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

1. Erste Schritte

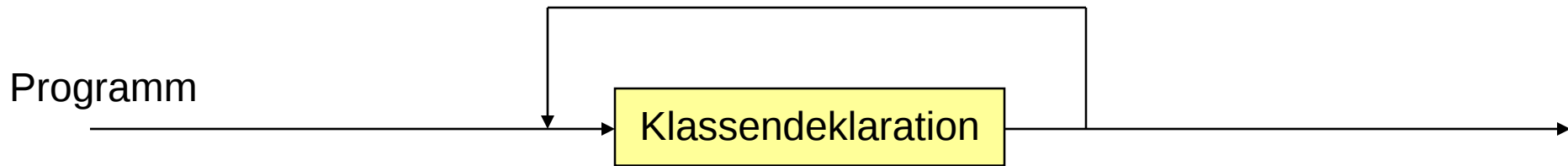
- **Syntax von Klassen- und Methodendeklarationen**
- Übersetzung und Ausführung von Java-Programmen
- Konstanten
- Eingabe und Methodenaufrufe
- Verkettung von Strings (& Polymorphismus)
- Syntax von Ausdrücken

Ein erstes Java-Programm

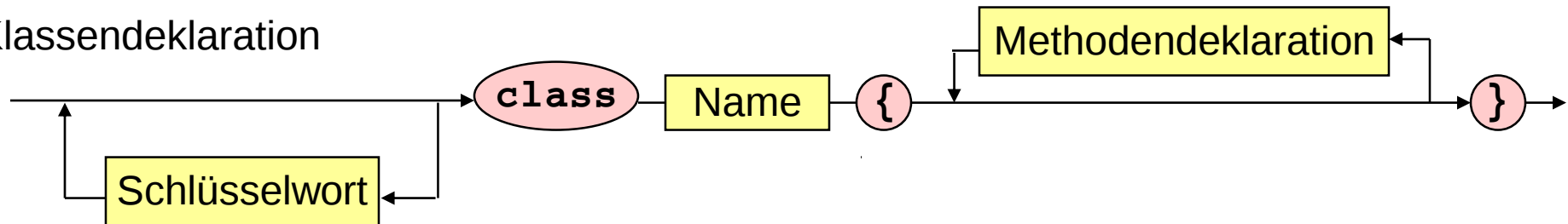
```
public class Rechnung {  
    public static void main (String [] arguments) {  
        int x, y;  
        x = 10;  
        y = -1 + 23 * 33 + 3 * 7 * (5 + 6);  
        System.out.print ("Das Resultat ist ");  
        System.out.println (x + y);  
    }  
}
```

Schlüsselwort → `public`
definiert eine Klasse → `class`
Name der Klasse → `Rechnung`
Klassenkopf → `public class Rechnung {`
Ergebnistyp (kein Ergebnis) → `void`
Name der Methode → `main`
Name des Arguments → `arguments`
Typ des Arguments → `String []`
Methodenaufrufe → `System.out.print` und `System.out.println`
Methodendeklaration → `public static void main (String [] arguments) {`
Methoden → `print` und `println`
Var-dekl → `int x, y;`
Anweisungen → `x = 10;`, `y = -1 + 23 * 33 + 3 * 7 * (5 + 6);`, `System.out.print`, `System.out.println`
zweisungen → `x = 10;`, `y = -1 + 23 * 33 + 3 * 7 * (5 + 6);`
sagt, wo die Methode zu finden ist → `}`
Klassenrumpf → `{ ... }`
Methoden (println springt nach Ausgabe in nächste Zeile) → `println`

