

9361025

11/27/89
2759
v. 3
C. Lins

The Modula-2 Software Component Library

Volume 3

With 30 Illustrations



E9361025



Springer-Verlag
World Publishing Corp

Charles Lins, Apple Computer, Inc., Cupertino, CA 95014, USA

Editors

Steven S. Muchnick, SUN Microsystems, Inc., Mountain View, CA 94043, USA

Peter Schnupp, InterFace Computer GmbH, D-8000 München 81, West Germany

All of the Modula-2 software in this book was written by C. Lins. Copyright © 1989 by Charles A. Lins. MacMETH, by Werner Heiz, Neugasse 71, CH-8005, Zürich, Switzerland. Copyright © 1986, Institut für Informatik Eidgenössische Technische Hochschule (ETH) Zürich, Clausiusstr. 55, CH-8096 Zürich, Switzerland.

TML Modula-2 Compiler, by Robert R. Campbell. Copyright © 1987, 1988 Robert R. Campbell.

Ada is a registered trademark of the U.S. Government (Ada Joint Program Office).

Apple, the Apple logo, AppleTalk, LaserWriter, and Lisa are registered trademarks of Apple Computer, Inc.

ImageWriter, MacDraw, Macintosh, MacPaint, and MacWrite are trademarks of Apple Computer, Inc.

Motorola is a trademark of Motorola, Inc.

UNIX is a trademark of AT&T Bell Laboratories.

Library of Congress Cataloging-in-Publication Data

Lins, C. (Charles)

The Modula-2 software component library

(Springer compass international)

Bibliography: v. 1, p.

1. Modula-2 (Computer program language)

I. Title. II. Series QA76.73.M63L56 1989

005.13'3 89-24956

© 1989 by Springer-Verlag New York Inc.

All rights reserved. This work may not be translated or copied in whole or in part without written permission of the publisher (Springer-Verlag, 175 Fifth Avenue, New York, New York, 10010, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use of general descriptive names, trade names, trademarks, etc., in this publication, even if the former are not explicitly identified, is not to be taken as a sign that such names, as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Reprinted by World Publishing Corporation, Beijing, 1992

for distribution and sale in The People's Republic of China only

ISBN 7-5062-1272-2

ISBN 0-387-97074-6 Springer-Verlag
ISBN 3-540-97074-6 Springer-Verlag

New York Berlin Heidelberg
Berlin Heidelberg New York

Dedicated to Doris Regina
whose love made this possible

and
to Robert, Lori, and Heather

and
to Modula-2 programmers everywhere

Contents

0	Introduction to Volume 3	1
---	--------------------------	---

Part 1 — Preliminaries

1	Specification	5
---	---------------	---

1.1	Specification of Procedure Abstractions	5
1.1.1	Header Section	5
1.1.2	Requires Section	6
1.1.3	Where Section	6
1.1.4	Modifies Section	6
1.1.5	Effects Section	6
1.1.6	Signals Section	6
1.2	Specification of Data Abstractions	7
1.3	Special Symbols	7
	References	8

2	Module Guide	9
---	--------------	---

2.1	Purpose	9
2.2	Characterization of Modules	9
2.3	Module Guide Organization	11
2.4	Binary Trees	12
2.4.1	Tree Types	13
2.4.2	AVL Tree — Sequential Unbounded Managed Iterator	13
2.4.3	BB Tree — Sequential Unbounded Managed Iterator	13
2.4.4	Binary Tree — Sequential Bounded Managed Iterator	14
2.4.5	Binary Tree — Sequential Unbounded Managed Iterator	15
2.4.6	IPB Tree — Sequential Unbounded Managed Iterator	15

2.5	Graphs	16
2.5.1	Graph Types	16
2.5.2	Graph — Directed Sequential Bounded Managed Iterator	17
2.5.3	Graph — Directed Sequential Unbounded Managed Iterator	17
2.5.4	Graph — Undirected Sequential Bounded Managed Iterator	18
2.5.5	Graph — Undirected Sequential Unbounded Managed Iterator	18
2.6	Graph Utilities	19
2.7	Module Names	19
	References	21

