

# table of contents

<b>begin.....</b>	<b>3</b>
A Book on Today's Language.....	4
Learn by Doing.....	4
Acknowledgments.....	5
About Myself, the Author.....	6
<b>Table of Contents.....</b>	<b>7</b>
<b>Part I: Foundations.....</b>	<b>17</b>
Summary of Part I.....	18
<b>01: Coding in Pascal.....</b>	<b>19</b>
Let's Start with Code.....	19
A First Console Application.....	20
A First Visual Application.....	21
Syntax and Coding Style.....	24
Comments.....	24
Comments and XML Doc.....	26
Symbolic Identifiers.....	27
Whitespace.....	29
Indentation.....	30
Syntax Highlighting.....	31
Language Keywords.....	32
The Structure of a Program.....	37
Unit and Program Names.....	38
Units and Scope.....	41
The Program File.....	43
Compiler Directives.....	43

## 8 - table of contents

Conditional Defines.....	44
Compiler Versions.....	45
Include Files.....	46
<b>02: Variables and Data Types.....</b>	<b>49</b>
Variables and Assignments.....	50
Literal Values.....	51
Assignment Statements.....	52
Assignments and Conversion.....	52
Initializing Global Variables.....	53
Initializing Local Variables.....	53
Inline Variables.....	54
Constants.....	55
Lifetime and Visibility of Variables.....	57
Data Types.....	59
Ordinal and Numeric Types.....	59
Boolean.....	64
Characters.....	64
Floating Point Types.....	67
Simple User-Defined Data Types.....	69
Named vs. Unnamed Types.....	69
Type Aliases.....	70
Subrange Types.....	71
Enumerated Types.....	72
Set Types.....	74
Expressions and Operators.....	75
Using Operators.....	75
Operators and Precedence.....	76
Date and Time.....	78
Type Casting and Type Conversions.....	81
<b>03: Language Statements.....</b>	<b>83</b>
Simple and Compound Statements.....	84
The If Statement.....	85
Case Statements.....	86
The For Loop.....	88
The for-in Loop.....	90
While and Repeat Statements.....	92
Examples of Loops.....	93
Breaking the Flow with Break and Continue.....	94
<b>04: Procedures and Functions.....</b>	<b>97</b>
Procedures and Functions.....	97
Forward Declarations.....	100
A Recursive Function.....	101
What Is a Method?.....	102
Parameters and Return Values.....	103
Exit with a Result.....	104
Reference Parameters.....	105

Constant Parameters.....	106
Function Overloading.....	107
Overloading and Ambiguous Calls.....	109
Default Parameters.....	110
Inlining.....	112
Advanced Features of Functions.....	115
Object Pascal Calling Conventions.....	115
Procedural Types.....	116
External Functions Declarations.....	118
<b>o5: Arrays and Records.....</b>	<b>121</b>
Array Data Types.....	121
Static Arrays.....	122
Array Size and Boundaries.....	123
Multi-Dimensional Static Arrays.....	124
Dynamic Arrays.....	125
Open Array Parameters.....	129
Record Data Types.....	132
Using Arrays of Records.....	134
Variant Records.....	135
Fields Alignments.....	136
What About the With Statement?.....	137
Records with Methods.....	139
Self: The Magic Behind Records.....	141
Initializing Records.....	142
Records and Constructors.....	142
Operators Gain New Ground.....	143
Operators and Custom Managed Record.....	147
Variants.....	151
Variants Have No Type.....	151
Variants in Depth.....	153
Variants Are Slow.....	154
What About Pointers?.....	155
File Types, Anyone?.....	158
<b>o6: All About Strings.....</b>	<b>159</b>
Unicode: An Alphabet for the Entire World.....	160
Characters from the Past: from ASCII to ISO Encodings.....	160
Unicode Code Points and Graphemes.....	161
From Code Points to Bytes (UTF).....	162
The Byte Order Mark.....	164
Looking at Unicode.....	165
The Char Type Revisited.....	167
Unicode Operations With The Character Unit.....	168
Unicode Character Literals.....	170
What about 1-Byte Chars?.....	171
The String Data Type.....	171
Passing Strings as Parameters.....	175
The Use of [] and String Character Counting Modes.....	176

