



Torsten Schlabach

# Java-Programmierung mit Visual J++ 1.1

Java verstehen und effektiv nutzen



**ADDISON-WESLEY**

---

An imprint of Addison Wesley Longman, Inc.  
Bonn • Reading, Massachusetts • Menlo Park, California  
New York • Harlow, England • Don Mills, Ontario  
Sydney • Mexico City • Madrid • Amsterdam

# Inhaltsverzeichnis

<b>Einleitung</b> .....	<b>11</b>
Wie dieses Buch aufgebaut ist	11
Welche Teile Sie lesen sollten	12
Wenn Sie als Entwickler bereits mit Objekten gearbeitet haben	13
<b>1 Das Foyer</b> .....	<b>15</b>
1.1 FAQ	15
1.2 Von Rationalisierung zur Informationsqualität	17
1.2.1 Das Ende der Rationalisierung	17
1.2.2 Kostenfragen	18
1.2.3 Der Return on Investment	18
1.2.4 Grenzen einreißen	19
1.2.5 Technische Bausteine	19
1.3 Objekte	20
1.3.1 Grundgedanke Objekt	21
1.3.2 Vererbung	21
1.3.3 Wiederverwendung	22
1.3.4 OO-Design	22
1.4 Die Kapitel in diesem Buch	23
1.4.1 Kapitel 2	23
1.4.2 Kapitel 3	23
1.4.3 Kapitel 4	23
1.4.4 Kapitel 5	24
1.4.5 Kapitel 6	24
1.5 Zusammenfassung	25
<b>2 Java, Visual J++, Microsoft, Sun und einfach alles</b> .....	<b>27</b>
2.1 FAQ	27
2.2 Microsoft und Java	28
2.2.1 Embrace und Extend	28
2.2.2 Die Rechte an Java	29
2.2.3 Ist Visual J++ Java?	30
2.3 Die Java-Technologie	30
2.3.1 Einmal entwickeln, überall einsetzen	30
2.3.2 Die Bausteine von Java	31

2.3.3	Die Programmiersprache Java	33
2.3.4	Klassenbibliotheken	34
2.3.5	Der Java-Compiler	37
2.3.6	Die Java Virtual Machine	39
2.4	Die Bestandteile von Visual J++	40
2.4.1	Das Visual Studio 97	40
2.4.2	Projektverwaltung	41
2.4.3	Der Editor	42
2.4.4	Die Wizards	42
2.5	Windows und Java	43
2.5.1	Java-Packages für Windows-Plattform-Funktionen	44
2.5.2	Komponentenmodelle: ActiveX, OLE, CORBA und JavaBeans	45
2.5.3	Die Java-VM für Win32 von Microsoft	46
2.6	Keine visuelle Programmierung	46
<b>3</b>	<b>Ein Blick auf Java</b> .....	<b>49</b>
3.1	FAQ	49
3.2	Ein Java-Applet	50
3.2.1	Die Berechnung der Versandkosten	51
3.2.2	Ein Projekt anlegen	52
3.2.3	Was der Applet Wizard erzeugt hat	64
3.2.4	Die Benutzeroberfläche erstellen	67
3.2.5	Die Programmlogik	75
3.3	Details zur Benutzeroberfläche	81
3.3.1	Windows-Ressourcen	81
3.3.2	Besonderheiten bei Java-Anwendungen	82
3.3.3	Das java.awt-Framework	84
3.3.4	Abbildung von Windows-Ressourcen auf das AWT	88
3.3.5	Die Portabilität des Quellcodes	94
3.3.6	Das Beste aus zwei Welten nutzen	95
3.4	Eine Java-Anwendung	95
3.4.1	Einschränkungen von Java-Applets	95
3.4.2	Anwendungen in Java	96
3.4.3	Technische Aspekte einer Java-Anwendung	98
3.4.4	Eine Java-Anwendung ausführen	102
3.4.5	Ausführbares Java	105
3.5	Die Versand-Anwendung	106
3.5.1	Organisatorische Aspekte einer Java-Anwendung	106
3.5.2	Umbauten am Quellcode	108
3.5.3	Das Menü einbauen	113
3.5.4	Menüauswahlen bearbeiten	116
3.5.5	Das Bestellannahme-Applet	120
3.6	Wiederverwendbare Komponenten	128
3.6.1	Die Anwendung erweitern	129
3.6.2	Die Wareneingangs-Funktion	129
3.6.3	Das Bestellungs-Applet modularisieren	131
3.6.4	Veränderungen konzentrieren	135

