

## 1 Klassifizierung

| Sprache     | funktional | objektorientiert | plattformunabhängig | Skript-Sprache |
|-------------|------------|------------------|---------------------|----------------|
| C           |            |                  |                     |                |
| C++         |            | X                |                     |                |
| <b>Java</b> |            | <b>X</b>         | <b>X</b>            |                |
| Python      |            | X                | X                   | X              |
| Ruby        |            | X                | X                   | X              |
| Haskell     | X          |                  |                     |                |

Tabelle 1: Einordnung von Java im Gegenzug zu anderen Programmiersprachen

In Tabelle 1 seht ihr eine Einordnung von Java im Verhältnis zu einigen, weiteren Programmiersprachen.

☞ Bitte Beachtet: Die Einordnung wurde Anhand ausgewählter Kriterien gemacht.  
Es gibt darüber hinaus noch viele andere Eigenschaften, welche hierfür verwendet werden könnten.

- **funktionale Sprachen:** Aufbau wie in der Mathematik (Keine Variablen/Nur Konstanten, keine Schleifen)
- **objektorientierte Sprachen:** Einführung neuer Datentypen mit bestimmten Eigenschaften (Bsp.: Objekt Buch hat Autor und Titel)
- **plattformunabhängige Sprachen:** Kann auf verschiedenen Betriebssystemen (Plattformen) ausgeführt werden
- **Skript-Sprachen:** Werden erst zur Laufzeit des Programmes in Maschinencode umgewandelt (Werden nicht kompiliert)

## 2 Plattformunabhängigkeit

Java realisiert die Plattformunabhängigkeit dadurch, dass der Quellcode (der geschriebene Programmcode) nicht direkt in Maschinencode (vom PC verstandene Anweisungen) übersetzt wird, sondern in eine Art "Zwischencode".

Die Übersetzung wird hierbei durch einen so genannten Compiler (im Fall von Java "javac") durchgeführt.

Dieser Zwischencode wird dann mit Hilfe einer virtuellen Maschine (JVM - Java Virtual Machine) während der Programmlaufzeit in Maschinencode übersetzt.

Diese JVM muss aus diesem Grund auf jedem System, welches das Programm ausführen soll, installiert sein!

Abb. 1 zeigt diesen Ablauf nochmals grafisch.

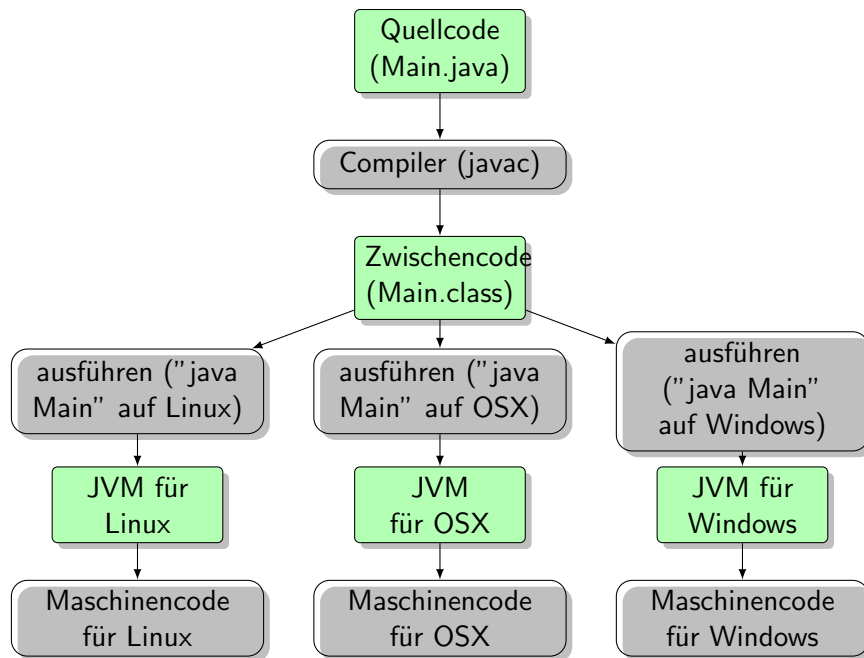


Abbildung 1: Plattformunabhängigkeit eines Java-Programms

### 3 Aufbau eines Java-Programmes

Listing 1 zeigt den Aufbau eines minimalen Java-Programms. Die Grün markierten Textbereiche sind hierbei Kommentare, welche der Compiler ignoriert und den Programmfluss nicht beeinflussen.

```
/**
 * Datei: Main.java
 * Ein Java-Programm besteht immer aus mindestens einer
 * Klasse
 * Diese wird durch das Schluesselwort "class" deklariert
 * und sie wird stetig gross geschrieben.
 * Darueber hinaus ist es wichtig, dass die Datei den
 * gleichen Namen wie die beinhaltende Klasse besitzt,
 * sowie die Endung ".java"
 * Der Inhalt einer Klasse ist umringt von geschweiften
 * Klammern
 */
class Main{
    /**
     * Jedes Java-Programm muss die folgende main-Methode
     * aufweisen. Diese wird bei Programmstart ausgefuehrt.
     */
    public static void main(String[] args){
        System.out.println("test");
        //hier stehen die auszufuehrenden Anweisungen
    }
}
```

Listing 1: Beispiel eines Java-Programms

### 4 Exkurs: Was sind Bits?

Ein Bit ist die kleinste Einheit mit welcher ein Computer rechnen kann. Ein Rechner ist eine Maschine und diese kennt an sich nur zwei Zustände (Strom oder kein Strom). In der Informatik wird dies oft durch die beiden Werte 1 (Strom) und 0 (kein Strom) ausgedrückt. Aus diesem Grund müssen innerhalb des Computers auch alle Werte als Bits "codiert" werden. Ist ein Wert 8-Bit lang, so kann dieser  $2^8$ , also 256, verschiedene Werte annehmen. Die  $2^8$  kommt daher, dass es sich um 8-Bits handelt, welche jeweils zwei verschiedene Zustände annehmen können.

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|

Abbildung 2: Beispiel eines 8-Bit langen Wertes

## 5 Die ersten Datentypen

```
/**
 * Datei: Variable.java
 * Dieses Java-Programm zeigt die grundlegende Verwendung
 * von Variablen.
 */
class Datentypen{
    public static void main(String[] args){
        int i;// definition der variable i vom Typ int.
        /*
         * Integer (int) sind Ganzzahlen. Ein int
         * belegt in Java 32-Bit und
         * kann deshalb  $2^{32}$  verschiedene Werte
         * annehmen.
         */

        i = 4; //Zuweisung/Initialisierung der Variable i
        /*
         * Initialisierung heisst es nur im bei der ersten
         * Zuweisung
         */

        short s;// definition der variable s, vom Typ short.
        /*
         * shorts sind 16-Bit Ganzzahlen.
         */

        int j = 42;// Definition und Initialisierung

        String s = "Hallo";//Strings sind Zeichenketten
        /*
         * Strings stehen in Java immer in doppelten
         * Anfuhrungsstrichen
         */
    }
}
```

}

Listing 2: Die standart Datentypen von Java