## 10 - table of contents

Concatenating Strings.....	178
The String Helper Operations.....	179
More String RTL Functions.....	182
Formatting Strings.....	183
The Internal Structure of Strings.....	185
Looking at Strings in Memory.....	186
Strings and Encoding.....	188
Other Types for Strings.....	190
The UCS4String type.....	191
Older String Types.....	191
<b>Part II: OOP in Object Pascal.....</b>	<b>193</b>
Summary of Part II.....	194
<b>o7: Objects.....</b>	<b>195</b>
Introducing Classes and Objects.....	195
The Definition of a Class.....	196
Classes in Other OOP Languages.....	197
The Class Methods.....	198
Creating an Object.....	199
The Object Reference Model.....	200
Disposing of Objects.....	201
What is “Nil”?.....	202
Records vs. Classes in Memory.....	202
Private, Protected, and Public.....	203
An Example of Private Data.....	204
Encapsulation and Forms.....	206
The Self Identifier.....	208
Creating Components Dynamically.....	209
Constructors.....	211
Managing Local Class Data with Constructors and Destructors.....	213
Overloaded Methods and Constructors.....	214
The Complete TDate Class.....	216
Nested Types and Nested Constants.....	219
<b>o8: Inheritance.....</b>	<b>221</b>
Inheriting from Existing Types.....	221
A Common Base Class.....	224
Protected Fields and Encapsulation.....	225
Using the “Protected Hack”.....	225
From Inheritance to Polymorphism.....	227
Inheritance and Type Compatibility.....	228
Late Binding and Polymorphism.....	229
Overriding, Redefining, and Reintroducing Methods.....	231
Inheritance and Constructors.....	233
Virtual versus Dynamic Methods.....	234
Abstracting Methods and Classes.....	235
Abstract Methods.....	236
Sealed Classes and Final Methods.....	238

Safe TypeCast Operators.....	238
Visual Form Inheritance.....	240
Inheriting From a Base Form.....	241
<b>09: Handling Exceptions.....</b>	<b>245</b>
Try-Except Blocks.....	246
The Exceptions Hierarchy.....	248
Raising Exceptions.....	250
Exceptions and the Stack.....	251
The Finally Block.....	252
Restore the Cursor with a Finally Block.....	254
Restore the Cursor with a Managed Record.....	254
Exceptions in the Real World.....	255
Global Exceptions Handling.....	256
Exceptions and Constructors.....	257
Advanced Features of Exceptions.....	259
Nested Exceptions and the InnerException Mechanism.....	259
Intercepting an Exception.....	262
<b>10: Properties and Events.....</b>	<b>265</b>
Defining Properties.....	266
Properties Compared to Other Programming Languages.....	267
Properties Implement Encapsulation.....	268
Code Completion for Properties.....	269
Adding Properties to Forms.....	270
Adding Properties to the TDate Class.....	271
Using Array Properties.....	273
Setting Properties by Reference.....	274
The published Access Specifier.....	276
Design-Time Properties.....	277
Published and Forms.....	277
Automatic RTTI.....	278
Event-Driven Programming.....	279
Method Pointers.....	280
The Concept of Delegation.....	282
Events are Properties.....	284
Adding an Event to the TDate Class.....	285
Creating a TDate Component.....	287
Implementing Enumeration Support in a Class.....	290
15 Tips About Mixing RAD and OOP.....	292
Tip 1: A Form is a Class.....	293
Tip 2: Name Components.....	293
Tip 3: Name Events.....	293
Tip 4: Use Form Methods.....	294
Tip 5: Add Form Constructors.....	294
Tip 6: Avoid Global Variables.....	294
Tip 7: Never use an Instance Variable in its Implementation.....	295
Tip 8: Seldom Use a Form's Variable.....	295
Tip 9: Remove the Global Form1 Variable.....	295

