

Erste Schritte mit Java

Diese „Einführung in die objektorientierte Programmierung mit Java“ wendet sich an Leserinnen und Leser mit Programmiererfahrung – das heißt: grundlegende Kenntnisse in der prozeduralen Programmierung werden vorausgesetzt. Da viele Lehrbücher scheinbar voraussetzungslos in die objektorientierte Programmierung einsteigen, bedarf diese ungewöhnliche Didaktik einer kurzen Begründung. Ein Blick in die paradigmatische Entwicklung der Programmiersprachen ist dazu hilfreich.

Die Techniken der Programmierung entwickelten sich als praktisches Teilgebiet der Software-Entwicklung über eine recht lange Zeitspanne - „lang“ meint hier: im Vergleich zur sonst üblichen kurzatmigen Rasanz in der Informatik. Es lassen sich mehrere aufeinanderfolgende Entwicklungsabschnitte unterscheiden, in denen jeweils unterschiedliche Prinzipien (Paradigmen) vorherrschten.

Frühe Programmiersprachen (bis ca. 1940) befolgten ein *sequentielles* Programmierschema, bei dem eine Codezeile auf die andere folgt und auch entsprechend abgearbeitet wird; die Codierung erfolgt in einer speziellen, an den jeweiligen Prozessortyp angepaßten Art und Weise. Die umständliche Programmierung in Maschinensprache verlangte bald nach einer leichter lesbaren, menschenverständlicheren Form – und vor allem nach einer Möglichkeit, in sich abgeschlossene Unterprogramme (Subroutinen, Prozeduren bzw. Funktionen) zu definieren und den Programmfluß durch weitere Ausführungs-Bedingungen und -Parameter zu steuern. Das daraus hervorgegangene *prozedurale* Paradigma legt fest, daß ein Programm aus einer Anzahl von Funktionen besteht, die aus dem Hauptprogrammabschnitt (der main-Funktion) beliebig oft aufgerufen werden. Es ist ein modulares Prinzip, das den Aufbau des Programms nach Teilaufgaben (Funktionen) strukturiert und dabei auch festlegt, das der Programmfluß nach der Abarbeitung einer Funktion automatisch an die richtige Aufrufstelle zurückfinden muß - nur so ist dynamischer Funktionsaufruf möglich. Die in den frühen siebziger Jahren entstandene, auch heute noch wichtige Sprache C ist ein typisches Beispiel für eine prozedurale Programmiersprache.

Die *Objektorientierten* Programmiersprachen sind als Reaktion auf die internationale „Softwarekrise“ in den 60-er Jahren entstanden, als es klar wurde, worum es bei der Software-Entwicklung eigentlich ging – nämlich darum, einen bestimmten Realitätsausschnitt (die jeweilige geschäftsbezogene Aufgabenstellung) in Software abzubilden. Dazu muß man sich aber an den Prinzipien der Wirklichkeit orientieren, die von handlungsfähigen Einheiten (und nicht nur von abstrakten Datenstrukturen) geprägt ist. Die herkömmlichen prozeduralen Sprachen erwiesen sich als zunehmend ungeeignet für die Software-Abbildung solch komplexer Wirklichkeitsausschnitte. Es bedurfte dazu eines neuen Konzeptes mit selbstständig agierenden Einheiten (Objekten), die Datenwerte (Eigenschaften, Attribute) und zugehörige spezifische Verhaltensweisen (Methoden, objekttypische Funktionen) kapseln. Attribute und Methoden werden in den Klassen definiert; Objekte stellen reale Ausprägungen der Eigenschaften ihrer Klasse dar und sind mit individuellen Datenwerten gefüllt.

Aus diesem kurzen geschichtlichen Abriß wird klar, daß das objektorientierte Paradigma eine Erweiterung und Ergänzung des prozeduralen Paradigmas ist. Die grundlegende Syntax und Grammatik einer Sprache wird dadurch nur unwesentlich verändert. In der Tat sind C, C++, Java und C# syntaktisch und grammatikalisch sehr eng miteinander verwandt und verwenden zu 98% identische Sprachbestandteile (Befehle, Anweisungen, Operationen).

