



Daniel Thalmann

MODULA-2

An Introduction

With 179 Figures

Springer-Verlag
Berlin Heidelberg New York Tokyo

Daniel Thalmann
Université de Montréal, Département d'Informatique
et Recherche Opérationnelle
C. P. 6128, Montréal (Quebec) H3C 3J7, Canada

Library of Congress Cataloging in Publication Data

Thalmann, Daniel.

MODULA-2: an introduction.

Bibliography: p.

1. Modula-2 (Computer program language) I. Title.

QA76.73.M63T43 1985 001.64'24 85-9758

ISBN-13: 978-3-540-13297-4 e-ISBN-13: 978-3-642-69668-8

DOI: 10.1007/978-3-642-69668-8

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks. Under § 54 of the German Copyright Law where copies are made for other than private use a fee is payable to "Verwertungsgesellschaft Wort", Munich.

©Springer-Verlag Berlin Heidelberg 1985

2145/3140-543210

Foreword

MODULA-2 is a new programming language which was created by Niklaus Wirth of the Swiss Federal Institute of Technology (ETH) in Zurich. The language is derived from PASCAL: it includes all aspects of PASCAL and sometimes improves on them. Moreover, MODULA-2 includes the important “module” concept, as well as multiprogramming capabilities and a way of implementing low-level software in an elegant manner. In summary, MODULA-2 may be used equally well as a general-purpose programming language and as a system implementation language. MODULA-2 provides the programmer with a good way of writing high quality software. In particular, modules are powerful tools for achieving modularity, reliability, readability, extensibility, reusability and machine-independence.

This book presents the complete MODULA-2 language from the beginning. Each topic is presented by means of numerous examples and each concept is justified. The syntax of the language is explained using syntactic diagrams.

This book is not a reference manual for MODULA-2, but a textbook from which the student can learn the language progressively. The most important concepts (i. e. procedures, modules and data structures) are explained in great detail and methodological aspects are also emphasized.

Beginning in the first chapter, the student may execute his/her own programs. Program examples in this book have been executed on several machines (APPLE II, IBM PC and VAX 11/780) and they may be taken as a basis for students.

At the end of each chapter, a summary is provided to remind the reader of the most important topics discussed.

Finally, there are two appendices: the first describes the complete syntax using syntactic diagrams and extended Backus-Naur forms, and the second lists the keywords of MODULA-2 and its standard modules.

The author would like to thank Luigi Zaffalon of the “Ecole d’Ingénieurs de Genève” for carefully reading and commenting on the manuscript; he also provided several program examples. The author is also grateful to Ann Laporte, who improved the English text. Finally, the author thanks Evelyne Kohl for typing the manuscript.

Table of Contents

1 Introduction	1
1.1 The Role and Origins of Programming	1
1.2 The Evolution of Programming Languages	1
1.3 The “Module” Concept	4
1.4 Lexical, Syntactic and Semantic Analysis	4
1.5 The MODULA-2 Vocabulary	5
1.6 Syntactic Diagrams	6
1.7 Structure of a MODULA-2 Program	7
1.8 A First Program: How to Write a String	9
1.9 Some Basic Considerations for Writing Programs	9
Summary	10
2 Numbers	11
2.1 Integer and Real Numbers	11
2.2 Operators	11
2.3 The MathLib0 Module	13
2.4 Arithmetic Expressions	15
2.5 How to Write Numbers	16
2.6 Program Examples	17
Summary	18
3 Constants, Variables and Data Types	19
3.1 Constants	19
3.2 Variables and Data Types	20
3.3 INTEGER, CARDINAL and REAL Types	21
3.4 The Assignment Statement	22
3.5 Type Conversion	23
3.6 An Example: Exchange of Two Values	25
Summary	26
4 The Boolean and CHAR Types	27
4.1 Boolean Constants and Operations	27
4.2 Boolean Expressions	27
4.3 Relational Operators	30
4.4 Two Program Examples	33
4.5 The Type CHAR	33
Summary	39

- 5 Basic Control Statements 40**
 - 5.1 The IF Statement 41
 - 5.2 The WHILE Statement 44
 - 5.3 The REPEAT Statement 48
 - 5.4 Program Examples 51
 - Summary 53

