

Table of Contents

1 Algorithms	1
1.1 Examples of Algorithms	1
1.2 Definition of the Term "Algorithm"	5
1.3 Objects and Actions	8
1.4 Means of Representing Algorithms	15
1.5 Some Simple Algorithms	22
1.6 Algorithms and Programs	28
Exercises	30
2 Principles of Program Development	33
2.1 The Principle of Stepwise Refinement	33
2.2 Data Capsules	41
2.3 The Module Concept	45
Exercises	48
3 The Programming Language Modula-2	50
3.1 Notation for Language Description	50
3.2 Lexical Elements of Modula-2	52
3.2.1 Identifiers	52
3.2.2 Key Words	53
3.2.3 Numbers	54
3.2.4 Character Strings	56
3.2.5 Operators and Delimiters	58
3.2.6 Separation of Symbols	59
3.2.7 Comments	59
Exercises	60
3.3 Elementary Program Structure	61
3.4 Declarations	64
3.4.1 Data Types	64
3.4.1.1 Predefined Data Types	64
3.4.1.2 Elementary User-Defined Data Types	69
3.4.1.3 Structured Data Types	70
3.4.2 Declaration of Constants	77

3.4.3	Declaration of Types	78
3.4.4	Declaration of Variables	79
3.4.5	Type Identity Among Objects	80
	Exercises	81
3.5	Expressions	82
3.5.1	Classes of Expressions	82
3.5.2	Operands	82
3.5.3	Operators	84
3.5.4	Rules for Writing Expressions	85
3.5.5	Expression Compatibility of Operands	89
	Exercises	89
3.6	Statements	90
3.6.1	Assignment	92
3.6.2	Conditional Statements	93
	3.6.2.1 The IF Statement	93
	3.6.2.2 The CASE Statement	95
	Exercises	98
3.6.3	Loops	98
	3.6.3.1 The WHILE Statement	99
	3.6.3.2 The REPEAT Statement	100
	3.6.3.3 The FOR Statement	102
	3.6.3.4 The LOOP and EXIT Statements	104
	Exercises	106
3.6.4	The WITH Statement	106
	Exercises	108
3.7	Procedures	109
3.7.1	Declaration of Procedures	109
3.7.2	Invocation of Procedures	112
3.7.3	The RETURN Statement	114
3.7.4	Function Procedures	116
3.7.5	Scope and Lifetime of Objects in Procedures	118
3.7.6	Data Exchange Between Procedures	120
3.7.7	ARRAY Parameters	123
3.7.8	Standard Procedures	124
3.7.9	PROCEDURE Types and Procedure Variables	129
	Exercises	133
3.8	The Module Concept of Modula-2	134
3.8.1	Local Modules	134
3.8.2	Scope and Lifetime of Objects in Modules	138
3.8.3	Modules and Separate Compilation	139
	3.8.3.1 Definition Modules	140
	3.8.3.2 Implementation Modules	143
	Exercises	145
3.9	System-Dependent Language Properties	145
3.9.1	The Module SYSTEM	145
3.9.2	Type Transfer Functions	150
3.9.3	Absolute Addressing of Variables	152

3.10 Processes and Coroutines	153
3.10.1 Creation and Synchronization of Processes	154
3.10.2 Interrupts and Priorities	159
4 Writing Modula-2 Programs	163
4.1 Some Simple Modula-2 Programs	163
4.2 Input/Output	169
4.2.1 The Module Terminal	171
4.2.2 The Modules InOut and ReaInOut	172
4.2.3 The Module FileSystem	177
4.3 Dynamic Data Structures	181
4.3.1 Working with Dynamic Data Structures	182
4.3.2 Dynamically Created RECORDs with Variants	187
4.4 Recursion	188
4.4.1 Recursive Procedures	189
4.4.2 Inner Structure of Recursive Procedures and How They Work	191
4.4.3 Some Examples of Recursive Algorithms	192
4.4.4 Advantages and Disadvantages of Recursive Programming	195
4.5 Language Extension Modules	195
4.5.1 The Module MathLib0	196
4.5.2 The Module String	197
4.5.3 The Module Storage	198
4.6 Parallel Processes	199
Exercises	209
5 Programming Style and Program Testing	211
5.1 Programming Style	211
5.1.1 Structure	212
5.1.2 Naming Conventions	217
5.1.3 Commentary	217
5.1.4 Outward Form of Programs	218
5.2 Program Testing	219
5.2.1 Types of Error	219
5.2.2 The Testing Procedure	220
5.2.3 Built-In Testing and Debugging Aids	221
5.2.4 Debuggers	223
6 Software Engineering with Modula-2	224
6.1 Modula-2 as a Tool for Specification	224
6.2 Modularizing and Structuring	225
6.3 Data Capsules and Data Abstraction	226
6.4 Separate Compilation and Type Binding	230

Appendix A: Syntax Diagrams	234
Appendix B: Predefined Identifiers	247
Appendix C: Compatibility Rules	248
Appendix D: ASCII Table	252
References	253
Index	254