer.parseInt(yString);  
  
    int maximum = Math.max(x,y);  
  
    IO.print    ("Das Maximum ist ");  
    IO.println (maximum);  
  
}
```

Eingabe und Methodenaufrufe

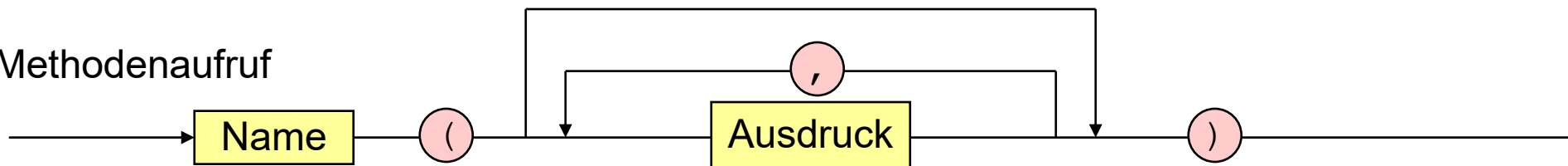
Anweisung



Zuweisung



Methodenaufruf



1. Erste Schritte

- Syntax von Methodendeklarationen
- Übersetzung und Ausführung von Java-Programmen
- Konstanten
- Eingabe und Methodenaufrufe
- **Verkettung von Strings (& Polymorphismus)**
- Syntax von Ausdrücken

Verkettung von Strings

```
void main () {  
  
    int y = -1 + 23 * 33 + 3 * 7 * (5 + 6);  
  
    String xString = IO.readLine("Gib eine Zahl ein: ");  
    int x = Integer.parseInt(xString);  
  
    IO.println ("Das Resultat ist " + (x + y));  
  
}
```

1. Erste Schritte

- Syntax von Methodendeklarationen
- Übersetzung und Ausführung von Java-Programmen
- Konstanten
- Eingabe und Methodenaufrufe
- Verkettung von Strings (& Polymorphismus)
- **Syntax von Ausdrücken**

Bedingter Ausdruck

```
void main () {  
  
    String xString = IO.readLine("Bitte eine Zahl eingeben: ");  
    int x = Integer.parseInt(xString);  
  
    int betrag;  
    betrag = x >= 0 ? x : -x;  
  
    IO.println("Betrag ist " + betrag);  
  
}
```

Ausdruck

