

Vorwort

Dieses Lehrbuch beschreibt die Methoden der Programmierung – von den Grundlagen bis zu den Strukturierungskonzepten für umfangreiche Programme. Im Vordergrund steht nicht eine bestimmte Programmiersprache, sondern die Erläuterung von Prinzipien wie z.B. Rekursion und Objektorientierung. Durch Diskussion der Möglichkeiten und Grenzen dieser Prinzipien soll ein tiefgehendes Verständnis vermittelt werden.

Damit richtet sich dieses Buch an alle, die sich intensiv mit modernen Methoden der Programmierung beschäftigen möchten. Neben dem Interesse (und vielleicht sogar einem bisschen Spaß) an diesem Thema muss der Leser keine Vorkenntnisse mitbringen. Erfahrungen im Umgang mit Computern sind natürlich nicht hinderlich, aber auch nicht notwendig. Das Durcharbeiten dieses Lehrbuchs, am besten begleitet von selbstständigen Programmierübungen, belohnt den Leser mit umfassenden Kenntnissen und einem guten Beurteilungsvermögen von Programmiertechniken. Als nächster Schritt kann sich das Thema Software Engineering anschließen, das sich den Arbeitsvorgängen des Softwareentwurfs widmet – von der Erfassung der Anforderungen bis zur Pflege des fertigen Softwareprodukts.

Wer einen schnellen, aber nur oberflächlichen Einstieg in die Programmierung sucht oder sich nur für ein Nachschlagewerk der Konstrukte einer Programmiersprache interessiert, mag mit anderen Werken besser bedient sein. Hier geht es darum, das gesamte Spektrum abzudecken, das sich vom Erlernen von Konzepten über ihre Anwendung in Programmen bis hin zur kritischen Beurteilung erstreckt. Dieses Buch kann daher Studenten und allen anderen empfohlen werden, die „der Sache auf den Grund gehen“ möchten.

Das Programmieren lässt sich nur anhand einer Programmiersprache sinnvoll erklären. Hier wird die neuere Sprache Java verwendet, die alle relevanten Konzepte von modernen imperativen Programmiersprachen einschließt und zunehmend Verbreitung findet. Java wird aber nur als Instrument benutzt, um die Konzepte, um die es eigentlich geht, darzulegen. Dabei treten neben vielen Vorteilen auch einige Nachteile von Java zu Tage.

Der Aufbau des Buches orientiert sich an der Informatik-Grundausbildung. Zunächst werden wichtige Grundbegriffe wie Algorithmus, Syntax oder Grammatik in einem kurzen Abriss dargestellt. Daran schließen sich die ersten Schritte

zur Programmierung mittels primitiver Datentypen und einfacher Anweisungen an. Die wichtigen Themen der objektorientierten Programmierung bestimmen dann den Rest des Buches: Klassendefinitionen, Objekte, Methoden, Konstrukto-ren usw. Damit wird auch ein Fundament geschaffen, auf dem sich typische, häufig benötigte Elemente des Programmierens erklären lassen, beispielsweise die Rekursion von Methoden und Datenstrukturen. Das wiederum liefert die Basis, um nützliche Strukturen wie Stapel, Listen, Bäume und Graphen einzuführen.

Ein tiefergehendes Konzept der Objektorientierung ist die Spezialisierung von Klassen – in vielen Sprachen auch Vererbung genannt. In diesem Buch wird nicht nur der Erweiterungsmechanismus von Java vorgestellt, sondern auch großes Gewicht auf die Darstellung gelegt, wie er sinnvoll einzusetzen ist und welche Alternativen existieren, z.B. die Spezialisierung durch Delegation. Das Gleiche gilt für die Definition von Schnittstellen (Interfaces sind eine Spezialität von Java). Es wird ausführlich diskutiert, wie Schnittstellen verwendet werden können, um wieder verwendbare Softwarekomponenten zu erhalten.

Die Vorstellung besonderer Programmierkonzepte wie Pakete, Ausnahmen und Threads rundet die Darstellung ab.

Das Buch schließt mit einer Kurzanleitung für das Erstellen von kleineren bis mittelgroßen Softwareprojekten. Das erworbene Wissen wird damit in den Rahmen des Softwareentwicklungs-Prozesses gestellt. Große, anspruchsvolle Projekte erfordern allerdings einen vertieften Einstieg in das Gebiet Software Engineering, das nicht Gegenstand dieses Buches ist.