Programm und Klassendeklaration



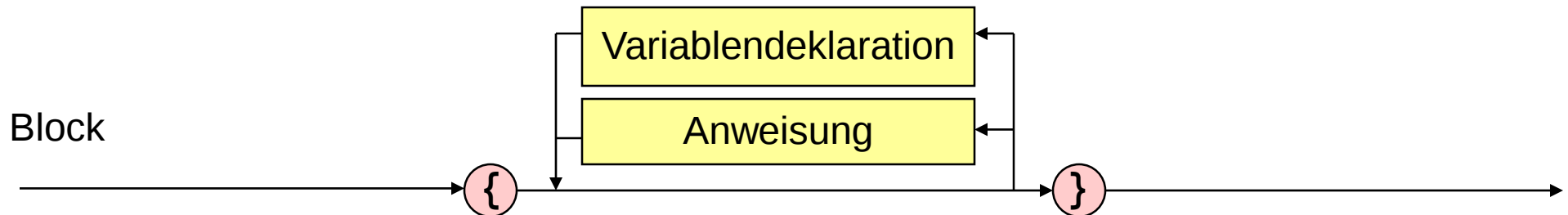
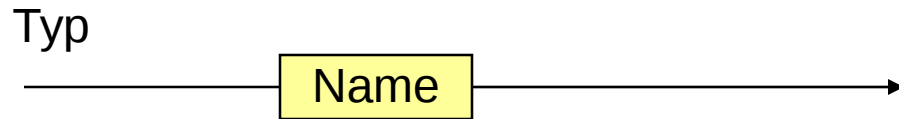
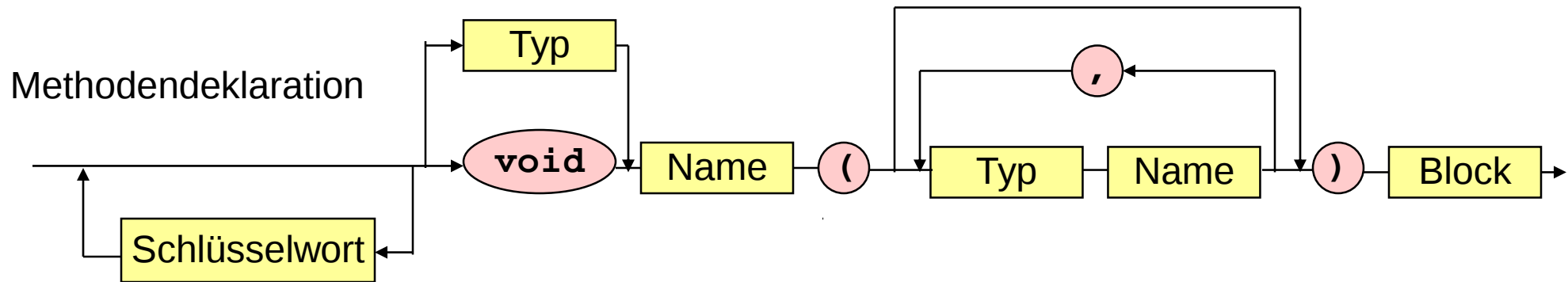
Klassendeklaration