Part 2 — Trees

3	Tree Abstraction	23
3.1	Concepts and Definitions	23
3.1.1	Basic Tree Definitions	23
3.1.2	Attributes of Nodes	24
3.1.3	Relationships Between Nodes	25
3.1.4	Attributes of Trees	25
3.1.5	Kinds of Trees	26
3.1.6	Tree Traversals	26
3.1.6.1	General Tree Traversals	26
3.1.6.2	Binary Tree Traversals	27
3.1.7	Forests	27
3.2	Applications and Uses	29
3.3	Binary Search Tree Operations	29
3.3.1	Constructors	31
3.3.1.1	Create	31
3.3.1.2	Destroy	31
3.3.1.3	Clear	31
3.3.1.4	Assign	32
3.3.1.5	Insert	32
3.3.1.6	MakeTree	33
3.3.1.7	Remove	33

3.3.2	Selectors	34
3.3.2.1	IsDefined	34
3.3.2.2	IsEmpty	34
3.3.2.3	IsEqual	34
3.3.2.4	ExtentOf	35
3.3.2.5	IsPresent	35
3.3.3	Passive Iterators	35
3.3.3.1	Preorder	35
3.3.3.2	Inorder	36
3.3.3.3	Postorder	36
3.3.4	Active Iterators	36
3.3.4.1	RootOf	37
3.3.4.2	LeftOf	37
3.3.4.3	RightOf	37
3.3.4.4	IsNull	37
3.3.4.5	KeyOf	38
3.3.4.6	DataOf	38
3.4	Balanced Binary Trees	38
3.4.1	Height-Balanced (AVL) Trees	39
3.4.2	Weight-Balanced (BB) Trees	39
3.4.3	Path-Balanced (IPB) Trees	39
3.4.4	Rotations	40
3.5	Tree Exceptions	40
3.5.1	Initialization Failed	40
3.5.2	Overflow	40
3.5.3	Tree is Null	42
3.5.4	Type Error	42
3.5.5	Undefined	42
3.6	Summary	42
3.6.1	Operations Summary	42
3.6.2	Exceptions Summary	43
	References	45
4	The Unbounded Binary Tree	47
4.1	Tree Types Module	47
4.1.1	Tree Operations	47
4.1.2	Tree Exceptions	48
4.1.3	Tree Types and Procedure Types	49

4.2	Unbounded Binary Search Tree Interface	50
4.2.1	Type Declarations	51
4.2.2	Exception Handling	52
4.2.3	Constructors	52
4.2.4	Selectors	54
4.2.5	Passive Iterators	55
4.2.6	Active Iterators	55
4.3	Unbounded Binary Search Tree Implementation	58
4.3.1	Internal Representation	59
4.3.2	Exception Handling	60
4.3.3	Local Operations	61
4.3.4	Constructors	62
4.3.5	Selectors	69
4.3.6	Passive Iterators	72
4.3.7	Active Iterators	74
4.3.8	Module Initialization	76
4.4	Unbounded Binary Search Tree Utilities Interface	76
4.4.1	Utility Selectors	77
4.4.2	Debugging Iterators	77
4.5	Unbounded Binary Search Tree Utilities Implementation	78
4.5.1	Utility Selectors	78
4.5.2	Debugging Iterators	79
	References	80
5	The Bounded Binary Tree	81
5.1	Bounded Binary Search Tree Interface	81
5.1.1	Type Declarations	82
5.1.2	Exception Handling	82
5.1.3	Constructors	83
5.1.4	Selectors	84
5.1.5	Passive Iterators	86
5.1.6	Active Iterators	86
5.2	Bounded Binary Search Tree Implementation	89
5.2.1	Internal Representation	90
5.2.2	Exception Handling	91
5.2.3	Free List Management	92
5.2.4	Local Operations	94

