Inhalt

1. Programmierung
2. Programmiersprache Java
3. Entwicklungsumgebung
4. Programmaufbau
5. Konsole

Programmierung

Was ist ein Algorithmus?

- Vorlage, um in einer endlichen Zeit aufgrund einer Eingabe die gewünschte Ausgabe zu bekommen
- muss terminieren (anahlten)
- Algorithmus: `Zur Flächenberechnung eines Rechtecks müssen die kürzere Seite und die längere Seite miteinander multipliziert werden.`

Was ist ein Programm?

- konkrete Formulierung/Umsetzung eines Alogrithmus
- Algorithmus: `Zur Flächenberechnung eines Rechtecks müssen die kürzere Seite und die längere Seite miteinander multipliziert werden.`
- Programm (nicht Java, nur **Pseudocode**)

```
shorterSide = 10
longerSide = 40
area = shorterSide * longerSide
```

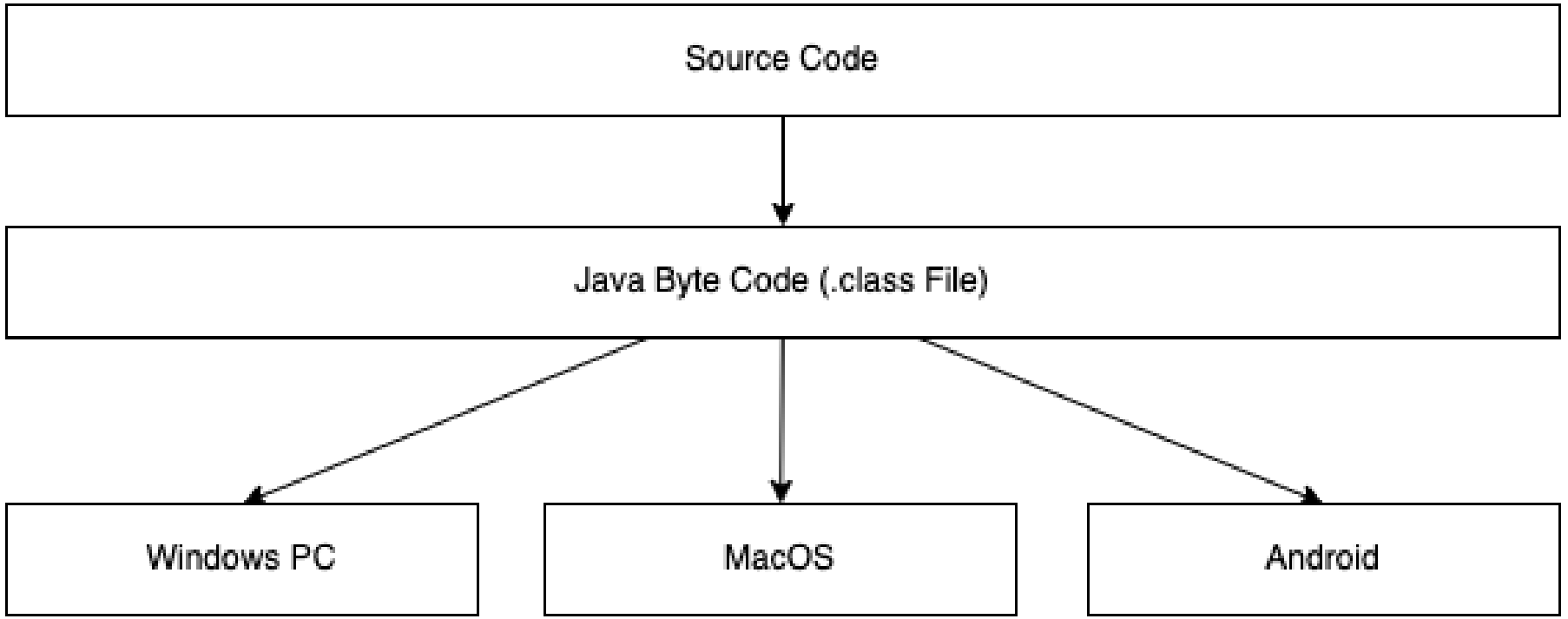
Übersicht Programmiersprachen

1. Maschinencode `0001001`
2. Assemblercode `mv %rx $6`
3. Höhere Programmiersprachen
    i. Interpretiert
        - PHP, Ruby, Python und JavaScript
    ii. Kompiliert
        - Java, C#, C, C++