## 12 - table of contents

Tip 10: Add Form Properties.....	295
Tip 11: Expose Component Properties.....	296
Tip 12: Use Array Properties when Needed.....	296
Tip 13: Starting Operations in Properties.....	296
Tip 14: Hide Components.....	297
Tip 15: Use an OOP Form Wizard.....	298
Tips Conclusion.....	298
<b>11: Interfaces.....</b>	<b>299</b>
Using Interfaces.....	300
Declaring an Interface.....	301
Implementing the Interface.....	302
Interfaces and Reference Counting.....	303
Errors in Mixing References.....	304
Weak and Unsafe Interface References.....	306
Advanced Interface Techniques.....	308
Interface Properties.....	308
Interface Delegation.....	310
Multiple Interfaces and Methods Aliases.....	311
Interface Polymorphism.....	312
Extracting Objects from Interface References.....	314
Implementing an Adapter Pattern with Interfaces.....	315
<b>12: Manipulating Classes.....</b>	<b>319</b>
Class Methods and Class Data.....	319
Class Data.....	320
Virtual Class Methods and the Hidden Self Parameter.....	321
Class Static Methods.....	321
Class Properties.....	323
A Class with an Instance Counter.....	324
Class Constructors (and Destructors).....	325
Class Constructors in the RTL.....	326
Implementing the Singleton Pattern.....	327
Class References.....	328
Class References in the RTL.....	329
Creating Components Using Class References.....	329
Class and Record Helpers.....	332
Class Helpers.....	333
Class Helpers and Inheritance.....	335
Adding Control Enumeration with a Class Helper.....	336
Record Helpers for Intrinsic Types.....	338
Helpers for Type Aliases.....	340
<b>13: Objects and Memory.....</b>	<b>341</b>
Global Data, the Stack, and the Heap.....	342
The Global Memory.....	342
The Stack.....	343
The Heap.....	344
The Object Reference Model.....	344

Passing Objects as Parameters.....	345
Memory Management Tips.....	346
Destroying Objects You Create.....	347
Destroying Objects Only Once.....	348
Memory Management and Interfaces.....	350
More on Weak References.....	350
The Unsafe Attribute.....	354
Tracking and Checking Memory.....	354
Memory Status.....	355
FastMM4.....	355
Tracking Leaks and Other Global Settings.....	356
Buffer Overruns in the Full FastMM4.....	357
Memory Management on Platforms Other than Windows.....	360
Tracking Per-Class Allocations.....	360
Writing Robust Applications.....	360
Constructors, Destructors, and Exceptions.....	361
Nested Finally Blocks.....	362
Dynamic Type Checking.....	363
Is this Pointer an Object Reference?.....	365
<b>Part III: Advanced Features.....</b>	<b>367</b>
Chapters of Part III.....	368
<b>14: Generics.....</b>	<b>369</b>
Generic Key-Value Pairs.....	370
Inline Variables and Generics Type Inference.....	373
Type Rules on Generics.....	373
Generics in Object Pascal.....	374
Generic Types Compatibility Rules.....	375
Generic Methods for Standard Classes.....	376
Generic Type Instantiation.....	377
Generic Type Functions.....	379
Class Constructors for Generic Classes.....	382
Generic Constraints.....	383
Class Constraints.....	383
Specific Class Constraints.....	385
Interface Constraints.....	386
Interface References vs. Generic Interface Constraints.....	388
Default Constructor Constraint.....	389
Constraints Summary and Combining them.....	390
Predefined Generic Containers.....	391
Using TList<T>.....	392
Sorting a TList<T>.....	393
Sorting with an Anonymous Method.....	394
Object Containers.....	396
Using a Generic Dictionary.....	396
Dictionaries vs. String Lists.....	400
Generic Interfaces.....	401
Predefined Generic Interfaces.....	403

