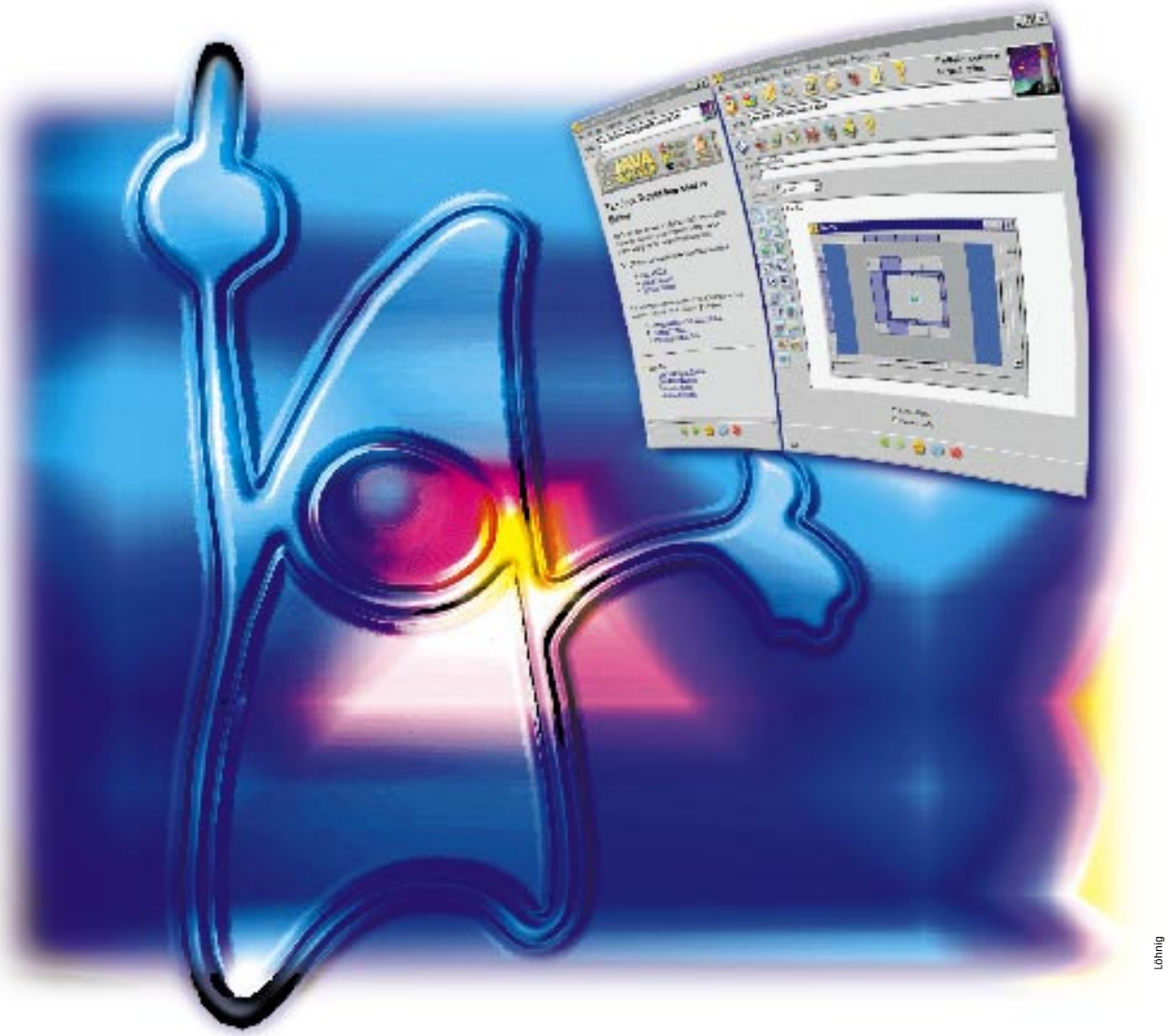


Löhnig

# Werkzeuge für Java

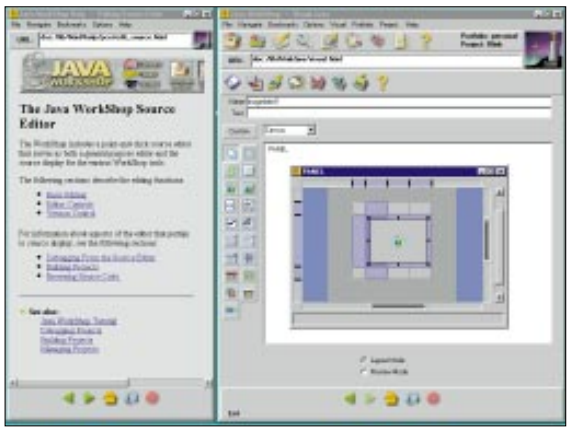
***Wer mit der Sprache Java Software entwickeln will, braucht Werkzeuge, die dabei helfen. CHIP hat sich die wichtigsten Entwicklungsumgebungen angesehen und zeigt, wofür sie sich eignen.***

Programmiersprachen verbindet eine Gemeinsamkeit: Erst mit den passenden Werkzeugen zur Bearbeitung des Quelltextes, zur schnellen Übersetzung und zur Fehlersuche werden die Sprachen lebendig. Das gilt auch für Java, das jüngste und derzeit berühmteste Mitglied der Szene.

Die neue Sprache sorgt immer noch für Aufsehen. Aufgrund der Plattformunabhängigkeit der fertigen Software versprechen sich viele Firmen eine Kostenersparnis, andere rechnen mit höherem Absatz ihrer Software, da neue Anwendergruppen hinzukommen. Grund: Eine Anwendung muß nur einmal programmiert werden, kann dann aber auf verschiedenen Rechnerarchitekturen und Betriebssystemen eingesetzt werden.

Für Java sind einige integrierte Entwicklungsumgebungen auf dem Markt. Fast ausnahmslos handelt es sich hierbei um Werkzeuge der ersten Generation. Firmen wie Symantec, Borland, IBM und Microsoft haben ihre bestehenden Werkzeuge für C++ mit einem Java-Compiler verheiratet. Das Resultat erweist sich als funktional, doch fehlt ihm das Flair von Java, da diese Umgebungen alle nicht in Java programmiert wurden.

