

---

# **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**
- **Verkettung von Strings (& Polymorphismus)**
- **Syntax von Methodenaufrufen**
- **Syntax von Ausdrücken**

# 1. Erste Schritte

---

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

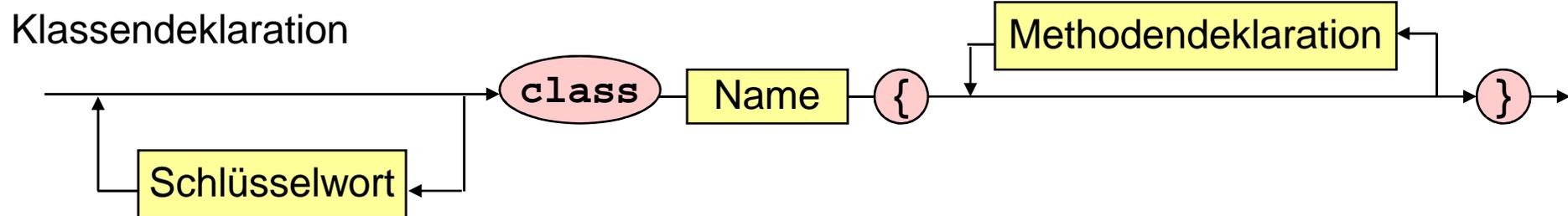
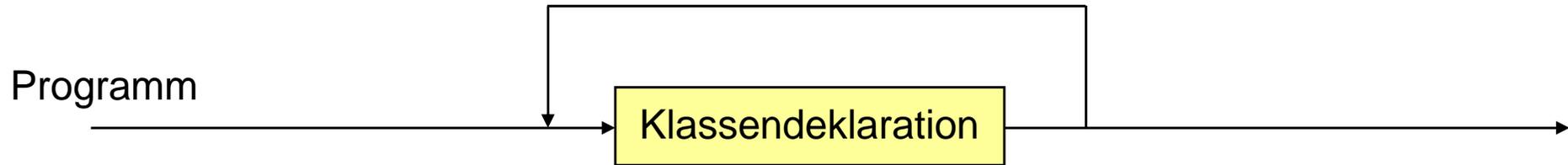
# Ein erstes Java-Programm

---

```
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);  
  
    }  
  
}
```

# Programm und Klassendeklaration

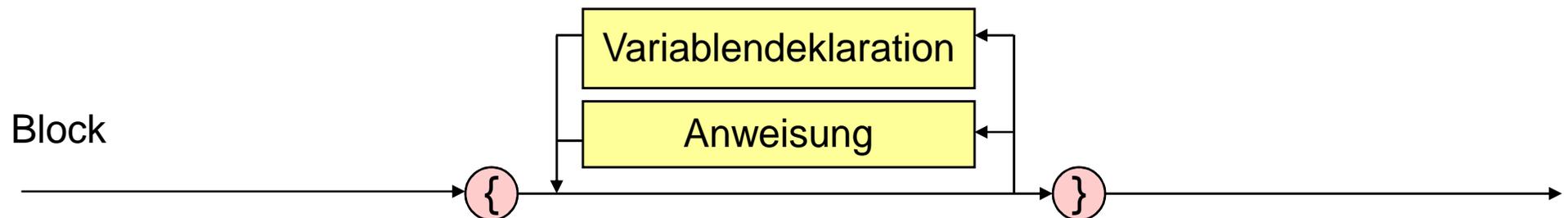
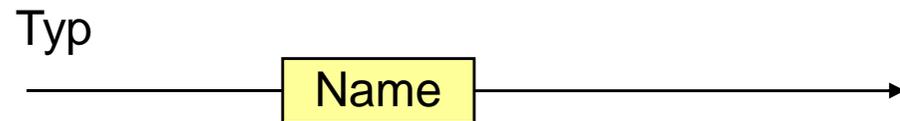
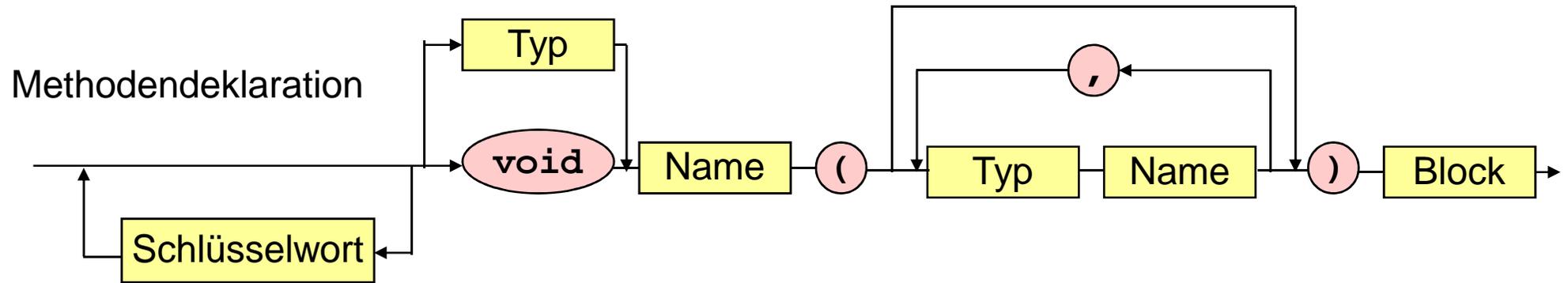
---