Es ist didaktisch geschickt, das Erlernen der Programmiersprache (mit ihrem grundlegenden Sprach- / Befehlsschatz) von dem Erlernen des objektorientierten Prinzips zu trennen. Man kann zwar auch mit Java versuchen, prozedural zu programmieren, aber Java ist von Grund auf objektorientiert - letztlich würde man die Sprache mißbrauchen und zweckentfremden. Die elementare Grammatik und Syntax von Java (Datentypen, Kontrollstrukturen, Schleifen, Funktionen, selbstdefinierte Datentypen) ist direkt von C abzuleiten. In den folgenden Kapiteln wird deshalb nur auf eventuelle Unterschiede hingewiesen; Nichterwähnung bedeutet syntaktische Identität mit C.

Geschichte der Programmiersprache Java

Die inhaltliche Ähnlichkeit von C und Java hat auch historische Wurzeln. Java wurde von einer Gruppe erfahrener Programmierer ([James Gosling](#), [Patrick Naughton](#), [Mike Sheridan](#) et al.) und mit starker Unterstützung von [Bill Joy](#) aus mehreren Vorläufern Anfang der 90er Jahre entwickelt. Sie waren Angestellte der Firma SUN-Microsystems in Kalifornien und hatten sich vorgenommen, einige bekannte Schwachpunkte von C zu überwinden und zudem eine Sprache zu entwickeln, die für mobile Geräte und vernetzte Anwendungen verwendbar ist. Die neue Sprache sollte von den Wurzeln an unabhängig von Hardware (Prozessortyp) und Betriebssystem sein. Entstanden ist eine Programmiersprache von hohem Abstraktionsniveau, die alle Prinzipien der objektorientierten Programmierung unterstützt.



“In their book *The C Programming Language*, Brian Kernighan and Dennis Ritchie said that they felt that the C language ‘wears well as one’s experience with it grows.’ If you like C, we think you will like the Java programming language. We hope that it, too, wears well for you.”

James Gosling, *Java Language Specification* Second Edition

Internetseiten zur Geschichte und Entwicklung von Java:

<http://www.wired.com/wired/archive/3.12/java.saga.html>
<http://www.wired.com/wired/archive/3.12/java.saga.html?pg=2>
<http://www.blinkenlights.com/classiccmp/javaorigin.html>

Bill Joy:

<http://www.wired.com/wired/archive/8.04/joy.html>
http://news.cnet.com/8301-13860_3-20005814-56.html

Die Entwicklung der Programmiersprache Java vollzog sich in vielen kleinen Schritten und mit mehreren Umbenennungen (Oak, Green, Java). Lange Zeit war überhaupt nicht klar erkennbar, auf welche Frage die neue Programmiersprache eine Antwort sein könnte. Eine Zeitlang versuchte man, Haushalts-Geräte mit einer eingebetteten Betriebssystem-Umgebung zu versehen, die Java-basiert war und in das Geschäft mit Interaktivem TV einzusteigen. (Star-7, Set-Top-Box, ITV). Aber alle solche Initiativen scheiterten letztlich und die Entwickler waren schon nahe daran, aufzugeben. Durch den Popularitätsschub des Internets um 1993 öffnete sich plötzlich eine unvorhergesehene Nische und die Vorteile von Java wurden praktisch verwendbar. Die Programmierung eines Webbrowsers (WebRunner, HotJava) war sehr erfolgreich und die Bedeutung der jetzt Java genannten Sprache wurde immer deutlicher. Letztlich waren es drei Faktoren die den Durchbruch herbei führten:

Im Internet übertragbarer kompilierter Zwischencode (class-code, byte-code)
Interpretierende Ausführung – Plattformunabhängigkeit
Strikte Objektzentriertheit

Die erste offizielle Version wurde Java 1.0.2 genannt. Zur weiteren Entwicklung siehe z.B.: https://en.wikibooks.org/wiki/Java_Programming/History