Table of Contents

5.2.5	Constructors	94
5.2.6	Selectors	101
5.2.7	Passive Iterators	105
5.2.8	Active Iterators	107
5.2.9	Module Initialization	109

References	110
------------	-----

6 The Unbounded AVL Tree 111

6.1 Unbounded AVL Tree Interface 111

6.1.1	Type Declarations	112
6.1.2	Exception Handling	112
6.1.3	Constructors	113
6.1.4	Selectors	115
6.1.5	Passive Iterators	115
6.1.6	Active Iterators	116

6.2 Unbounded AVL Tree Implementation 119

6.2.1	Internal Representation	120
6.2.2	Exception Handling	121
6.2.3	Local Operations	122
6.2.4	Constructors	125
6.2.5	Selectors	135
6.2.6	Passive Iterators	138
6.2.7	Active Iterators	140
6.2.8	Module Initialization	142

6.3 Unbounded AVL Tree Utilities Interface 142

6.3.1	Utility Selectors	143
6.3.2	Debugging Iterators	143

6.4 Unbounded AVL Tree Utilities Implementation 144

6.4.1	Utility Selectors	144
6.4.2	Debugging Iterators	145

References	146
------------	-----

7 The Unbounded BB Tree 147

7.1 Unbounded BB Tree Interface 147

7.1.1	Type Declarations	148
7.1.2	Exception Handling	149

7.1.3	Constructors	149
7.1.4	Selectors	151
7.1.5	Passive Iterators	152
7.1.6	Active Iterators	152
7.2	Unbounded BB Tree Implementation	155
7.2.1	Internal Representation	156
7.2.2	Exception Handling	157
7.2.3	Local Operations	158
7.2.4	Constructors	160
7.2.5	Selectors	168
7.2.6	Passive Iterators	172
7.2.7	Active Iterators	174
7.2.8	Module Initialization	175
References		176
8	The Unbounded k-Balanced Binary Tree	177
8.1	Unbounded k-Balanced Binary Tree Interface	177
8.1.1	Type Declarations	178
8.1.2	Exception Handling	179
8.1.3	Constructors	179
8.1.4	Selectors	181
8.1.5	Passive Iterators	182
8.1.6	Active Iterators	182
8.2	Unbounded k-Balanced Binary Tree Implementation	185
8.2.1	Internal Representation	186
8.2.2	Exception Handling	187
8.2.3	Local Operations	189
8.2.4	Constructors	192
8.2.5	Selectors	200
8.2.6	Passive Iterators	204
8.2.7	Active Iterators	205
8.2.8	Module Initialization	207
8.3	Unbounded k-Balanced Binary Tree Utilities Interface	208
8.3.1	Utility Selectors	208
8.3.2	Debugging Iterators	208

8.4	Unbounded k-Balanced Binary Tree Utilities Implementation	209
8.4.1	Utility Selectors	209
8.4.2	Debugging Iterators	210
	References	212

Part 3 Graphs

9	Graph Abstraction	213
9.1	Concepts and Definitions	213
9.1.1	Directed Graphs	213
9.1.2	Undirected Graphs	214
9.1.3	Labeled and Weighted Graphs	216
9.2	Applications and Uses	217
9.3	Directed Graph Specifications	217
9.3.1	Directed Graph Types	217
9.3.2	Directed Graph Constructors	218
9.3.2.1	Create	218
9.3.2.2	Destroy	219
9.3.2.3	Clear	219
9.3.2.4	Assign	219
9.3.2.5	Insert	220
9.3.2.6	Remove	220
9.3.2.7	SetLabel	221
9.3.2.8	Link	221
9.3.2.9	Unlink	222
9.3.2.10	SetAttribute	222
9.3.3	Directed Graph Selectors	223
9.3.3.1	IsDefined	223
9.3.3.2	IsEmpty	223
9.3.3.3	OrderOf	223
9.3.3.4	SizeOf	224
9.3.3.5	OutDegree	224
9.3.3.6	InDegree	224
9.3.3.7	LabelOf	224

