

- Java ist eine Plattform
  - Java als Programmiersprache + Java Compiler
  - Java als Laufzeitumgebung (Java Runtime Environment / Java Virtual Machine)
  - Java Standardbibliothek: wiederverwendbare Programmierbausteine
- Java ist statisch typisiert, d.h. zur Übersetzungszeit prüft der Compiler, ob der Quelltext typkonform ist
  - alle Variablen haben einen festen Datentyp, der sich nicht im Laufe der Zeit verändern kann
  - dadurch werden schon im Vorherein viele Programmierfehler ausgeschlossen
  - Entwicklungstools haben mehr Informationen und können besseren Code-Support leisten
- Java ist eine kompilierte und teilweise interpretierte Programmiersprache
  - Der Quelltext muss in Bytecode übersetzt werden. Dieser ist jedoch nicht von einer realen CPU ausführbar.
  - Die Laufzeitumgebung (Java Virtual Machine) kann den Bytecode dann interpretieren und auf der echten CPU ausführen.
  - Ggf. kann die Laufzeitumgebung häufig interpretierte Programmbestandteile (Hot-Spots) in echten Maschinencode übersetzen (Just-in-Time Compilation)
  - Mittlerweile ist es möglich, Java Quelltext in echten Maschinencode vollständig umzuwandeln (Ahead-of-Time-Compilation)
- Die Java Virtual Machine hat u.a. folgende Aufgaben:
  - Laden und Prüfen des Bytecodes
  - Ausführen des Bytecodes auf der realen CPU
  - Kommunikation mit Betriebssystem
  - Stellt Ressourcen wie Speicher, Rechenzeit, Dateien, Netzwerkverbindungen zur Verfügung
  - Überwachung des gerade ausgeführten Programmes (z.B. Vermeidung von Pufferüberläufen, fehlende/knappe Ressourcen melden)
- Java ist eine objektorientierte Programmiersprache, d.h. Programme werden als Netz miteinander interagierender Objekte modelliert bzw. kodiert. Das entspricht der menschlichen Denkweise, ist also sehr intuitiv.
- Java besitzt eine einfache Syntax und ist "leicht" zu erlernen.