

---

# **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

---

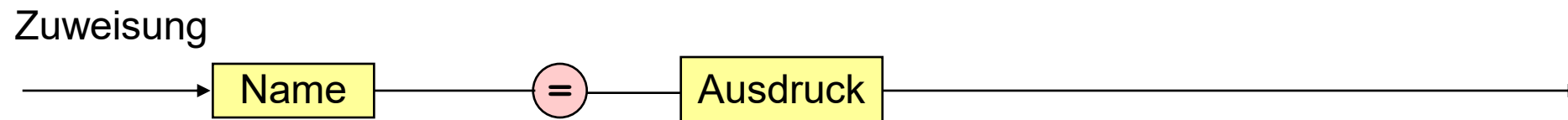
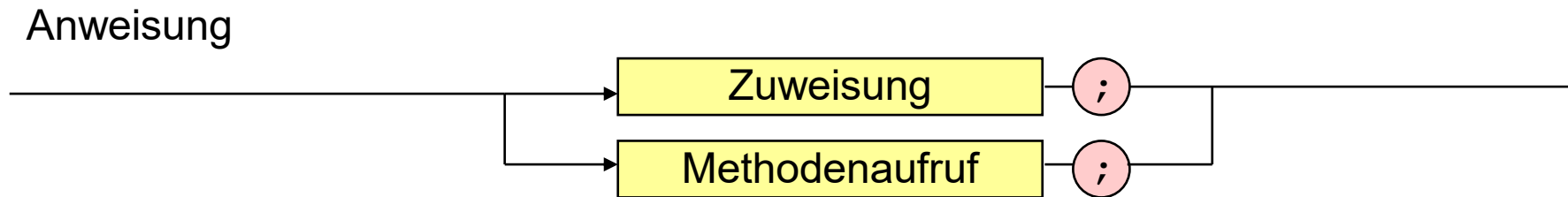
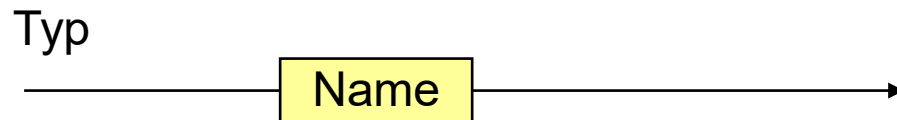
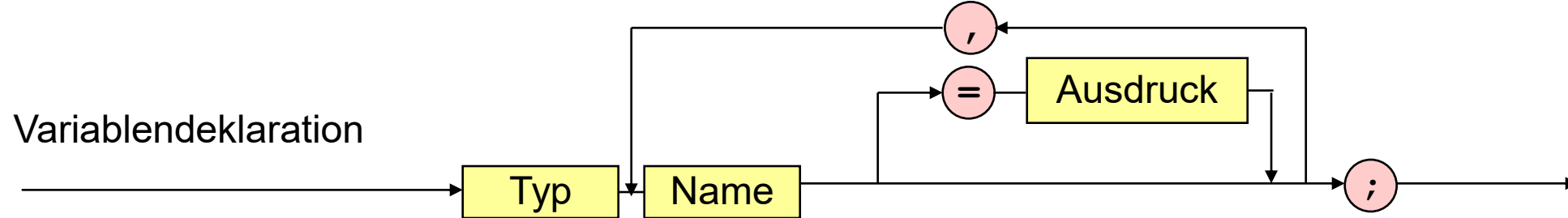
```
void main () {  
  
    int x, y;  
    x = 10;  
    y = -1 + 23 * 33 + 3 * 7 * (5 + 6);  
    IO.print ("Das Resultat ist ");  
    IO.println (x + y);  
  
}
```

II.1.1 Erste Schritte - 7 -

The diagram illustrates the structure of a function declaration node. It consists of a sequence of components connected by arrows: an arrow pointing to a pink oval labeled 'void', followed by a yellow rectangle labeled 'Name', then a pink circle labeled '(', then an arrow pointing to a pink circle labeled ')', and finally a yellow rectangle labeled 'Block'. An arrow points out from the right side of the 'Block' rectangle.

The diagram illustrates the structure of a block in a programming language. It features a horizontal line representing the flow of execution. On the left, the word "Block" is written. The line enters a pair of curly braces, represented by two pink circles. Inside the braces, there are two yellow rectangular boxes stacked vertically. The top box is labeled "Variablendeklaration" (Variable Declaration) and the bottom box is labeled "Anweisung" (Statement). Arrows indicate the flow: from the left into the opening brace, then down into the "Variablendeklaration" box, then down into the "Anweisung" box, and finally up into the closing brace. From the closing brace, the line continues to the right.