Java

- Entwickelt von Sun. Jetzt **Oracle**
- Weit verbreitet (Server, Android, Desktop, ...)
- Imperativ und Objektorientiert
- Kompilierte Sprache

Java


Source Code
Java Byte Code (.class File)
Windows PC
MacOS
Android

Wichtige Eigenschaften von Java

- Leerzeichen zwischen Wörtern sind egal
- Statements werden mit `;` beendet
- Groß-/Kleinschreibung ist wichtig
  - `KONSTANTEN` WERDEN GROSS GESCHRIEBEN
  - `Klassen` werden in **CamelCase** geschrieben (1. Buchstabe groß)
  - `methoden` werden in **amelCase** geschrieben (1. Buchstabe klein)
  - `variablen` werden klein geschrieben

Programmieren ohne IDE

1. JDK (Java Development Kit) am Computer installieren
2. Programm in Texteditor schreiben (zB. Notepad)
3. in Konsole kompilieren: `"javac MeinProgramm.java"`
4. Programm in Konsole ausführen: `"java MeinProgramm"`

⇒ **Wir verwenden für diesen Prozess eine Entwicklungsumgebung**

Integrated Development Environment (IDE)

Zusammenfassung wichtiger Entwicklungs-Tools

Warum eine IDE? (Auszug)

- Editieren, Kompilieren
- Debugging
- Exportieren, Veröffentlichen, ...

Beispiele:

- BlueJ (nur zum Lernen)
- Eclipse
- Netbeans
- **IntelliJ** (wird im Kurs verwendet)

Programmaufbau: Konstrukte

- **Variablen** `int x = 10;`
  - Wert 10 wird in x gespeichert.
  - x kann überschrieben werden
- **Konstanten** `final int KONST_Y = 10;`
  - Schlüsselwort `final`
  - Wert von KONST_Y kann **nicht** verändert werden
- **Methoden**
  - Ausagern von Funktionsblöcken
  - Wiederverwendung
- **Klassen**
  - Überkonstrukt, beinhalten Methoden, Variablen, Konstanten, ...

Programmaufbau

```java
public class Main{

  public static void main(String[] args){
    // Einstiegspunkt ins Programm
    // Methodenrumpf
  }

}
```

- **Main**: Klasse
- **main**: Methode
- `// ...` : Kommentar => wird nicht ausgeführt

Programmaufbau

Main Methode

```
class Main{
  public static void main(String[] args){
    // Einstiegspunkt ins Programm
    // Methodenrumpf
  }
}
```

- Einstiegspunkt in Programm
- `String[] args` sind Programmargumente. Diese werden bei Programmstart mitgegeben
- Beispiel: Starten in Konsole von: `java main Java Intro`
    - Das Programm mit dem Namen `main` hat zwei Start-Parameter: `"Java"` und `"Intro"`

Klasse Main

- Bei uns ist der Einstiegspunkt zu Beginn **immer** in *Main.java*. Darum heißt die Klasse auch Main
- Klassenname soll mit Dateiname übereinstimmen

XYZ.java

```java
class XYZ{ // Klassenname kann anders gewählt werden
    public static void main(String[] args){
    // Methode muss main heißen
    // Hier können Befehle (STATEMENTS) stehen
  }
}
```

Ausgabe auf Konsole

- System-Library liefert verschiedene Hilfsmethoden
  - **Schreiben auf Konsole**
    - Schreiben mit Zeilenumbruch: **println**
      - `System.out.println("Hello, world!");`
    - Schreiben ohne Zeilenumbruch
      - `System.out.print("Hello, world!");`