Erst die Vertreter der nächsten Generation erfüllen das Kriterium, das Borland etwa bei Delphi angewandt hat: Die Entwicklungsumgebung entsteht aus der eigenen Sprache. Bei Java hat das einen dramatischen Effekt: Die Umgebungen laufen dann nicht nur auf einem, sondern auf allen Java-fähigen Betriebssystemen.




**Alles im Internet-Look: *Java Workshop* integriert sämtliche Werkzeuge in einem Web-Browser**

Zu haben ist derzeit schon der *Java Workshop* von Sunsoft, einem Tochterunternehmen des Java-Erfinders Sun Microsystems. Angekündigt wurden *Latte* von Borland und *Tazza*, eine Experimentalumgebung von IBM. Die Nachteile dieser Anwendungen liegen in der Geschwindigkeit. Da Java erst zur Laufzeit in Maschinencode übersetzt wird, sind die Oberflächen nicht so schnell wie die in C++ programmierten.

## JAVA DEVELOPMENT KIT
### Enthält alles, was der Programmierer braucht

Startpunkt für die Entwicklung von Java-Software ist sicherlich der *Java Development Kit* (JDK) von Sun. Derzeitige Version ist die 1.0.2, erhältlich für Solaris, Windows 95/NT und Mac OS. Für die nichtkommerzielle Nutzung kann man sich diesen Kit kostenlos aus dem Netz herunterladen (*http://java.sun.com* oder in AOL, Kennwort JAVA).