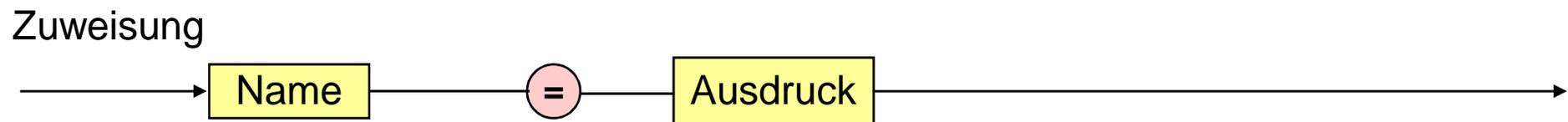
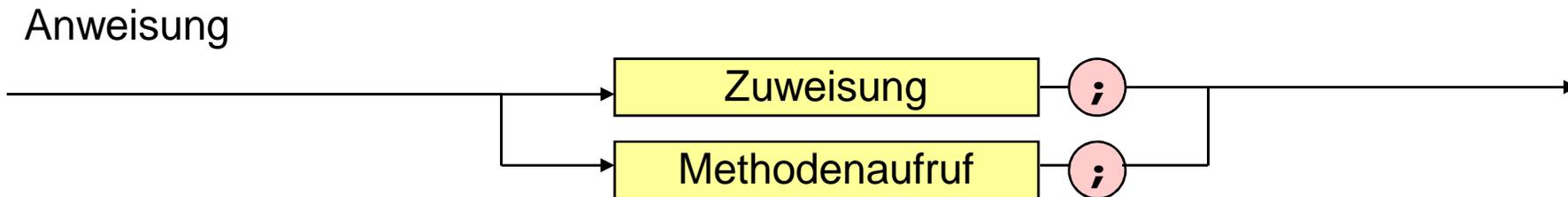
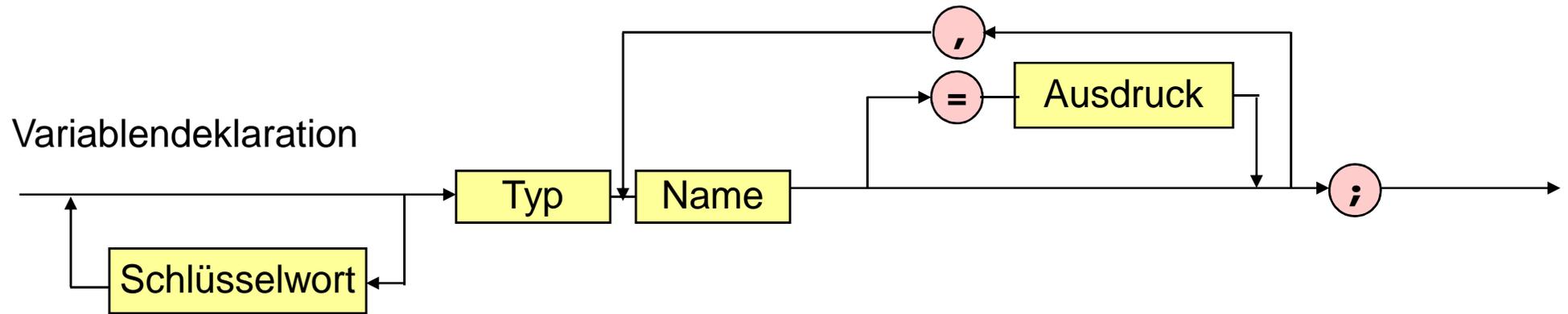
Name: Zeichenreihe

