



Java SE, EE und ME

Viktor Styrbul

Softwarearchitekturen
Universität Siegen

Inhaltsverzeichnis

- Historischer Hintergrund
- Grundkonzepte der Sprache
- Eigenschaften von Java
- Merkmale von Java
- Java Editions
 - Java Standard Edition
 - Java Micro Edition
 - Java Enterprise Edition
- Zusammenfassung

Historischer Hintergrund

- Anfang der 1990er Jahre wollte Bill Joy eine neue Programmiersprache schaffen, die alle Vorteile von C und MESA vereinigen sollte
- neue Sprache sollte objektorientiert sein und in ihren Grundzügen auf C++ aufbauen
- Entstehung der Urversion von Java durch „The Green Project“
- *Oak* (**O**bject **A**pplication **K**ernel)

Historischer Hintergrund

- im Herbst 1992 wird ein Gerät mit dem Namen *7 (Star Seven) entwickelt und intern vorgestellt
- Entstehung von „First Person“
- Ausbreitung von Internet
- Richtungswechsel der Entwicklung auf das World Wide Web
- Anforderungen an die Programmiersprache
 - Programmcode über das Netzwerk aufnehmen
 - schadhafte Programme sollten kein Schaden verursachen

Historischer Hintergrund

- die meisten Konzepte aus C und C++ ungeeignet
 - z.B. Zugriffe über ungültige Zeiger, über die man den Speicher beliebig beschreiben kann
- Oak als passender Kandidat erkannt
- Änderung des Namens aufgrund rechtlicher Probleme
- man entschied sich für den Namen Java nach einer Kaffee-Sorte, die von den Entwicklern bevorzugt getrunken wurde

Historischer Hintergrund

- in Java hat man den Prototyp des Browsers mit den Namen WebRunner entwickelt
- der Browser HotJava genannt und im Mai auf der SunWorld '95 der Öffentlichkeit vorgestellt
- Lizenzierung von Java-Technologie durch Netscape
- schon im Januar nächsten Jahres wurde das JDK 1.0 freigegeben, was den Entwicklern ermöglicht hatte, Java-Applikationen und Web-Applets zu programmieren

Grundkonzepte der Sprache

- sie sollte sicher und robust sein
- sie sollte eine objektorientierte, verteilte, einfache und vertraute Architektur haben
- sie sollte interpretierbar, parallelisierbar und dynamisch sein
- sie sollte portabel und architekturneutral sein
- sie sollte sehr leistungsfähig sein

Architekturneutralität und Portabilität

- ein einmal geschriebenes Programm funktioniert auf jeder beliebigen Computerhardware
 - unabhängig von Prozessor
 - unabhängig von anderen Hardwarebestandteilen
- alle primitiven Datentypen sind standardisiert
 - in ihrer internen Darstellung
 - in ihrer Größe
 - in ihrem arithmetischen Verhalten

Einfachheit und Vertrautheit

- Java besitzt einen reduzierten Sprachumfang
 - Operatorüberladung wird nicht unterstützt
 - Mehrfachvererbung wird auch nicht unterstützt
- erfahrene Programmierer haben sofort den Durchblick bei Java
 - syntaktische Ähnlichkeit mit C++
 - ursprüngliche Ähnlichkeit der Klassenbibliothek zu Smalltalk-Klassenbibliothek
 - Einsatz von Design Patterns

Objektorientierung, Verteilt und Leistungsfähigkeit

- Java gehört zu den objektorientierten Sprachen
- Java enthält eine Reihe einfacher Möglichkeiten für Netzwerkkommunikation, von TCP/IP-Protokollen über Remote Method Invocation bis zu Webservices
- dank der dynamischen Optimierungen der Virtual Maschine erbringen Java-Programme ähnliche Geschwindigkeiten wie C++- oder C#-Programme

Interpretierbarkeit und Dynamisch

- Java gehört zu den interpretierbaren Sprachen. Java-Bytecode wird von der Java Virtual Machine interpretiert, bevor sie ihn aus Leistungsgründen optimiert und übersetzt
- Java lässt sich an dynamisch ändernde Rahmenbedingungen anpassen
 - Bibliotheken können neu ausgeliefert werden, ohne die restlichen Programmteile anpassen zu müssen, weil die Module erst bei der Laufzeit gelinkt werden

Sicherheit und Robustheit

- Sicherheit wird bei Java durch folgenden Konzepte realisiert:
 - der Class-Loader steuert die sichere Zuführung von Klasseninformation zu JVM
 - Security-Manager stellt sicher, dass nur auf Programmobjekte zugegriffen werden kann, für die entsprechende Rechte vergeben wurden
- die starke Typisierung, Garbage Collection, Ausnahmebehandlungen auch Verzicht auf Zeigerarithmetik

Parallelisierbarkeit

- Java unterstützt Multithreading. Dadurch ist es möglich den parallelen Ablauf von eigenständigen Programmabschnitten durchzuführen
- bei modernen JVMs wird ein Java-Thread auf Betriebssystem-Thread abgebildet. Dadurch kann ein Programm von Prozessoren mit mehreren Rechenkernen profitieren