JDK enthält alles, was der Programmierer braucht: den Compiler *javac*, den Byte-Code-Interpreter *java*, den Debugger *jdb* und eine Reihe von anderen Hilfsprogrammen. Dabei handelt es sich allerdings nur um Kommandozeilen-Versionen. Eine Entwicklungsumgebung gehört nicht dazu und auch eine Möglichkeit, um die Oberfläche der neuen Software visuell zu gestalten, fehlt. Mit einer Portion Initiative kann ein Entwickler sich mit einem Editor und einer Stapeldatei eine Umgebung zusammenbasteln. Zu haben ist auf dem Server von Sun auch die Dokumentation zu Java.

**Urteil:** JDK bildet die Basis für die Java-Entwicklung. Für erste kostenlose Kontakte mit der Sprache in Form auch von umfangreichen Beispielen ist JDK auf jeden Fall einen Download wert.

## JAVA WORKSHOP
### Interessante Architektur, aber nicht sehr schnell

In einer zeitgemäßen Architektur präsentiert sich Java Workshop von Sunsoft. Die Oberfläche ist ein an Hotjava angelehnter Web-Browser, der spezifische HTML-Seiten lädt. Diese wiederum enthalten Aufrufe zu einer Art Java-Applets, die dann etwa den Quellcode-Editor darstellen. Für den Anwender bedeutet das, daß er zu jeder Zeit mit den Navigationsknöpfen beispielsweise vom Build-Manager zum Editor zurückspringen kann. Mit dieser sehr intuitiven Technik kommt auch ein Einsteiger sofort zurecht.

Java Workshop umfaßt alle Programmteile, die für das Erzeugen von Java-Anwendungen benötigt werden. Die Projektverwaltung sortiert alle Anwendungsmodule. Ein Editor gestattet die Bearbeitung des Quelltextes. Der Debugger hilft bei der Fehlersuche, und ein visueller Editor macht die Gestaltung der Oberfläche zum Kinderspiel. Mit einem Mausklick startet man über den Build-Manager die Übersetzung der Software. Ein weiterer Klick ruft das Programm auf.

Selbstverständlich ist auch eine Hilfefunktion integriert. Die Hilfetexte sind – wie könnte es anders sein – als HTML-Seiten abgelegt. Der Workshop zeigt sie in einem separaten Fenster an. Ein Online-Tutorial führt den Neuling durch alle Schritte bis zum fertigen Programm.

Besonders hilfreich ist auch die Dokumentierfunktion von Java Workshop: Klickt man auf das Symbol für den Klassenbrowser, legt die Software eine HTML-Seite mit der Klassenhierarchie des Projekts an. Als Hyperlinks sind hier auch die Methoden mit aufgeführt, die überschrieben wurden. Ein Klick auf so einen Link öffnet den Quellcode an der entsprechenden Stelle.

**Urteil:** Java Workshop ist allein schon wegen der neuen Architektur interessant. Da die Software aus Java-Code besteht, ist die Geschwindigkeit in manchen Fällen nicht optimal.

## CAFÉ
### Gute integrierte Umgebung mit einer schlechten Hilfe

Café zählt zu den ersten Entwicklungsumgebungen für Java, die auf dem Markt waren. Als Oberfläche dient hier Symantec C++ für Windows, das an die Notwendigkeiten von Java angepaßt wurde. Das ist sicher einer der Gründe dafür, daß Café so früh verfügbar war. Die Bedienung von Café unterscheidet sich etwas von den anderen klassischen Umgebungen. Über Reiter im oberen Bereich wechselt der Anwender schnell zwischen verschiedenen Fensteranordnungen. So kann man sich unterschiedliche Layouts zusammenstellen: eines für die Bearbeitung der Quelltexte, eines zum Debuggen und eines, um das Projekt zu verwalten.

