

von Paul Conte

Beobachtet man die Aufmerksamkeit, die Java in den Medien genießt, drängt sich die Frage auf, ob Java wirklich ein wunderbares neues Werkzeug ist, mit dem man Software herstellen kann, die besser ist als alles, was wir jemals erlebt haben – oder nutzen die Hersteller schamlos die clevere Namensgebung aus, um uns davon zu überzeugen, daß ihre Produkte so unverzichtbar sind wie Kaffee- der wirkliche Java? Skepsis scheint angebracht. Doch die Erfahrungen mit Java zeigen, daß es sich hierbei um eine wirklich aufregende neue Entwicklung im Bereich der Software-Technologie handelt. Bilden Sie sich selbst eine Meinung oder machen Sie den Test „Ihr Javaprofil“ und stellen Sie fest, wie gut Java zu Ihnen paßt.



Java Applets und Anwendungen

Java ist eine Programmiersprache, mit der * kleine Programme (Applets) in Word-Seiten * die Entwicklung von eigenständigen Anwendungen eingefügt werden können. Wie die meisten Programmiersprachen verfügt Java über die Möglichkeit zum Deklarieren von Variablen, Durchführen von Rechenvorgängen zur Ablaufkontrolle mittels If-Abfragen und Schleifenstrukturen. Applets laufen immer in einem Web-Browser ab, dem sogenannten „Sandkasten“ und sind daher in ihren Fähigkeiten begrenzt. So kann ein Applet normalerweise nicht auf eine Festplatte zugreifen. Applets werden üblicherweise dafür verwendet, daß Interface zu einer Web-Seite aufzumotzen oder die Usereingabe zu verbessern.

Richtige Anwendungen: Java als ein besseres C++

Obwohl Java dem Web seinen kometenhaften Aufstieg verdankt, so ist doch der Sandkasten nicht seine wahre Bestimmung. Java kann auch anstelle von C++, Visual Basic, Delphi und – mit der Zeit – RPG verwendet werden, um vollständige Anwendungen zu schreiben. Noch wichtiger ist, daß man Java-Anwendungen für fast jede vorstellbare Plattform schreiben kann, z.B. für alle wichtigen IBM-Betriebssysteme (OS/400, OS/390, AIX, OS/2), alle Versionen von Windows, MacOS, und praktische alle Versionen von Unix. C++ ist die einzige vergleichbare Sprache mit einer derartigen Bandbreite. Java verfügt über alle wichtigen Programmierfunktionen: arithmetische Rechenvorgänge, Zeichenketten-Operationen, Programmlogik, Datenbankzugriff, User Interface und Communications. Es steht C++ in nichts nach – mit der einen Ausnahme, daß es keinen Pointerzugriff auf Speicheradressen gibt, was aber für kommerzielle Anwendungen nicht erforderlich ist. Ein wichtiger Punkt ist auch, daß Java Applets zwar interpretiert werden, Java Anwendungen aber zur Verbesserung der Performance kompiliert werden können. Es mag zwar noch eine Weile dauern, bis Java Compiler das hohe Leistungsvermögen der heutigen C++ Compiler erreicht haben. Doch sollte kompiliertes Java für die meisten kommerziellen Anwendungen ausreichend schnell laufen. Java hat zwei Fähigkeiten, die bei C++ fehlen: Einfachheit und Sicherheit. Java kommt ohne einige der sehr

komplexen C++ Konstrukte aus. Hierzu gehören die mehrfache Vererbung (Klassen) und der Einsatz von Pointern (Speicheradressen). Stattdessen bietet Java eine Anzahl sehr eleganter Features, die den Programmiervorgang und die Lesbarkeit des Codes wesentlich verbessern. So kann man mit Java Interfaces für relationale Datentypen definieren, und zwar unabhängig von der Implementierung der Datentypen. Dadurch entfallen die meisten Probleme, die sich aus dem Prinzip der mehrfachen Vererbung bei C++ ergeben. Datentypen und Vererbung bilden die wichtigsten Stützfeiler objektorientierter Programmiersprachen und sind für Systemsoftware wie kommerzieller Anwendungen gleichermaßen unverzichtbar. In Java gibt es keine direkte Speicherverwaltung und keine Objekt-Deallocation. Diese beiden Features führen bei C++ und Delphi wiederholt zu Programmabstürzen und dem Blockieren von PC's. Mit V3R7 hat IBM RPG IV mit Pointern ausgestattet, so daß die Gefahr der Pointerfehler jetzt auch in RPG IV-Programmen besteht. Bei Java erinnert die Art des Umgangs mit Objekten stark an SmallTalk, wobei auch hier die Möglichkeit zur automatischen Speicherplatzfreigabe besteht. Es gibt zwar auch bei Java verschiedene Möglichkeiten, Zuordnungsfehler zu machen, doch fängt der Compiler oder die Runtime-Umgebung fast alle derartigen Fehler ab und gibt eine aussagefähige Nachricht aus. So könnte ein Java Programm zwar immer noch abstürzen, doch würde ein Fehler sich nicht so dramatisch auswirken wie bei C++, Delphi oder RPG IV, und es ist viel leichter, Fehler aufzuspüren als mit einem C++ Fehlerpointer. Von allen verbreiteten Programmiersprachen verfügt Java über den elegantesten Umgang mit Datentypen. Javas Datentypensystem ist wesentlich einfacher und verständlicher als C++ und bietet wesentlich mehr Sicherheit. Auch Delphi und Visual Basic 4.0 (VB 5.0 ist noch zu neu für einen Vergleich) ist es in dieser Hinsicht weit überlegen. Mit diesem Datentypensystem kann man Datentypen wie Angestellter, Projektressourcen und Druckausgabe erstellen, um sie dann als Programmvariablen zu deklarieren. Compiler und Runtime prüfen die Zulässigkeit der Operationen, die mit einer Variablen durchgeführt werden, auf Grund des Datentyps dieser Variablen. Die meisten Sprachen mit einem guten Datentypensystem benötigen entweder komplexe Regeln für die Definition relationaler Datentypen (wie bei C++) oder gewähren nur begrenzte Möglichkeiten zur Definition relationaler Datentypen wie bei Delphi.

Sie müssen sich als Abonnent anmelden um den hier fehlenden Teil des Inhalts zu sehen. Bitte [Login](#) für Zugriff.

Noch nicht Abonnent? [Sonderaktion nutzen](#).

- [7 Euro/Monat NEWSabo digital - sofort zugreifen & online bezahlen.](#)
- [13,5 Euro/Monat NEWSabo plus inkl. 5x Logins & Print-Ausgaben - sofort zugreifen & per Firmen-Rechnung bezahlen.](#)