Eigenschaften von Java

- Bytecode und die virtuelle Maschine
 - durch den Java-Compiler wird Bytecode für eine virtuelle Maschine erzeugt
 - Bytecode kann man mit Mikroprozessorcode für einen erdachten Prozessor vergleichen
 - ist im Stande die Anweisungen wie arithmetische Operationen, Sprünge und weiteres auszuführen
 - um den Programmcode des virtuellen Prozessors auszuführen, wird der Bytecode nach der Übersetzungsphase von der Laufzeitumgebung ausgeführt
 - somit ist Java eine compilierbare, aber auch interpretierbare Programmiersprache

Eigenschaften von Java

- Just-In-Time Compilation und Optimierung
 - da der Code interpretiert wird gibt es natürlich Geschwindigkeitsprobleme, weil das Erkennen, Dekodieren und Ausführen der Befehle Zeit kostet
 - die ersten Java-Programme waren im Vergleich zu übersetzten C++-Programme deutlich langsamer
 - durch einen JIT-Compiler wird die Ausführung von Programmen beschleunigt, indem er zur Laufzeit die Programmanweisungen der virtuellen Maschine in Maschinencode der entsprechenden Plattform übersetzt
 - so entsteht ein an die Architektur angepasstes Programm im Speicher, das der physikalische Prozessor ohne Interpretation ziemlich schnell ausführen kann

Merkmale von Java

- Java wurde als Sprache entwickelt, die es möglich machen sollte ein fehlerfreies Programm zu schreiben
- alle Klassen und Schnittstellen werden von der Basisklasse „Object“ abgeleitet
- der Objektzugriff in Java wurde über Verweise auf Objekte implementiert, die den aus C++ bekannten Zeiger ähnlich sind
 - Die Verweise werden von der Sprachdefinition als „Reference Values“ bezeichnet, um zu verdeutlichen, dass sie als Call by Value übergeben werden

Merkmale von Java

- aus Sicherheitsgründen wird es nicht erlaubt:
 - Verweis auf andere Verweise
 - Verweis auf Methoden
 - die tatsächliche Speicheradresse rauszufinden und zu modifizieren
- Zeigerarithmetik wurde aus denselben Gründen auch nicht implementiert

Merkmale von Java

- die Unions, Strukturen und Aufzählungstypen können in Java, wie in jeder objektorientierten Programmiersprache, durch geeignete Klassen ersetzt werden
- deswegen werden explizite Typendefinitionen überflüssig
- durch diese Maßnahme ist die Syntax bei Java wesentlich schlanker als z.B. bei C++

Merkmale der Sprache

- alleinstehende Funktionen, die keine Methoden einer Klasse sind, werden bei Java nicht unterstützt
- dadurch werden die Widersprüche und Mehrdeutigkeit in der Klassenhierarchie vermieden
- es werden nur Klassenvariablen und Klassenmethoden unterstützt, das allein stehende Funktionen überflüssig macht

Merkmale von Java

- zusammengehörige Klassen werden in Paketen (englisch packages) zusammengefasst
- durch die Pakete wird die Sichtbarkeit von Klassen eingeschränkt und eine Strukturierung von größeren Projekten sowie eine Trennung des Namensraums für verschiedenen Entwickler ermöglicht
- bei Java wird es explizit zwischen Klassen und Schnittstellen unterschieden
- eine Klasse kann beliebig viele Schnittstellen implementieren, hat aber immer genau eine Basisklasse

Merkmale von Java

- in Java gibt es keine Header Files
- alle Klassen werden komplett in einem File codiert
- dadurch wird es verhindert, dass durch Änderungen in einer zentralen Klasse in einem Header File das gesamte Projekt neu übersetzt werden muss
- so wird bei Java bereits beim Design der Sprache ein häufiger Typ von Fehlern, die in anderen Programmiersprachen auftreten, von vornherein verhindert

Merkmale von Java

■ Garbage-Collector

- Garbage-Collector ist ein Teil des Laufzeitsystems von Java
- kümmert sich um die Verwaltung von angelegten Objekten
- wenn ein Objekt den Wirkungsbereich verlässt, wird es vom System als nicht mehr referenziertes Objekt erkannt und gelöscht
- diese Technik macht es möglich, dass viele Probleme mit Speicher damit verschwinden

Merkmale von Java

- Kein Präprozessor für Textersetzung
 - Präprozessor-Direktiven wie `#include`, `#define` oder auch `#if` zum Einbinden von Prototyp-Definitionen oder zur bedingten Übersetzung hat man in Java aus unterschiedlichen Gründen nicht implementiert:
 - Java besitzt die Datentypen einer festen Größe, die immer gleiche Länge haben, dadurch entfällt die Notwendigkeit, abhängig von der Plattform unterschiedliche Längen zu definieren.
 - da die virtuelle Maschine ohne äußere Steuerung Programmoptimierungen vornimmt, sind Pragma-Steuerungen im Programmcode unnötig.