# Variablendeklaration, Anweisung



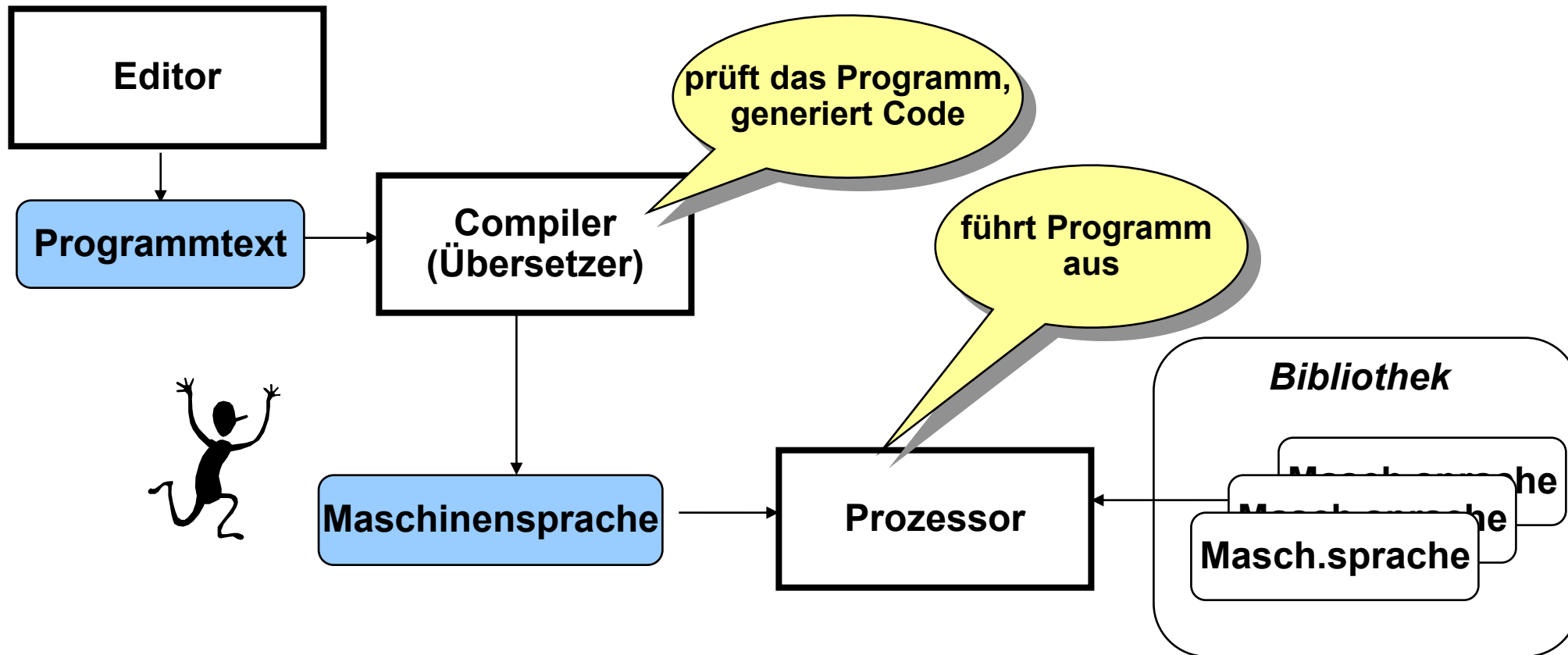
# 1. Erste Schritte

---

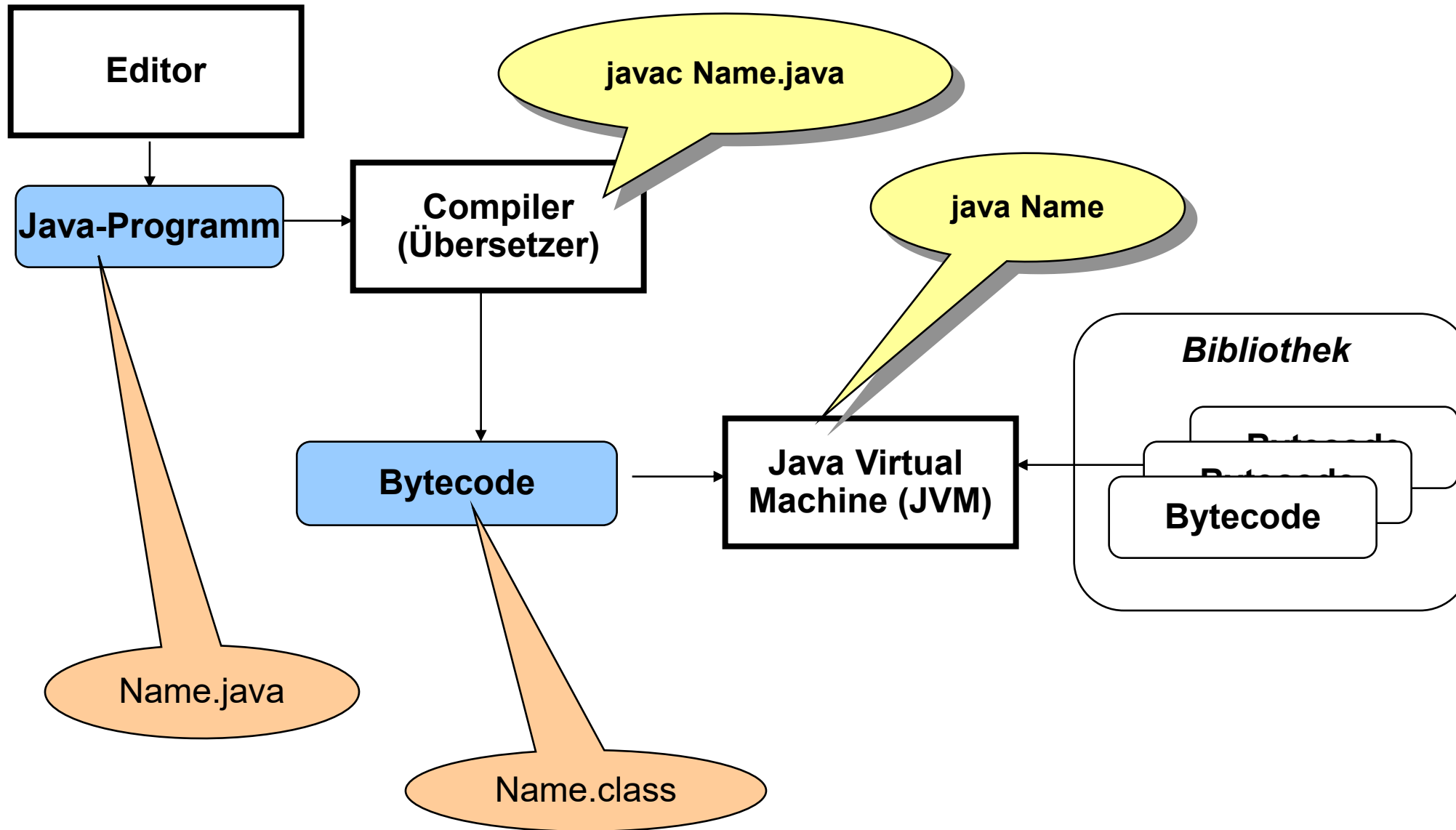
- Syntax von Methodendeklarationen
- Übersetzung und Ausführung von Java-Programmen
- Konstanten
- Eingabe und Methodenaufrufe
- Verkettung von Strings (& Polymorphismus)
- Syntax von Ausdrücken