Zu allen Kapiteln werden zahlreiche Beispielprogramme angegeben. Ein größeres Beispiel, ein Ausschnitt aus dem Mahnwesen einer Bibliothek, wird an verschiedenen Stellen in diesem Buch aufgegriffen und ist im Anhang A zusammenhängend dargestellt. Die Autoren haben die angegebenen Beispiele sorgfältig programmiert und getestet. Das schließt aber nicht aus, dass sich trotzdem an der einen oder anderen Stelle ein Fehler eingeschlichen haben könnte. Auch konnten nicht alle verfügbaren Java-Systeme und Plattformen ausprobiert werden. Leider hat die Erfahrung gezeigt, dass Programme in den verschiedenen Rechnerwelten (meist nur geringfügig) unterschiedliches Verhalten zeigen können.

Dem Leser wird empfohlen, die Beispielprogramme nachzuvollziehen, um die Anwendung der Programmierkonzepte besser zu verstehen. Darüber hinaus ist auch das selbstständige Erstellen von Programmen nützlich. Im Text selbst sind zwar keine Übungsaufgaben enthalten. Diese sind im World-Wide Web unter www.informatik.uni-essen.de/Prog/ zu finden.

In diesem Buch werden die verbalen Beschreibungen durch zahlreiche grafische Darstellungen ergänzt. Dabei werden (etwas entgegen dem Trend der Zeit) nicht die Darstellungsmittel von UML gewählt. Die verschiedenen Diagrammformen von UML eignen sich nicht so gut als Erklärungshilfsmittel und weisen nicht immer die von den Autoren gewünschte Detailtiefe auf. Die gewählten Grafiken

sind aber nicht so weit von UML entfernt, dass ein Umstieg und die Einarbeitung in UML-Werkzeuge schwer fielen.

Die Autoren hoffen, dem Leser die Methoden der Programmierung gut verständlich und in sich geschlossen dargestellt zu haben und ihm ein nützliches Fundament für seine weitere Beschäftigung mit der Programmierung zu bieten.

Allen, die uns geholfen haben, danken wir für ihre Unterstützung.

Essen im März 2000
Klaus Echte und Michael Goedicke

Inhaltsverzeichnis

1	Grundbegriffe der Informatik	1
1.1	Motivation	1
1.2	Problemlösung durch Methoden und Maschinen der Informatik	3
1.3	Verarbeitung von Information durch einen Algorithmus	7
1.4	Formale Sprache und Grammatik	11
1.5	Programm	17
2	Grundelemente der Programmierung	21
2.1	Erste Schritte	21
2.2	Primitive Datentypen	30
2.3	Anweisungen	43
2.4	Arrays	65
3	Objekte und Klassen	83
3.1	Grundzüge der Objektorientierung	83
3.2	Verweisvariablen und Zugriffe auf Objekte	89
3.3	Methoden und ihre Parameter	98
3.4	Konstruktoren	107
3.5	Gültigkeitsbereich von Bezeichnern	114
4	Rekursion	121
4.1	Beschreibungen mit Selbstbezug	121
4.2	Rekursive Algorithmen	127
4.3	Rekursive Datenstrukturen	134
4.4	Arten rekursiver Beschreibungen	158
5	Nützliche Datenstrukturen	165
5.1	Zeichenkette	165
5.2	Puffer und Stapel	169
5.3	Suchbaum	172
5.4	Hashtabelle	179
5.5	Gerichteter Graph	185

6	Erweiterung von Klassen	193
6.1	Erweiterung einer Klassenimplementierung	193
6.2	Erzeugung von Objekten bei erweiterten Klassendefinitionen	205
6.3	Verdecken von Variablen und Überschreiben von Methoden	212
6.4	Vererbungshierarchien	218
6.5	Verwendung der Erweiterungsmechanismen: Das Bibliotheksbeispiel	224
6.6	Anonyme Erweiterungen von Klassen	231
6.7	Beziehungen zwischen Klassen	236
7	Flexible Softwarekomponenten: Generische Objektstrukturen	243
7.1	Verwendung von Programmteilen für ähnliche Aufgaben	243
7.2	Abstrakte Klassen	250
7.3	Definition von Schnittstellen	256
7.4	Verwendung von Schnittstellen	263
8	Spezielle Konzepte der Programmierung	277
8.1	Pakete	277
8.2	Ausnahmen	291
8.3	Threads	302
9	Eine Kurzanleitung zur Softwareentwicklung	313
9.1	Zweck dieser Kurzanleitung	313
9.2	Stadien der Softwareentwicklung	314
9.3	Die Rolle von Werkzeugen	326
9.4	Bewertung und Ausblick	327
A	Anhang: Beispiel eines Softwareentwurfs	329
B	Anhang: Der EinAusRahmen	355
	Literatur	361
	Index	363