Merkmale von Java

- da der Compiler die benötigten Informationen wie Methodensignaturen direkt aus den Klassendateien liest, sind Header-Dateien in Java nicht nötig.
- ohne den Präprozessor werden solche Tricks wie `#define private public` oder Makros, die Fehler durch eine doppelte Auswertung erzeugen, von vornherein ausgeschlossen

Java Editions

- „WORA“ (Write Once Run Anywhere) „Einmal schreiben – läuft überall“
- Geräte, auf denen die Programme laufen sollen, sind doch zu verschieden
 - Java Standard Edition (Java SE) ist für gängigen PC mit Betriebssystem Windows, Linux, OS X, oder Solaris
 - Java Enterprise Edition (Java EE) ist für unternehmensweite Programmen mit sehr vielen Anwendern geeignet.(viele Erweiterungen zur Serverprogrammierung)
 - Java Micro Edition (Java ME) hat man speziell für kleine Geräte mit kleiner Rechenleistung und wenig Speicher entwickelt (Handys, PDAs, usw)

Java Standard Edition

- auch Java SE oder auch J2SE
- eine Sammlung von Java-APIs
- Grundlage für die Java Enterprise Edition und Java Micro Edition
- aktuelle Version 6 erschien am 11. Dezember 2006
- ist in der Version Update 26 seit dem 7. Juni 2011 verfügbar

Java Standard Edition

- das Java-SE-6-API besteht aus:
 - lang & util Base Libraries
 - User Interface Toolkits (AWT, Swing, Java 2D)
 - Höheren Base Libraries (I/O, JNI, Serialisierung)
 - Integration Libraries (IDL, JDBC, JNDI, RMI, RMI-IIOP)
- die wichtigsten Klassenbibliotheken befinden sich in den Paketen `java.lang` und `java.util`, Ein- und Ausgaben im Paket `java.io`

Fundamentale Klassen (java.lang)

- in diesem Paket und dessen Unterpakete sind Klassen definiert, die für die grundlegendsten Mechanismen der Sprache Java benötigt werden
 - die absolute Basisklasse Object, von der alle anderen Schnittstellen und Klassen abgeleitet sind
 - die Metaklasse Class, die die Grundlage für den Mechanismus der Selbstreflexion bildet
 - Wrapper für grundlegende Datentypen, wie z.B. Boolean, Character oder Integer
 - die Klasse String für Zeichenketten
 - die Klasse System dient zum Abfragen von Umgebungsvariablen und sogenannten System-Properties

Java Micro Edition

- eine Umsetzung der Programmiersprache Java für Mobiltelefone oder PDAs etc.
- ist gegenüber der JSE abgespeckt worden
- im Gegensatz zu den Standard- und Enterpriseausgaben ist kein einheitliches Paket
- ist in mehrere Configurations und Profiles genannte Gruppen unterteilt, welche die unterschiedlichen Fähigkeiten der verschiedenen Geräte voneinander abgrenzen
- wird durch Standard Edition ersetzt

Java Enterprise Edition

- ist die Spezifikation einer Softwarearchitektur für die transaktionsbasierte Ausführung von Java-Programmen und insbesondere Web-Anwendungen
- größter Konkurrent ist die .NET-Plattform von Microsoft
- Java-EE-Komponenten erfordern als Laufzeitumgebung einen Java EE Application Server

Java Enterprise Edition

- Java EE Application Server stellt verschiedene technische Funktionalitäten zur Verfügung:
 - Sicherheit
 - Transaktionsmanagement
 - Namens- und Verzeichnisdienste
 - Kommunikation zwischen Java-EE-Komponenten
 - Persistenzdienste zum langfristigen Speichern von Java-Objekten
 - Management der Komponenten über den gesamten Lebenszyklus
 - Unterstützung für die Installation

Java Enterprise Edition

- ein Java-EE-Server wird in verschiedene logische Systeme unterteilt, die Container genannt werden
- die aktuelle Spezifikation erfordert die folgenden Container:
 - einen EJB-Container als Laufzeitumgebung für Enterprise Java Beans
 - einen Web-Container als Laufzeitumgebung für Servlets und JavaServer Pages
 - einen JCA-Container als Laufzeitumgebung für JCA Connectoren.
 - einen Java Message Service Provider als Verwaltungssystem für Nachrichtenwarteschlangen

Zusammenfassung

- Urversion von Java am Anfang der 1990er Jahre
- Grundkonzepte
 - Sicherheit, Robustheit
 - Architekturneutralität und Portabilität
 - Einfachheit und Vertrautheit
 - Objektorientierung, Verteilt und Leistungsfähigkeit
 - Interpretierbarkeit und Dynamisch
 - Parallelisierbarkeit
- durch den Java-Compiler wird Bytecode für eine virtuelle Maschine erzeugt und erst dann interpretiert

Zusammenfassung

- durch einen JIT-Compiler wird die Ausführung von Programmen beschleunigt, indem er zur Laufzeit die Programmanweisungen der virtuellen Maschine in Maschinencode der entsprechenden Plattform übersetzt
- diverse Entscheidungen beim Design um häufige Fehlern, die in anderen Programmiersprachen auftreten, von vornherein verhindert
- Java Editions (SE, EE und ME)