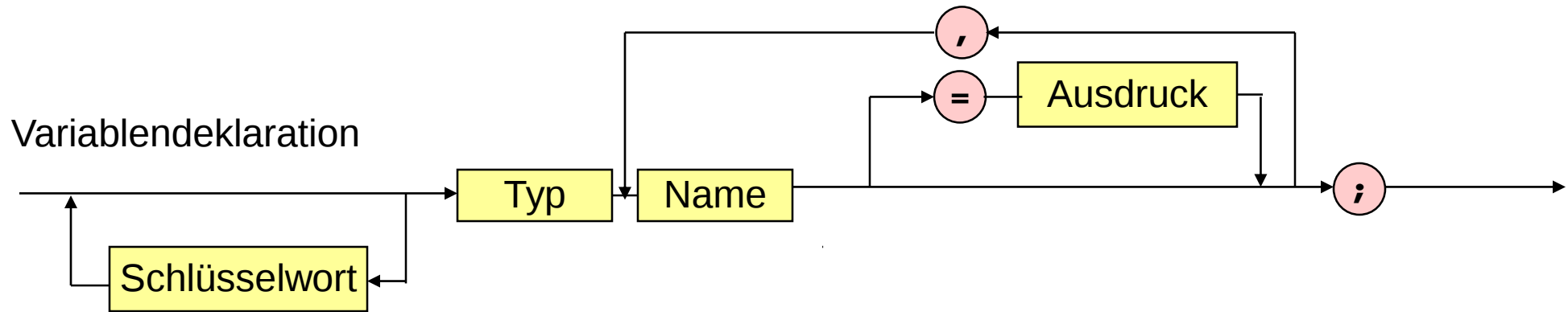
Name: Zeichenreihe

Schlüsselwort: `public, static, final, ...`

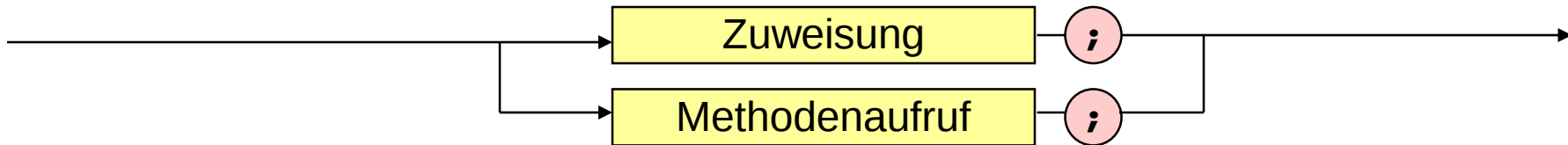
Methodendeklaration, Typ, Block



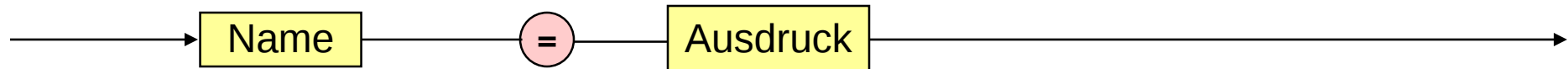
Variablendeklaration, Anweisung



Anweisung