3.7	Zurück in die Webseite	136
3.7.1	Das Versandkosten-Applet in einer Webseite ausführen	137
3.7.2	Ein Menü in HTML	138
3.7.3	Die Website testen	143
3.8	Zusammenfassung	144
<b>4</b>	<b>Anwendungslogik in Java</b>	<b>145</b>
4.1	FAQ	145
4.2	Die Versand-Anwendung weiterentwickeln	146
4.3	Grundlagen des OOSE	147
4.3.1	Probleme von herkömmlichen Pflichtenheften	148
4.3.2	OOSE – Object Oriented System Engineering	148
4.3.3	Unterschiedliche OO-Notationen, der Methoden-Krieg und die UML	149
4.3.4	Welche Methode paßt zur UML?	150
4.3.5	Leitsätze beim OO-Design	150
4.4	Komponenten	151
4.4.1	Komponenten definieren	151
4.4.2	Public Interfaces	152
4.5	Die Komponenten in Visual J++ einrichten	152
4.5.1	Komponenten zur Orientierung	152
4.5.2	Mehrere Projekte für eine Anwendung	153
4.5.3	Java-Packages	154
4.5.4	Unterprojekte in Visual J++	156
4.5.5	Packages und Unterprojekte synchronisieren	162
4.6	Das Package VersandLogik entwickeln	166
4.6.1	Auf der Suche nach Objekten	166
4.6.2	Das Objekt Lagerist	169
4.6.3	Das Objekt Bestellung	178
4.6.4	Integrationstest mit Lagerist und Bestellung	188
4.6.5	Exkursion: Ausländische Anschriften und Textpositionen	193
4.7	Verbindung zwischen Logik und Benutzeroberfläche	199
4.7.1	Das Prinzip Modell – View – Controller	199
4.7.2	Die Steuerungs-Schicht	200
4.7.3	Das Package VersandSteuerung	200
4.7.4	Die Methode verarbeite() aufrufen	206
4.8	Zusammenfassung	210
4.8.1	Fehlende Bestandteile	210
<b>5</b>	<b>Lebensräume von Anwendungen</b>	<b>211</b>
5.1	FAQ	211
5.2	Was ist eine Umgebung?	212
5.2.1	Das Programm „Hello World“	213
5.2.2	Binärkompatibilität	214
5.2.3	Quellcode-Kompatibilität	216
5.2.4	Grafische Benutzeroberflächen und die bidirektionale Kommunikation	217

5.3	Verteilte Anwendungen	222
5.3.1	Zentrale Rechner mit Terminals	223
5.3.2	Unabhängige PC mit zentraler Datenhaltung	224
5.3.3	Client-Server-Anwendungen	224
5.3.4	Anwendungen im Internet	225
5.3.5	Die Umgebung anpassen	225
5.4	Das Versandhaus im Internet	226
5.4.1	Look & Feel	227
5.4.2	Die Versand-Website	227
5.4.3	HTML-Formulare	229
5.4.4	Die HTML-Seiten anlegen	232
5.4.5	Java-Verarbeitung auf dem Server	241
5.4.6	Die Umgebung auf einem Netscape-Server	244
5.4.7	Die Versandkosten-Steuerung entwickeln	253
5.4.8	Die Umgebung auf einem Java Web Server	264
5.4.9	Adapter	274
5.5	Zusammenfassung	293
<b>6</b>	<b>Wired Java</b> .....	<b>295</b>
6.1	FAQ	295
6.2	Verteilte Anwendungen	296
6.2.1	Wo wird eine Anwendung ausgeführt?	297
6.2.2	Wie kommt eine Anwendung auf einen Computer?	297
6.2.3	Programme aus dem Netzwerk laden	298
6.2.4	Client-Server-Anwendungen	298
6.2.5	Gründe für den Einsatz verteilter Anwendungen	299
6.3	Grundlagen verteilter Anwendungen	299
6.3.1	Verständigungsprobleme von Objekten	300
6.3.2	Ein lokaler Methoden-Aufruf	300
6.3.3	Ein Methoden-Aufruf über das Netzwerk	301
6.4	Vorbereitungen	302
6.4.1	Das TCP/IP-Protokoll	302
6.4.2	Die notwendige JDK-Version	305
6.4.3	JDK 1.1 in Visual J++ installieren	305
6.4.4	Eine Test-Anwendung	307
6.5	Verteilte Versandkosten-Berechnung	313
6.5.1	Die Intranet-Umgebung	314
6.5.2	Die Server-Komponente	314
6.5.3	Die Client-Komponente	321
6.5.4	Nutzen der Verteilung	327
6.6	Verteilte Bestellannahme	329
6.6.1	Die Lebensdauer von Objekten	329
6.6.2	Das Lagerist-Objekt	329
6.6.3	Die Teilung der Logik	330
6.6.4	Das Remote Interface für das Lagerist-Objekt	331
6.6.5	Die Implementation des LageristRemoteInterface	332
6.6.6	Die Klasse VersandServer übersetzen und testen	335
6.6.7	Das Bestellannahme-Applet anpassen	335

6.6.8	Das Bestellannahme-Applet übersetzen und testen	340
6.6.9	Hintergrund: Laden der .class-Dateien	340
6.7	Die Echtzeit-Anzeige für den Lageristen	341
6.7.1	Die Anzeige für den Lageristen	342
6.7.2	Der Callback-Mechanismus in RMI	342
6.7.3	Das WartendeBestellungen-Applet entwickeln	343
6.7.4	Die Klasse VersandServer erweitern	346
6.7.5	Die Methode registriereClient()	346
6.7.6	Die Aktualisierung	347
6.7.7	Die Lageristen-Anzeige testen	350
6.7.8	Abmelden von Clients	352
6.8	Zusammenfassung	352
6.9	Ausblick	353
	<b>Stichwortverzeichnis</b> .....	<b>355</b>