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

# Methodendeklaration, Typ, Block



# Variablendeklaration, Anweisung



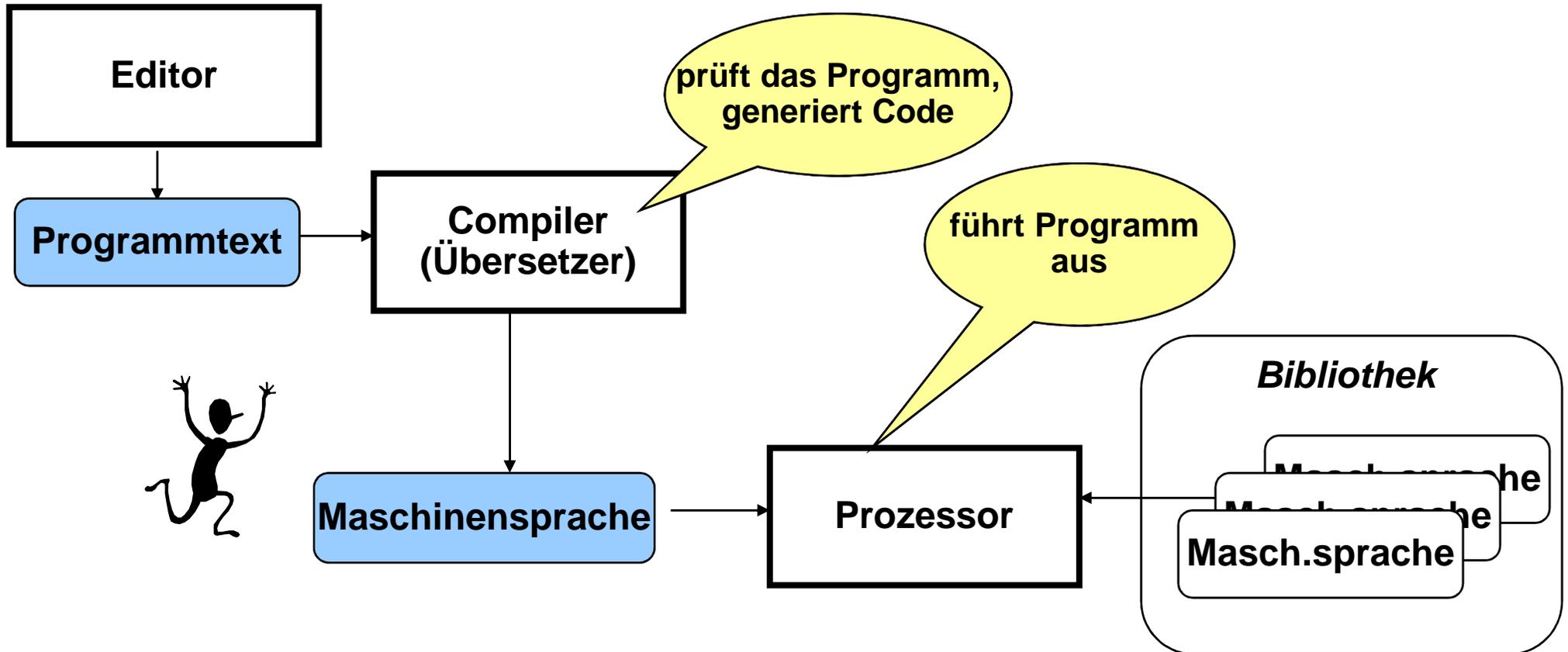
# 1. Erste Schritte

---

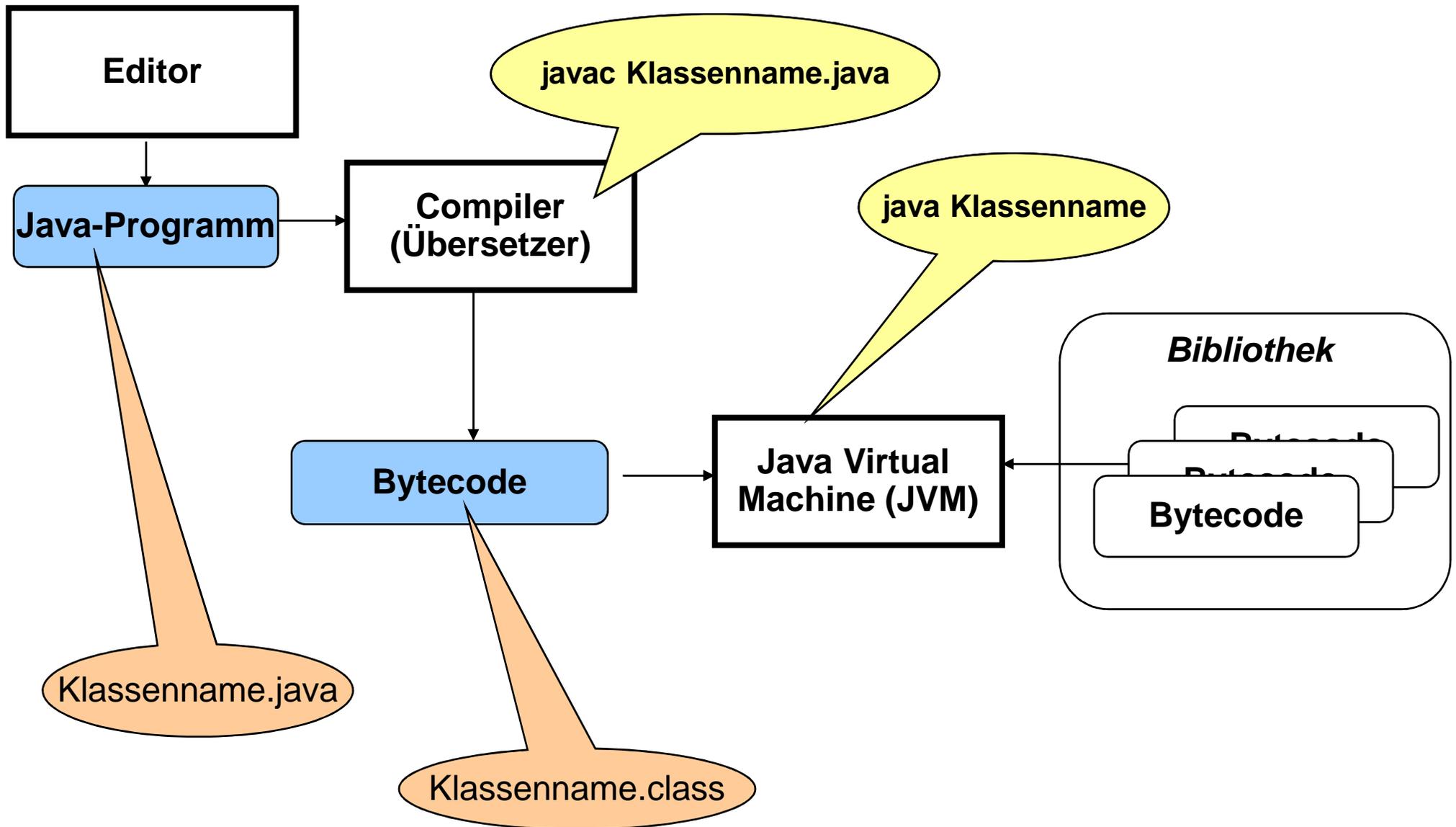
- Syntax von Klassen- und Methodendeklarationen
- **Übersetzung und Ausführung von Java-Programmen**
- Konstanten
- Eingabe
- Verkettung von Strings (& Polymorphismus)
- Syntax von Methodenaufrufen
- Syntax von Ausdrücken

# 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
- Verkettung von Strings (& Polymorphismus)
- Syntax von Methodenaufrufen
- 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**
- Verkettung von Strings (& Polymorphismus)
- Syntax von Methodenaufrufen
- Syntax von Ausdrücken

# Eingabe

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

# Eingabe

```
public class Rechnung {  
  
    public static void main (String [] arguments) {  
        int y = -1 + 23 * 33 + 3 * 7 * (5 + 6);  
  
        System.out.print("Bitte Zahl eingeben: ");  
  
        // Lies einen String ein  
        String s = System.console().readLine();  
  
        // Konvertiere String in ganze Zahl  
        int x = Integer.parseInt(s);  
  
        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
- **Verkettung von Strings (& Polymorphismus)**
- Syntax von Methodenaufrufen
- Syntax von Ausdrücken

# Verkettung von Strings

---

```
