

Elvira Preiwisch-Seibold

**Das Erlernen einer Programmiersprache  
im Wirtschaftsinformatikunterricht  
unter besonderer Berücksichtigung  
wissenspsychologischer Theorien,  
dargestellt am Beispiel  
der prozeduralen Sprache COBOL**

Herbert Utz Verlag  
Wissenschaft  
München  
1984  
120 Seiten  
ISBN 3 7089 0123 4



Herbert Utz Verlag · Wissenschaft  
München

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

**Preiwisch-Seibold, Elvira:**

Das Erlernen einer Programmiersprache im  
Wirtschaftsinformatikunterricht unter besonderer Berücksichtigung  
wissenspsychologischer Theorien, dargestellt am Beispiel  
der prozeduralen Sprache COBOL /  
Elvira Preiwisch-Seibold. - München : Utz, Wiss., 1999

(Pädagogik)

Zugl.: Hamburg, Univ., Diss., 1998

ISBN 3-89675-553-6

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Wiedergabe auf photomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei nur auszugsweiser Verwendung, vorbehalten.

Copyright © Herbert Utz Verlag GmbH 1999

ISBN 3-89675-553-6

Printed in Germany

Herbert Utz Verlag GmbH, München

Tel.: 089/277791-00 - Fax: 089/277791-01

<b>Problemaufriß und Überblick</b>	<b>1</b>
<b>Kapitel I:</b>	
<b>Zur Fragestellung der Untersuchung</b>	<b>8</b>
1. Entwicklung eines kategorialen Rahmens für pädagogische Fragen zum Programmierunterricht	9
1.1 Didaktische und fachdidaktische Konzeptionen	9
1.2 Ganzheitlichkeit als berufs- und wirtschaftspädagogische Kategorie	19
2. Ziele des Programmierunterrichts im Fach Wirtschaftsinformatik	27
2.1 Bildungsziele und fachspezifische Richtziele	27
2.2 Zu den Begriffen der informationstechnischen und informatischen Bildung	30
2.3 Lehrplandeterminanten	32
3. Beschreibung des Lehrgegenstandes	34
3.1 Philosophie der Sprache COBOL	34
3.2 COBOL als Unterrichtssprache	37
4. Spezifische Problemfelder des COBOL-Programmierunterrichts	38
4.1 Die Wissensvernetzung	38
4.2 Das Denken in bildlichen Strukturen	39
4.3 Das problemlösende Denken	39
5. Schlußfolgerungen für den COBOL-Programmierunterricht aus bildungstheoretischer, didaktischer und curricularer Sicht	40

<b>Kapitel II:</b>	
<b>Problemlage im Kontext pädagogisch- psychologischer Literatur</b>	<b>43</b>
1. Die Wissensvernetzung	44
1.1 Kognitionspsychologische Ansätze zur Wissensvernetzung	46
1.1.1 Vernetzung durch Schemabildung	46
1.1.2 Vernetzung durch Prozeduralisierung	48
1.2 Ansätze der Instructional Design-Forschung	51
1.2.1 Mikro- und Makrostrukturen unter dem Gesichtspunkt der Sequenzierung	52
1.2.2 Gestaltung von Hypertextsystemen nach 'ID'-Kriterien	53
1.3 Systemdynamisches Modellieren	57
1.4 Zusammenfassende Betrachtung	58
2. Das Denken in bildlichen Strukturen	60
2.1 Grundlagen der Veranschaulichung	60
2.2 Kognitionspsychologische Erkenntnisse zur Bildung mentaler Modelle	61
2.2.1 Das Verstehen von Graphiken	61
2.2.2 Das Bilden mentaler Modelle	62
2.3 Präskriptive Regeln einer Grammatik für Gra- phiken und bildliche Darstellungen im Hypertext im Rahmen der Instructional Design-Forschung	65
2.4 Zusammenfassende Betrachtung	67

3.	Das problemlösende Denken	69
3.1	Kognitionspsychologische Ansätze zum Problemlösen	72
3.1.1	Problemrepräsentation nach dem Expertenparadigma	73
3.1.2	Problemlösen durch abstrakte Kategorien	74
3.1.3	Problemlösestrategien	81
3.2	Ansätze der Instructional Design-Forschung	87
3.2.1	Elementarisierung	87
3.2.2	Algo-heuristische Strukturen	88
3.3	Zusammenfassende Betrachtung	89
4.	Bewertung der pädagogisch-psychologischen Erkenntnisse	90