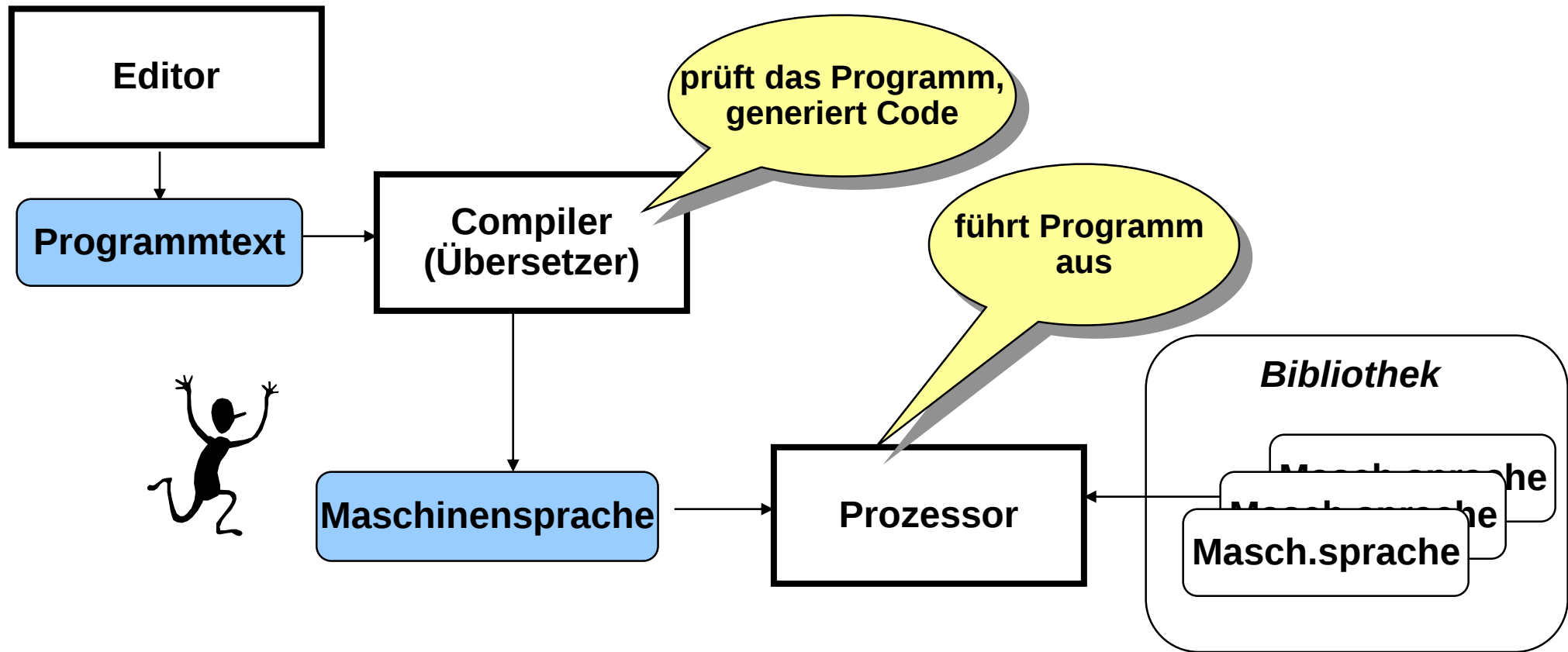
Zuweisung



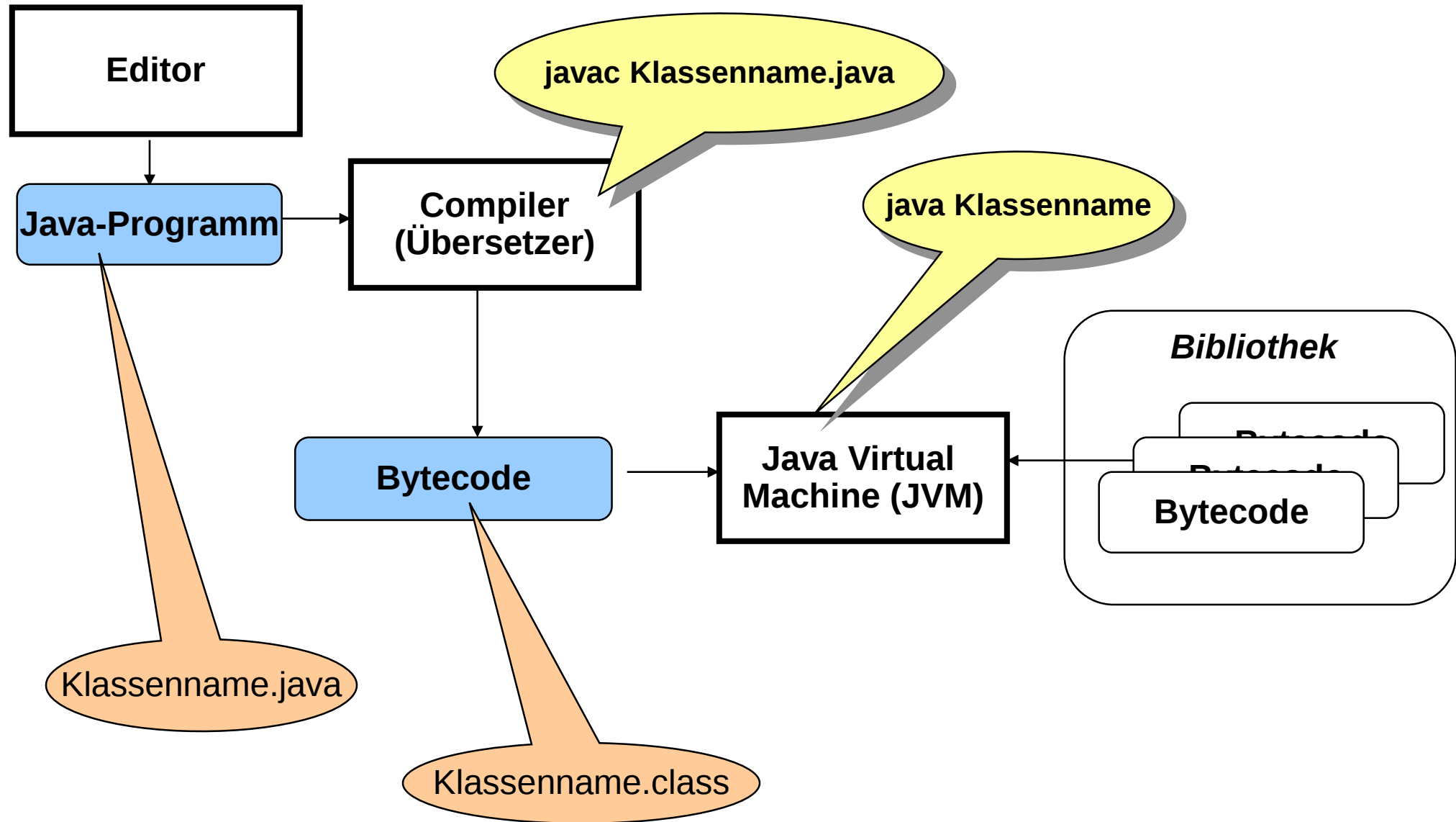
1. Erste Schritte

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

Konstanten

```
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);  
  
    }  
  
}
```