9.3.3.8	IsVertex	225
9.3.3.9	GraphOf	225
9.3.3.10	IsEdge	225
9.3.3.11	AttributeOf	225
9.3.3.12	InitialOf	226
9.3.3.13	FinalOf	226
9.3.4	Directed Graph Passive Iterators	226
9.3.4.1	TraverseVertices	227
9.3.4.2	TraverseEdges	227
9.3.4.3	Iterate	227
9.3.4.4	LoopVertices	227
9.3.4.5	LoopEdges	228
9.3.4.6	LoopIterate	228
9.3.5	Directed Graph Active Iterators	228
9.3.5.1	FirstVertex	228
9.3.5.2	NextVertex	229
9.3.5.3	FirstEdge	229
9.3.5.4	NextVertex	229
9.4	Undirected Graph Specifications	230
9.4.1	Undirected Graph Constructors	230
9.4.1.1	Remove	230
9.4.1.2	Link	231
9.4.2	Undirected Graph Selectors	231
9.4.2.1	DegreeOf	231
9.4.2.2	FirstOf	232
9.4.2.3	SecondOf	232
9.4.2.4	IncidentOn	232
9.4.3	Undirected Graph Passive Iterators	232
9.5	Graph Exceptions	233
9.5.1	EdgeIsNull	233
9.5.2	EdgeNotInGraph	233
9.5.3	Initialization Failed	233
9.5.4	Overflow	233
9.5.5	Undefined	234
9.5.6	VertexIsNull	234
9.5.7	VertexNotInGraph	234
9.6	Graph Utilities	234
9.6.1	BreadthFirstSearch	235
9.6.2	DepthFirstSearch	235
9.6.3	HasSelfLoops	236
9.6.4	IsIsolated	236

Table of Contents

9.6.5	IsReachable	237
9.6.6	IsTerminal	237
9.6.7	MaxDegree	237
9.6.7	MinDegree	238
9.7	Summary	238
9.7.1	Directed Graph Operations Summary	238
9.7.2	Directed Graph Exceptions Summary	239
9.7.3	Undirected Graph Operations Summary	241
9.7.4	Undirected Graph Exceptions Summary	242
9.7.5	Graph Utility Operations Summary	243
9.7.6	Graph Utility Exceptions Summary	244
	References	245

10 The Unbounded Directed Graph 247

10.1	Graph Types Interface	247
10.2	Unbounded Directed Graph Interface	248
10.2.1	Type Declarations	250
10.2.2	Exception Handling	251
10.2.3	Graph Constructors	252
10.2.4	Vertex Constructors	253
10.2.5	Edge Constructors	253
10.2.6	Graph Selectors	254
10.2.7	Vertex Selectors	255
10.2.8	Edge Selectors	256
10.2.9	Passive Iterators	257
10.2.10	Active Iterators	258
10.3	Unbounded Directed Graph Implementation	260
10.3.1	Internal Representation	261
10.3.2	Exception Handling	264
10.3.3	Local Routines	265
10.3.4	Graph Constructors	269
10.3.5	Vertex Constructors	277
10.3.6	Edge Constructors	279
10.3.7	Graph Selectors	282
10.3.8	Vertex Selectors	284
10.3.9	Edge Selectors	287
10.3.10	Passive Iterators	288
10.3.11	Active Iterators	292
10.3.12	Module Initialization	293