Für ein neues Projekt erzeugt Café auf Wunsch den Quellcode und eine minimale HTML-Seite, die das Applet aufruft. Der Anwender kann zwischen einem Kommandozeilen-Programm, einer Anwendung mit einem eigenen Fenster und einem Applet wählen. Entscheidet man sich beispielsweise für ein Fensterprogramm, legt die Software den Quellcode für ein Applikationsfenster mit Menüleiste und den wichtigsten Menüs wie „Datei“ oder „Bearbeiten“ an. Die Ressourcen wie Menüs und Dialoge liegen in einer eigenen Datei und können mit dem Ressourcen-Editor bearbeitet werden. ▶

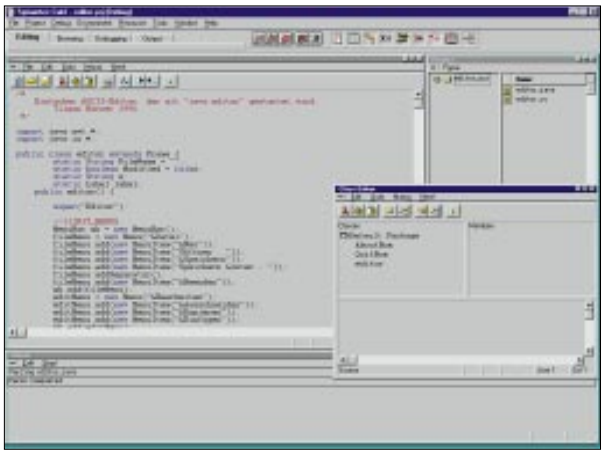
**Basics**

### Das ist Java

Das Besondere an der neuen Programmiersprache Java ist, daß damit geschriebene Programme auf allen Betriebssystemen laufen, für die es einen sogenannten Interpreter gibt. Diese Software liest jeden Befehl einer Anwendung und führt die Anweisung aus. Es gibt Interpreter bereits für Solaris, Windows 95/NT, Mac OS und Linux. In OS/2 Version 4 ist der Interpreter eingebaut, für Windows 3.x in Arbeit.

Der Effekt ist dramatisch: Eine auf Windows 95 entwickelte Software kann auch auf den anderen Betriebssystemen ablaufen. Der potentielle Kundenkreis steigt damit an.

Nachteil von Java ist die Geschwindigkeit. Durch die Übersetzung zur Laufzeit sind Java-Programme langsamer als Software, die auf das Betriebssystem optimiert ist.



**Vorreiter: *Café* von Symantec war eine der ersten Entwicklungsumgebungen für Java**

Der intgrierte Debugger durchläuft ein Applet schrittweise. Selbstverständlich bietet dieses Werkzeug auch Haltepunkte (Breakpoints), überwachte Ausdrücke (Watches) und einen Überblick über den Aufrufstapel (Call stack).

Auf Knopfdruck füttert Café den Quellcode in den Compiler. Hier hat man die Wahl zwischen dem Original *javac* von Sun und einem Code von Symantec. Auf Wunsch startet die Software nach einer fehlerfreien Übersetzung die Anwendung.

Enttäuschend ist bei Café die Hilfe. Kaum jemand kann sämtliche Klassen eines Packages und deren Methoden im Kopf behalten. Zwar gibt Café Auskunft, welche Funktion wozu gehört, doch sind die Erklärungen der Wirkungsweise äußerst dürftig. Hier empfiehlt es sich, die Dokumentation des Java-API (Application Programming Interface) von Sun aus dem Internet herunterzuladen. Sie erklärt bei weitem ausführlicher.

**Urteil:** Café versorgt den geübten Programmierer mit einer guten integrierten Umgebung. Die Programmhilfe zur Java-API läßt allerdings zu wünschen übrig.

## TAZZA

### Derzeit noch eine Studie – auf jeden Fall ansehen

Eine sehr frühe Version einer Entwicklungsumgebung für Java-Module stellt IBM auf dem Server *http://www.alphaworks.IBM.com* kostenlos zur Verfügung. Tazza (italienisch für „Tasse“) ist wie Java Workshop in Java geschrieben. Dem Namen könnte man durchaus noch ein „Visual“ voranstellen, da der Anwender das neue Programm aus vorhandenen Elementen zusammenklicken kann. Aus der Komponentenpalette holt man sich dazu erst einmal die Elemente wie Schaltflächen, Editierfelder oder Auswahlboxen, die das Programm später benötigt. Über einen Klick auf ein Icon gelangt man in den Formulardesigner und legt dort die Anordnung der Komponenten im Fenster der Applikation fest. An dieser Stelle merkt man Tazza noch die Alpha-Version an: Bedienung und Bildschirmdarstellung haben ihre Macken.