# 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 () {
```

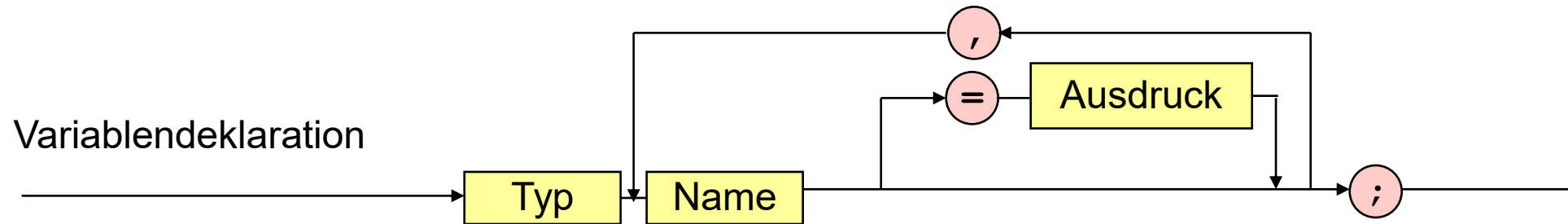
```
    final int x = 10, y = -1 + 23 * 33 + 3 * 7 * (5 + 6);
```

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

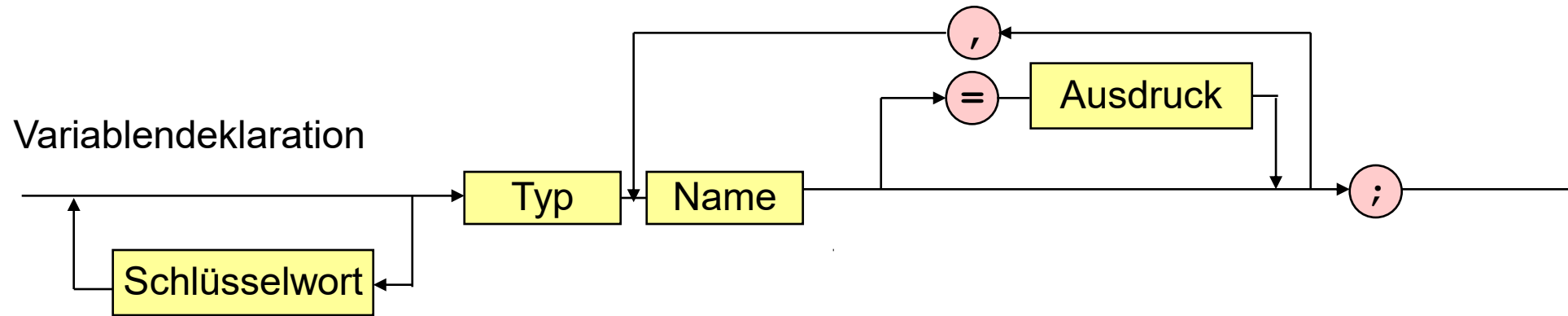
```
    IO.println (x + y);
```

```
}
```

# 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);
```

```
}
```

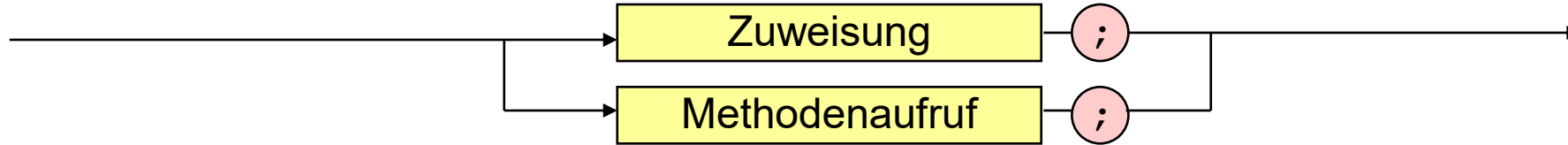
# Eingabe und Methodenaufrufe

---

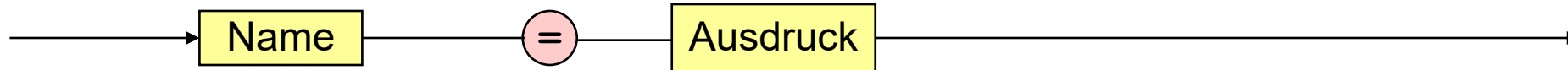
```
void main () {  
  
    String xString = IO.readLine("Bitte erste Zahl eingeben: ");  
    String yString = IO.readLine("Bitte zweite Zahl eingeben: ");  
  
    int x = Integer.parseInt(xString);  
    int y = Integer.parseInt(yString);  
  
    int maximum = Math.max(x,y);  
  
    IO.print   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

# Eingabe und Methodenaufrufe

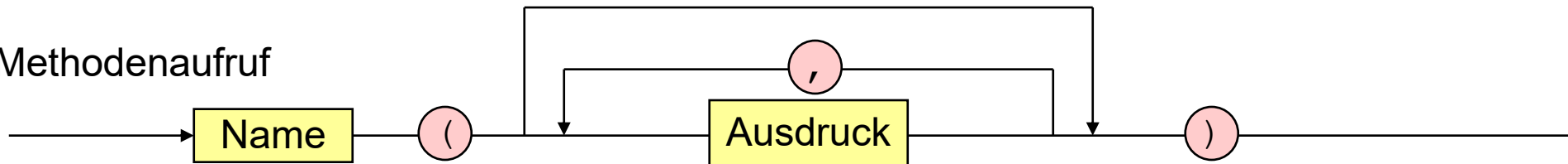
Anweisung



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