Konstanten

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

1. Erste Schritte

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

Eingabe und Methodenaufrufe

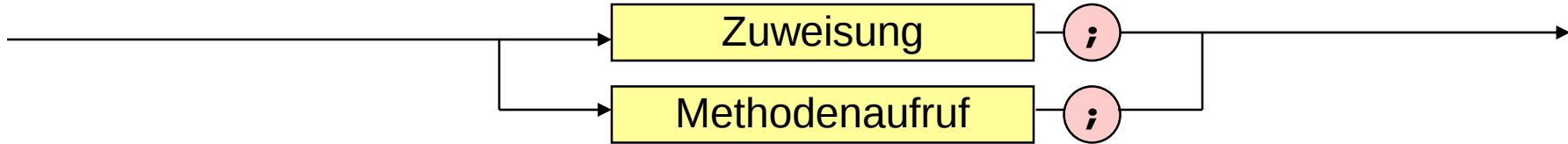
```
public class Rechnung {  
  
    public static void main (String [] arguments) {  
        int y = -1 + 23 * 33 + 3 * 7 * (5 + 6);  
  
        /* Verwende (von uns) vordefinierte Klasse SimpleIO  
           zur Eingabe von Werten  
           */  
  
        int x = SimpleIO.getInt("Gib eine Zahl ein");  
  
        // Jetzt hat x den eingegebenen Wert.  
  
        System.out.print ("Das Resultat ist ");  
        System.out.println (x + y);  
    }  
}
```

Eingabe und Methodenaufrufe

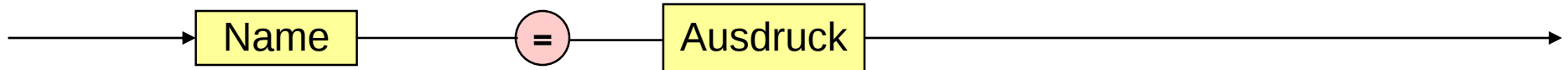
```
public class Maximum {  
  
    public static void main (String [] arguments) {  
  
        int x = SimpleIO.getInt("Bitte erste Zahl eingeben");  
        int y = SimpleIO.getInt("Bitte zweite Zahl eingeben");  
  
        int maximum = Math.max(x,y);  
  
        System.out.print ("Das Maximum ist ");  
        System.out.println (maximum);  
    }  
}
```

Eingabe und Methodenaufrufe

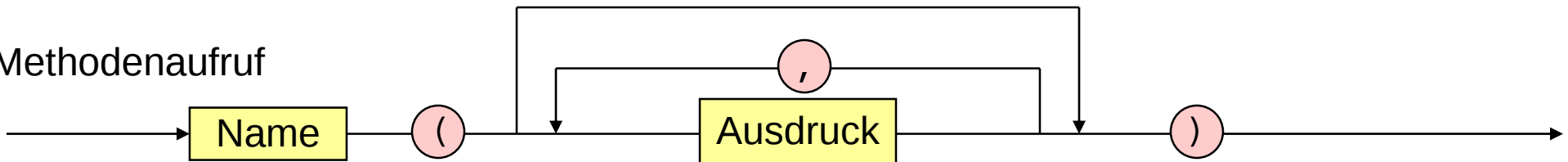
Anweisung



Zuweisung



Methodenaufruf



1. Erste Schritte

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

Verkettung von Strings

```
public class Rechnung {  
    public static void main (String [] arguments) {  
        int y = -1 + 23 * 33 + 3 * 7 * (5 + 6);  
        int x = SimpleIO.getInt("Gib eine Zahl ein");  
  
        SimpleIO.output("Das Resultat ist " + (x + y), "Ergebnis");  
    }  
}
```

1. Erste Schritte

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

Bedingter Ausdruck

```
public class Betrag {  
  
    public static void main (String [] arguments) {  
  
        int x = SimpleIO.getInt("Bitte eine Zahl eingeben");  
  
        int betrag;  
        betrag = x >= 0 ? x : -x;  
  
        SimpleIO.output("Betrag ist " + betrag, "Ergebnis");  
  
    }  
  
}
```

Ausdruck