References	294
11 The Bounded Directed Graph	295
11.1 Bounded Directed Graph Interface	295
11.1.1 Type Declarations	296
11.1.2 Exception Handling	298
11.1.3 Graph Constructors	298
11.1.4 Vertex Constructors	299
11.1.5 Edge Constructors	300
11.1.6 Graph Selectors	301
11.1.7 Vertex Selectors	302
11.1.8 Edge Selectors	302
11.1.9 Passive Iterators	303
11.1.10 Active Iterators	304
11.2 Bounded Directed Graph Implementation	307
11.2.1 Internal Representation	308
11.2.2 Exception Handling	311
11.2.3 Local Routines	313
11.2.4 Graph Constructors	316
11.2.5 Vertex Constructors	325
11.2.6 Edge Constructors	328
11.2.7 Graph Selectors	331
11.2.8 Vertex Selectors	333
11.2.9 Edge Selectors	335
11.2.10 Passive Iterators	337
11.2.11 Active Iterators	341
11.2.12 Module Initialization	342
References	343
12 The Unbounded Undirected Graph	344
12.1 Unbounded Undirected Graph Interface	345
12.1.1 Type Declarations	346
12.1.2 Exception Handling	347
12.1.3 Graph Constructors	348
12.1.4 Vertex Constructors	349
12.1.5 Edge Constructors	350
12.1.6 Graph Selectors	350
12.1.7 Vertex Selectors	351

Table of Contents

12.1.8	Edge Selectors	352
12.1.9	Passive Iterators	353
12.1.10	Active Iterators	354
12.2	Unbounded Undirected Graph Implementation	357
12.2.1	Internal Representation	358
12.2.2	Exception Handling	363
12.2.3	Local Routines	364
12.2.4	Graph Constructors	370
12.2.5	Vertex Constructors	379
12.2.6	Edge Constructors	382
12.2.7	Graph Selectors	384
12.2.8	Vertex Selectors	386
12.2.9	Edge Selectors	388
12.2.10	Passive Iterators	390
12.2.11	Active Iterators	393
12.2.12	Module Initialization	394
	References	396
13	Graph Utilities	397
13.1	DigraphSUMI Utilities Interface	397
13.1.1	Graph Selector Utilities	398
13.1.2	Vertex Selector Utilities	398
13.1.3	Graph Traversal Utilities	399
13.2	DigraphSUMI Utilities Implementation	399
13.2.1	Graph Selector Utilities	400
13.2.2	Vertex Selector Utilities	402
13.2.3	Graph Traversal Utilities	404
	References	407

Appendices

A	Modula-2 Syntax Diagrams	409
B	Standard Modula-2 Routines	419
C	Modula-2 Compilers	423
D	Module Tables	427
E	Import Graphs	443

Bibliography**447**

Table of Figures

Figure 2.1	Tree Module Organization	12
Figure 2.2	Graph Module Organization	16
Figure 2.3	Component Forms	20
Figure 3.1	Binary Tree Abstraction	22
Figure 3.2	Multiway Tree Abstraction	22
Figure 3.3	Ordered vs. Unordered Trees	24
Figure 3.4	A Forest of Trees	26
Figure 3.5	Binary Search Tree	28
Figure 3.6	Binary Search Tree Node	28
Figure 3.7	Single Left Rotation	39
Figure 3.8	Double Rotation	39
Figure 4.1	Unbounded Binary Tree Internal Representation	57
Figure 5.1	Bounded Binary Tree Internal Representation	89
Figure 6.1	Unbounded AVL Tree Internal Representation	119
Figure 7.1	Unbounded BB Tree Internal Representation	154
Figure 8.1	Unbounded IPB Tree Internal Representation	185
Figure 9.1	Directed Graph Abstraction	213
Figure 9.2	Undirected Graph Abstraction	213
Figure 9.3	The Directed Network Abstraction	214
Figure 9.4	A Multigraph	214
Figure 10.1	Unbounded Directed Graph Internal Representation	260
Figure 11.1	Bounded Directed Graph Internal Representation	307
Figure 12.1	Unbounded Undirected Graph Internal Representation	357
Figure 12.2	Relationship between Vertices, Edges, and Edge References	357
Figure E.1	Tree Types Import Graph	441
Figure E.2	Tree Definition Module Import Graph	441
Figure E.3	Tree Implementation Module Import Graph	442
Figure E.4	Graph Types Import Graph	442
Figure E.5	Graph Definition Module Import Graph	442
Figure E.6	Graph Implementation Module Import Graph	443