Der Anwender kann zwischen verschiedenen angebotenen Layout-Grundformen wählen. Entscheidet er sich beispielsweise für ein Grid-Layout, ordnet Tazza alle Elemente selbständig nebeneinander an. Auf diese Weise muß er nicht einzelne Elemente in Feinarbeit ausrichten, denn die Klasse justiert sie für ihn. Zur Wahl steht aber auch ein Freeform-Layout, bei dem der Anwender die Elemente nach Wunsch plazieren kann.

Nach der Optik geht es an die Logik. Über einen Event-Designer weist man den Komponenten bestimmte Aufgaben zu. Per Mausklick wird die Anwendung in Java-Bytecode übersetzt und gestartet.

**Urteil:** Tazza ist eine sehr interessante Studie, die als solche noch ihre Macken hat. Da die Software kostenlos erhältlich ist, sollte man sie sich einmal ansehen.

## JAMBA

### Wenn Sie nichts mit Quellcode zu tun haben wollen

Auch ohne das mühselige Eingeben von Java-Code ist man in der Lage, seine Web-Seiten mit Leben zu füllen. *Jamba* der Firma Aimtech ist ein Werkzeug, mit dem das möglich ist. Die Arbeitsweise hier erinnert an das Zusammenstellen einer Präsentation oder einer Multimedia-Show. Ein Projekt besteht bei Jamba aus einzelnen Seiten. Es können Hintergrundseiten gestaltet und anderen Seiten „untergeschoben“ werden. Dadurch erscheint beispielsweise das Firmenlogo auf allen Seiten, ohne daß man es jeweils separat einfügen muß.

Über eingesetzte Schaltflächen kann der Anwender später zwischen den Seiten hin- und herspringen. Die Blätter nehmen natürlich noch andere Elemente auf, zum Beispiel Textboxen, Auswahlfelder, Bilder oder Schaltflächen. Aber auch der Aufruf eines Web-Browsers mit einer vorgegebenen URL ist kein Problem. Mit dieser Funktion baut man beispielsweise eine Hilfefunktion in die neue Applikation ein: Der Web-Browser holt sich spezielle HTML-Hilfe-Seiten über das Netzwerk auf den eigenen PC und zeigt sie an. Mit von der Partie ist ein Objekt, um über das Common Gateway Interface (CGI) andere Programme zu starten und mit Daten zu versorgen.

Wie bei Programmiertools nach dem Muster von Visual Basic verändert man die Attribute der Objekte im Eigenschaftseditor. Hier stellt der Anwender etwa eine andere Hintergrundfarbe ein. Die To-Do-Liste verbindet ein bestimmtes Ereignis mit einer Aktion. Soll durch Mausklick auf eine Schaltfläche etwa eine andere Seite aufgeschlagen werden, läßt sich das Klick-Ereignis über Auswahlboxen mit der Aktion „Umblättern“ verknüpfen. Der Anwender muß auch dafür keinen Code eingeben.

Mit einem Mausklick startet der Programmierer das Projekt und hat so einen unmittelbaren Eindruck von Aussehen und Funktionsweise. Die Weitergabe der fertigen Anwendung wird mit einem wei-

## Basics

### Lange Dateinamen und Groß-/Kleinschreibung

Der Neuling im Java-Land muß zweierlei beachten. Zum einen verwendet Java lange Dateinamen. Der Compiler *javac* erwartet Quellcodedateien mit der Namenserweiterung JAVA. Wird aus Versehen so eine Datei mit einem alten DOS-Programm kopiert, verstümmelt es den Namen.