## 14 - table of contents

Smart Pointers in Object Pascal.....	404
Using Records for Smart Pointers.....	404
Implementing a Smart Pointer with a Generic Managed Record.....	405
Implementing a Smart Pointer with a Generic Record and an Interface.....	407
Adding Implicit Conversion.....	409
Comparing Smart Pointer Solutions.....	410
Covariant Return Types with Generics.....	410
About Animals, Dogs, and Cats.....	410
A Method with a Generic Result.....	412
Returning a Derived Object of a Different Class.....	412
<b>15: Anonymous Methods.....</b>	<b>415</b>
Syntax and Semantics of Anonymous Methods.....	416
An Anonymous Method Variable.....	416
An Anonymous Method Parameter.....	417
Using Local Variables.....	418
Extending the Lifetime of Local Variables.....	418
Anonymous Methods Behind the Scenes.....	420
The (Potentially) Missing Parenthesis.....	420
Implementation of Anonymous Methods.....	421
Ready-To-Use Reference Types.....	422
Anonymous Methods in the Real World.....	423
Anonymous Event Handlers.....	423
Timing Anonymous Methods.....	425
Threads Synchronization.....	427
AJAX in Object Pascal.....	429
<b>16: Reflection and Attributes.....</b>	<b>433</b>
Extended RTTI.....	434
A First Example.....	434
Compiler Generated Information.....	435
Weak- and Strong-Type Linking.....	437
The RTTI Unit.....	438
The RTTI Classes in the Rtti Unit.....	440
RTTI Objects Lifetime Management and the TRttiContext record.....	441
Displaying Class Information.....	442
RTTI for Packages.....	444
The TValue Structure.....	445
Reading a Property with TValue.....	447
Invoking Methods.....	447
Using Attributes.....	448
What is an Attribute?.....	449
Attribute Classes and Attribute Declarations.....	449
Browsing Attributes.....	451
Virtual Method Interceptors.....	453
RTTI Case Studies.....	457
Attributes for ID and Description.....	457
XML Streaming.....	461
Other RTTI-Based Libraries.....	468



<b>17: TObject and the System Unit.....</b>	<b>469</b>
The TObject Class.....	470
Construction and Destruction.....	470
Knowing About an Object.....	471
More Methods of the TObject Class.....	472
TObject's Virtual Methods.....	474
TObject Class Summary.....	477
Unicode and Class Names.....	478
The System Unit.....	479
Selected System Types.....	479
Interfaces in the System Unit.....	480
Selected System Routines.....	481
Predefined RTTI Attributes.....	481
<b>18: Other Core RTL Classes.....</b>	<b>483</b>
The Classes Unit.....	484
The Classes in the Classes Unit.....	484
The TPersistent Class.....	485
The TComponent Class.....	486
Modern File Access.....	489
The Input/Output Utilities Unit.....	489
Introducing Streams.....	491
Using Readers and Writers.....	492
Building Strings and String Lists.....	494
The TStringBuilder class.....	494
Using String Lists.....	496
The Run-Time Library is Quite Large.....	496
<b>In Closing.....</b>	<b>499</b>
<b>end.....</b>	<b>501</b>
Appendix Summary.....	501
<b>A: The Evolution of Object Pascal.....</b>	<b>503</b>
Wirth's Pascal.....	504
Turbo Pascal.....	504
The early days of Delphi's Object Pascal.....	505
Object Pascal From CodeGear to Embarcadero.....	506
Going Mobile.....	507
The Delphi 10.x Period.....	507
<b>B: Glossary.....</b>	<b>509</b>
A.....	509
B.....	510
C.....	510
D.....	512
E.....	512
F.....	513
G.....	513

## 16 - table of contents

H.....	514
I.....	514
M.....	515
O.....	515
P.....	516
R.....	517
S.....	518
U.....	518
V.....	519
W.....	519
<b>C: Index.....</b>	<b>521</b>