- 6 The Input/Output Modules 54**
 - 6.1 The Role of the Read Operation 54
 - 6.2 Text Input 56
 - 6.3 The InOut and RealInOut Modules 59
 - 6.4 Program Examples 63
 - Summary 63

- 7 Enumeration and Subrange Types 64**
 - 7.1 Enumeration Types 64
 - 7.2 Order in Enumeration Types 66
 - 7.3 Subrange Types 68
 - 7.4 A Program Example 71
 - Summary 71

- 8 Other Control Statements 72**
 - 8.1 The LOOP Statement 72
 - 8.2 The FOR Statement 75
 - 8.3 The CASE Statement 79
 - 8.4 Program Examples 84
 - Summary 85

- 9 Array Types 86**
 - 9.1 One-Dimensional Arrays 86
 - 9.2 Program Examples 91
 - 9.3 String Types 92
 - 9.4 Program Examples: Palindromes 95
 - 9.5 Multi-Dimensional Arrays 95
 - 9.6 Program Examples 101
 - Summary 101

- 10 Procedures and Locality 102**
 - 10.1 The Procedure Concept 102
 - 10.2 Locality 104
 - 10.3 Parameters 113
 - 10.4 A Program Example 121
 - Summary 121

11 Transmission of Parameters and Scope	122
11.1 Variable and Value Parameters	122
11.2 Open Array Parameters	125
11.3 Rules of Scope	127
Summary	131
12 Function Procedures and Procedure Types	132
12.1 The Function Procedure Concept	132
12.2 A Complete Example	137
12.3 Procedure Types	137
12.4 A Complete Example	142
Summary	144
13 Recursion	145
13.1 The Concept of Recursion	145
13.2 The Towers of Hanoi	147
13.3 Quicksort and Permutations	150
13.4 Indirect Recursion and Recursive Descent	151
Summary	158
14 Modules	159
14.1 The Module Concept	159
14.2 Definition Modules	160
14.3 Implementation Modules	162
14.4 Separate Compilation	163
14.5 A Complete Example: The Stack	165
Summary	167
15 Local Modules	169
15.1 The Concept of Local Modules	169
15.2 Import/Export Lists	170
15.3 Rules of Scope	171
15.4 Local Modules Defined Side by Side	173
15.5 Modules Defined Local to a Procedure	174
15.6 A Complete Example	179
Summary	180
16 Records	181
16.1 RECORD Types	181
16.2 Manipulation of Records and Fields	183
16.3 The WITH Statement	186
16.4 An Example of Data Abstraction: The Use of Multiple Stacks	188
16.5 Records with Variant Parts	191
Summary	196

17 Sets	197
17.1 SET Types and the BITSET Type	197
17.2 Set Constants and Variables	198
17.3 Set Operators	200
17.4 The Standard Procedures INCL and EXCL	204
17.5 An Example of a Module for Processing Bitsets	205
Summary	209
18 Streams	210
18.1 The Stream Concept	210
18.2 Access to a Stream Component	211
18.3 Text Streams	211
18.4 A Program Example: Bold Text	216
Summary	216
19 Dynamic Data Structures	217
19.1 Pointers	217
19.2 Dynamic Allocation	218
19.3 Lists	221
19.4 A Complete Example of List Processing with a Local Module	226
19.5 Stacks and fifo Queues: Two List Processing Modules	226
19.6 Binary Trees	234
Summary	241
20 Processes	243
20.1 Concurrency	243
20.2 Processes	244
20.3 Monitors	244
20.4 Signals	245
20.5 A Classical Example: A Shared Buffer	249
Summary	249
21 Low-Level Features	250
21.1 Why Low-Level Programming?	250
21.2 The Type WORD	250
21.3 The Type ADDRESS	253
21.4 Type Transfer Functions	255
21.5 Coroutines	256
21.6 An Application of Coroutines	259
21.7 Dynamic Storage	259
21.8 Device Handling, Interrupts and Fixed Addresses	260
21.9 Low-Level File Operations	262
21.10 Generic Modules	264
Summary	268

Table of Contents	XI
References	269
Appendix 1: The Syntax of MODULA-2	270
Appendix 2: Keywords and Standard Identifiers	291