## **Kapitel III:**

### **Lösungsansätze für den COBOL-Pro- grammierunterricht**

		96
1.	Überlegungen zur Ausgangssituation	96
2.	Vorschlag zu einem Unterrichtsmodell	98
2.1	Unterrichtsvorschläge zur Wissensvernetzung	101
2.1.1	Schemabildender Unterricht und Prozeduralisierung	101
2.1.2	Sequenzierung von Unterrichtsstrukturen	106
2.1.3	Vernetzung durch Hypertext	110
2.2	Förderung des Denkens in bildlichen Strukturen	113
2.2.1	COBOL-Graphiken im Unterrichtseinsatz	113
2.2.2	Die vierstufige Schemaentwicklung	123
2.3	Unterrichtsvorschläge zum problemlösenden Denken	124
2.3.1	Kategorienbildung	124
2.3.2	Strategien zur Problemlösung	130
2.3.3	Beispiele zu algo-heuristischen Strukturen	136

## **Kapitel IV:**

### **Evaluation der Lösungsansätze** 145

#### 1. Empirische Untersuchungsreihen 145

1.1 Spezifische Fragestellungen zur empirischen Evaluation 145

1.2 Methoden und Grenzen der Untersuchung 145

1.3 Durchführung der multivariaten Varianzanalyse 149

1.4 Auswertung 157

#### 2. Ergebnisberichte 159

2.1 Spezifische Fragestellungen zur Evaluation durch subjektive Einschätzungen 159

2.2 Subjektive Unterrichtsbeobachtungen 159

2.3 Protokolle zum Hypertexteinsatz 162

2.3.1 Design, Stichprobe und Auswertungsmodus 162

2.3.2 Ergebnisse 163

2.3.3 Auswertung 166

2.4 Fragebogenuntersuchung 168

2.4.1 Befragungen zur Kategorisierung 168

2.4.2 Befragungen zum Einsatz der Graphiken 171

2.5 Auswertung der Ergebnisse 172

#### 3. Bewertung der Ergebnisse 173

### **Zusammenfassung und Ausblick** 179

Literaturverzeichnis 187

Abbildungsverzeichnis 206

Tabellenverzeichnis 208

Anhang

## *Problemaufriß und Überblick*

Die revolutionären Entwicklungen im Programmiersprachenbereich, seien es das 'visuelle Programmieren' mit 'Visual Basic' als dem populärsten Vertreter mit visuellem Oberflächeneditor<sup>1</sup> oder die graphischen Benutzeroberflächen, haben in den letzten Jahren die Anwendung und den Umgang mit Hard- und Software für den Benutzer erheblich erleichtert und den Zugang zu immer mehr EDV-Anwendern gefunden. Selbst im professionellen Bereich hat sich das "benutzerfreundliche" Programmieren durchgesetzt. Vor diesem Hintergrund muß der Untersuchungsgegenstand der vorliegenden Arbeit, die Programmiersprache COBOL im Schulunterricht, gesehen werden.

Die prozedurale Sprache COBOL, die für Großrechenanlagen konzipiert wurde und deshalb nicht als benutzerfreundlich nach modernsten Erfordernissen angesehen werden kann, erfordert umfangreiches Syntax- und Semantikwissen. Ihre Struktur ist durch starre Schemata und vielzählige Definitionszeilen bestimmt. Erfahrungen aus eigener Unterrichtspraxis zeigen, daß diese besonderen Begebenheiten Programmiernovizen das Erlernen der Sprache COBOL erschweren. Lernschwierigkeiten und Lernblockaden treten im Unterricht vermehrt auf. Bei der Ursachenforschung zeigt sich, daß diesen Schwierigkeiten nicht allein mit Fachwissen begegnet werden kann.

**Das Ziel der vorliegenden Arbeit besteht darin, Lösungsansätze zur Bewältigung typischer Lernschwierigkeiten zu entwickeln.**

Typische Problemfelder des Programmierunterrichts werden somit exemplifiziert und unter dem Blickwinkel der Kognitionspsychologie und ausgesuchter 'Instructional Design'-Theorien betrachtet. Die daraus resultierenden Erkenntnisse, auf ihren Einsatz zur Verbesserung des Programmierunterrichtes geprüft, ergänzt durch Unterrichtserfahrungen, -beobachtungen und -analysen, führen in ihrer Anwendung zu Unterrichtsvorschlägen für die Praxis. Den Abschluß bildet eine Evaluation des vorgestellten 'Konzepts' anhand empirischer Daten.

---

<sup>1</sup> vgl. Instant-Software Visuell. In: c't 4/95, S. 115.

## *Untersuchungsrahmen*

Den Hintergrund dieser Arbeit bildet eine achtjährige Unterrichtspraxis an der Berufsoberschule, Fachrichtung Wirtschaft. In dieser Zeit wurde ein 'Konzept' zur Unterrichtsgestaltung beim Programmiersprachenerwerb entwickelt. In jedem Schuljahr wurde mit zwei Klassen der gleichen Jahrgangsstufe die Effektivität der ausgearbeiteten Strategien erprobt, Zufallsergebnisse sollten so minimalisiert werden. Bewährte Methoden des 'Konzeptes' wurden im nächsten Schuljahr fortgesetzt. Schülerbefragungen, Beobachtungen über Mitarbeit und Motivation sowie Fehleranalysen und Notenevaluationen der schriftlichen Arbeiten halfen, die Unterrichtsvorschläge zu verbessern. Im vergangenen Schuljahr wurde eine zweite Lehrkraft in die Untersuchungen einbezogen, die nach den gleichen Kriterien unterrichtete und mit drei Klassen in der zweiten Jahrgangsstufe eine Evaluation durchführen konnte.