Der andere Fallstrick ist die Unterscheidung von Groß- und Kleinschreibung. Am besten gewöhnt man sich die Schreibweisen der Packages von Java an. Danach werden Konstanten groß geschrieben (WINDOW_MOVED), Funktionen gemischt, erster Buchstabe klein (setText), Variablentypen klein (int), Klassenbezeichner gemischt, erster Buchstabe jeweils groß (FileInputStream).

teren Menübefehl vorbereitet. Ein Assistent führt den Anwender durch die notwendigen Schritte, um das Programm beispielsweise auf Disketten zu bannen. Das Ergebnis befriedigt quantitativ allerdings nicht. Jamba erzeugt keinen echten Java-Code, sondern läßt eine Projektdatei über den Jamba-Player ablaufen. Lediglich der Player ist eine Java-Anwendung. Aus diesem Grund müssen gewisse Klassen, die das Laufzeitmodul benötigt, immer mitgegeben werden. Für eine kleine Anwendung, die nur aus einer Seite besteht, entstehen Dateien, die je nach Kompression zwischen 2 und 3 Megabyte groß sind.

**Urteil:** Jamba ist für Anwender interessant, die nichts mit Quellcode zu tun haben wollen. Allerdings muß man Abstriche beim Verhältnis zwischen Umfang und Leistung des generierten Programms machen.

## VISUAL J++

### Zufriedenstellendes Tool mit glänzender Online-Hilfe

Trotz des späten Bekenntnisses zum Internet und dessen Technik hat auch Microsoft ein Werkzeug für die Programmierung von Java-Anwendungen zu bieten. Wie bei Symantec und Borland setzt *Visual J++* auf die Visual C++ auf. Somit integriert sich Visual J++ in das *Microsoft Developer Studio*, eine Oberfläche für alle Programmiertools aus dem Hause Microsoft. Darüber hinaus sind Online-Dokumentationen und Beispielprogramme eingebaut.

In drei Fenstern kommunizieren Programm und Anwender miteinander. In einem Fenster zeigt die Software die Module, Methoden und Variablen an. Ein Doppelklick auf eine Methode bringt im zweiten Rahmen den Quellcode auf den Schirm. Das dritte Fenster zeigt unter anderem Meldungen des Compilers. Über die Fenster hat der Anwender auch Zugriff auf die Online-Dokumentation. Ein Klick genügt, um aus der Projektanzeige in die Hilfe zu schalten.

Visual J++ erzeugt auf Wunsch den Quellcode für eine Rumpfanwendung. Hier kann man wählen, ob es sich um ein Applet für einen Browser oder eine eigenständige Anwendung handeln soll. Wie Café versieht Visual J++ den Quellcode mit Kommentaren, die zeigen, wo noch Code fehlt. Eine HTML-Seite legt die Software für das Applet ebenfalls an.

Auf Knopfdruck übersetzt der Compiler das Projekt. Je nach eingestelltem Ziel ruft Visual J++ anschließend den eingestellten Browser oder den eigenen Interpreter *jview* auf.

**Urteil:** Visual J++ erfüllt seine Aufgabe sehr zufriedenstellend und glänzt durch die Integration der Online-Dokumentation in die Entwicklungsumgebung.