public class Rechnung {  
  
    public static void main (String [] arguments) {  
  
        int y = -1 + 23 * 33 + 3 * 7 * (5 + 6);  
  
        System.out.print("Bitte Zahl eingeben: ");  
        String s = System.console().readLine();  
        int x = Integer.parseInt(s);  
  
        System.out.print ("Das Resultat ist ");  
        System.out.println (x + y);  
  
    }  
}
```

# Verkettung von Strings

---

```
public class Rechnung {  
  
    public static void main (String [] arguments) {  
  
        int y = -1 + 23 * 33 + 3 * 7 * (5 + 6);  
  
        System.out.print("Bitte Zahl eingeben: ");  
        String s = System.console().readLine();  
        int x = Integer.parseInt(s);  
  
        System.out.println ("Das Resultat ist " + (x + y));  
  
    }  
}
```

# 1. Erste Schritte

---

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

# Methodenaufruf

---

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



# 1. Erste Schritte

---

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

# Bedingter Ausdruck

---

```
public class Betrag {  
  
    public static void main (String [] arguments) {  
  
        System.out.print("Bitte eine Zahl eingeben: ");  
        int x = Integer.parseInt(System.console().readLine());  
  
        int betrag;  
        betrag = x >= 0 ? x : -x;  
  
        System.out.println ("Der Betrag ist " + betrag);  
    }  
}
```

# Ausdruck