## *Zielgruppe*

Zielgruppe dieser Untersuchungen bilden die Schüler der eingangs erwähnten Berufsoberschule. Diese berufliche Schule unterscheidet zwei Jahrgangsstufen und bietet folgende Bildungsabschlüsse an: die fachgebundene Hochschulreife und nach erfolgreicher Prüfung in einer zweiten Fremdsprache die allgemeine Hochschulreife. Derzeit wird ein Schulversuch zur dreistufigen Berufsoberschule an der Berufsoberschule Wirtschaft in München durchgeführt: Er sieht zusätzlich den Erwerb des mittleren Bildungsabschlusses nach Beendigung der Stufe 1 vor. Die Fachhochschulreife wird nach erfolgreich abgelegter Prüfung in der Stufe 2 erworben und schließlich die Hochschulreife (allgemein bzw. fachgebunden) nach Stufe 3. Die Schüler dieses Schulversuches wurden nicht in die Zielgruppe miteinbezogen, da noch nicht ausreichende Unterrichtserfahrungen zu dieser Schulform vorlagen.

Die Schüler der herkömmlichen Berufsoberschule, Zielgruppe der empirischen Erhebungen, rekrutieren sich aus Absolventen einer kaufmännischen Lehre mit mittlerem Bildungsabschluß. Laut curricularen Lehrplänen der kaufmännischen Berufsschulen sollen alle Schüler Grundkenntnisse der EDV nach Abschluß der Prüfung erworben haben. Ein Großteil der Berufsoberschüler besitzt bei Beginn des Informatikkurses rudimentäre Kenntnisse in der theoretischen Informatik und hat einige Anwendungssoftwarepakete oberflächlich kennengelernt. Die Programmier-



erfahrungen sind gering und beinhalten Kenntnisse in den prozeduralen Sprachen BASIC oder PASCAL. Manche Schüler können jedoch durchaus als ausgesprochene Programmierexperten bezeichnet werden, die bereits als Programmierer in den Sprachen COBOL, C, ASSEMBLER, etc. in ihren Ausbildungsfirmen arbeiteten und zum Teil Netzwerksysteme aufbauten und verwalteten. Die Spannweite der Vorkenntnisse umfaßt somit sowohl Anfänger, die zahlenmäßig die größte Gruppe repräsentieren, Fortgeschrittene und Experten.

## *Problemfelder*

Die in dieser Arbeit untersuchte Programmiersprache COBOL zählt zu den imperativen (oder prozeduralen) Sprachen, die aufgrund komplizierter Syntax und umfangreicher Semantik große kognitive Anforderungen an den Lernenden stellt. Das notwendige Syntax- und Konzeptwissen ist bald nicht mehr beherrschbar: Resignation kann auftreten - der Anschluß an die Lerngeschwindigkeit der übrigen Klasse ist in Frage gestellt. Ursachen hierfür stellen die **Schwierigkeiten beim Verbinden der Fakten, beim Herstellen von Beziehungen und beim Zusammenfügen wichtiger Bausteine** dar.

Die Syntax einer prozeduralen Sprache ist für das Erlernen einer Erstsprache anfänglich sehr abstrakt. Verständnisbarrieren entstehen und lassen sich durch verbale Erklärungen des Lehrenden oft nicht abbauen. Die Ursache hierfür liegt vielfach in der mangelnden Fähigkeit, mentale Vorstellungen, d.h. **mentale Modelle** von den prozeduralen Abläufen zu bilden.

Ist das Syntax- und Semantikwissen in Form von Befehlsformaten vermittelt und von der Lernenden verstanden, bleibt immer noch ein großes Problemfeld: Die Umsetzung gelernter deklarativen<sup>2</sup> und prozeduralen Wissens<sup>3</sup> in konkrete Anwendungen, die beim Programmieren das **Lösen eines Problems** durch Generieren eines Quellcodes umfassen. Das im Unterricht vorgestellte und erarbeitete Programm wird häufig nur memorierend aufgenommen und nicht in seiner Zusammensetzung und Struktur kognitiv erfaßt - so fällt es Schülern schwer, das Gelernte in konkrete Problemlösungsprozesse umzusetzen.

---

<sup>2</sup> Faktenwissen.

<sup>3</sup> Regelwissen; siehe Erläuterungen in Kapitel II, Abschnitt 1.1.2.