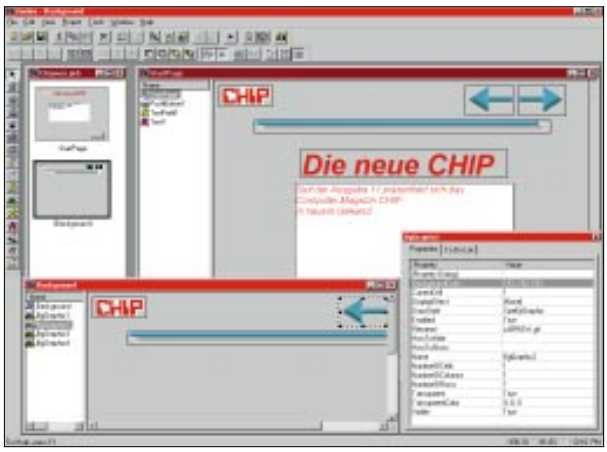
## LATTE

### Verspricht interessante Ansätze

Von Borlands *Latte* (italienisch für „Milch“) lag CHIP zum Zeitpunkt des Tests leider noch keine Version vor. Deshalb stützen sich die Informationen auf Aussagen von Borland. Lattes Oberfläche erinnert an Delphi, eine Pascal-Umgebung. In einem Fenster wird die Anwendung aus einzelnen Komponenten zusammengesetzt. In einem weiteren Fenster lassen sich die Elemente durch Code verbinden.

Ein dreigeteiltes Fenster zeigt hierarchisch die Klassen, die enthaltenen Funktionen und den Quellcode jeder Funktion. Auf diese Weise lassen sich gezielt einzelne Funktionen ansteuern.

**Urteil:** Latte verspricht interessante Ansätze. Für eine Beurteilung fehlt leider noch die Software.




**Software machen ohne Quelltextsachen: Bei *Jamba* muß der Anwender keinen Code eintippen**

## FAZIT

Trotz des kindlichen Alters von Java haben sich viele Firmen ins Zeug gelegt, um Werkzeuge dafür bereitzustellen. Was man an Entwicklungsumgebungen bekommt, ist durchaus brauchbar. Richtig spannend wird es allerdings erst, wenn die Werkzeuge der nächsten Generation wie Latte oder Tazza auf den Markt kommen. *Tilman Börner*

### Weitere Umgebungen

Leider reicht der Platz nicht aus, um alle Werkzeuge, die bei der Programmierung von Java-Anwendungen unter die Arme greifen, vorzustellen. So treibt der Shareware-Bereich täglich neue Blüten. Daneben hat Borland sein *C++ 5.0* mit Fähigkeiten zur Java-Code-Erzeugung ausgestattet. Mit dabei ist auch der *Just-in-Time Compiler* von Borland, der eine Java-Anwendung beim Aufruf in Maschinencode übersetzt und so die Ausführungsgeschwindigkeit erhöht.

Nach der Markteinführung des Betriebssystems OS/2 Warp Version 4, das einen Java-Interpreter bereits im Kern integriert hat, bietet die Firma IBM mit *Visual Age* auch eine integrierte, visuelle Entwicklungsumgebung für Java an.

**Weiterführende Literatur:** Das ist Java, „Reif für die Insel“, CHIP 3/1996, S. 212

So funktioniert Java, „Java“, CHIP 8/1996, S. 216

**Anbieter: JDK** Sun Microsystems, Bretonischer Ring 3, Postfach 1336, 85630 Grasbrunn, Tel. (089) 46 00 80, *http://java.sun.com*, Preis: kostenlos für private Nutzung

**Café** Symantec, Grafenberger Allee 136, 40237 Düsseldorf, Tel. (0211) 991 70, *http://www.symantec.com*, Preis: ca. 260 Mark

**Visual J++** Microsoft, Edisonstr. 1, 85716 Unterschleißheim, Tel: (089) 317 60, *http://www.microsoft.com*, Preis: ca. 200 Mark

**Java Workshop** siehe JDK, Preis: bis Ende 1996 ca. 200 Mark, dann ca. 550 Mark

**Jamba** Aimtech, über DMC, Friedrichstr. 22, 70736 Fellbach, Tel. (0711) 510 10, *http://www.aimtech.com*, Preis: ca. 500 Mark

**Latte** Borland, Monzastraße 4c, 63225 Langen, Tel. (06103) 97 90, *http://www.borland.com*, Preis: stand noch nicht fest

**Visual Age** IBM, Pascalstr. 100, 70569 Stuttgart, Tel. (0711) 78 50, *http://www.ibm.com*, Preis: ca. 790

**Borland C++** Borland, Monzastraße 4c, 63225 Langen, Tel. (06103) 97 90, *http://www.borland.com*, Preis: ca. 1500 Mark (Dev. Suite)