Table of Contents

Part I Classes 1

Chapter 1 Total Programming 5

1.1 Class Interface 6
1.1.1 Class heading 7
1.1.2 Methods 10
1.1.2.1 Getters and setters 13
1.1.2.2 Mutators and inspectors 15
1.1.2.3 Instance methods versus class methods 18
1.1.2.4 Explicit arguments 23
1.1.2.5 Overloading 24
1.1.2.6 Constructors 25
1.1.3 Documentation 27
1.1.3.1 Method summary 32
1.1.3.2 Parameters 33
1.1.3.3 Postconditions 34
1.1.4 Class usage 47
1.1.4.1 Construction of new objects 48
1.1.4.2 Invocation of instance methods 50
1.1.4.3 Invocation of class methods 53
1.1.4.4 Application Program Interface (API) 53
1.1.4.5 Main method 58

1.2 Class Representation 60
1.2.1 Instance variables and class variables 62
1.2.1.1 Instance variables 63
1.2.1.2 Class variables 69
1.2.1.3 Final qualifications 70
1.2.2 Value semantics and reference semantics 73
1.2.2.1 Value semantics 74
1.2.2.2 Reference semantics 75
1.2.2.3 Boxing and unboxing 78
1.2.2.4 Call by value 79
1.2.3 Primitive types 82
1.2.3.1 Integral types 82
1.2.3.2 Floating point types 86
1.2.3.3 Characters 89
1.2.3.4 Booleans 90

1.3 Class Implementation 91
1.3.1 Instance methods 98
1.3.2 Class methods 105

1.3.3 Constructors 106

1.4 Class Verification 109

1.4.1 Principles of black box testing 110

1.4.2 Black box testing with JUnit 113

1.4.2.1 Test fixture 114

1.4.2.2 Individual tests 116

Chapter 2 Nominal Programming 121

2.1 Class Invariants 122

2.1.1 Encapsulation 125

2.1.2 Formal specification 126

2.1.3 Semantics 129

2.1.3.1 Class properties 129

2.1.3.2 Object properties 131

2.2 Preconditions 139

2.2.1 Semantics 146

2.2.2 Verification 149

2.2.2.1 Static verification 149

2.2.2.2 Dynamic verification 151

2.3 Postconditions Revisited 153

2.3.1 New qualification 153

2.3.2 Effect clauses 155

2.3.3 Inspectors 156

2.3.4 Verification 157

2.3.5 Model methods 163

2.4 Multi-Valued Properties 164

2.4.1 Representation 169

2.4.1.1 Declaration of arrays 170

2.4.1.2 Construction of arrays 172

2.4.2 Getters and setters 174

2.4.2.1 Number of elements 174

2.4.2.2 Accessing individual elements 175

2.4.2.3 Restrictions on individual elements 177

2.4.2.4 Initialization 178

2.4.2.5 Additional methods 180

2.5 Structured Programming 182

2.5.1 Iterative statements 183

2.5.1.1 While statement 183

2.5.1.2 Do-while statement 185

2.5.1.3 For statement 186

2.5.1.4 Enhanced for statement 190

2.5.2 Loop invariants 192

2.5.3 Correctness of iterative algorithms 194

2.5.4 Termination of iterative constructs 198

2.5.5 Alternative implementations of algorithms 200

2.6 A Word on Formal Specifications 205

Chapter 3 Defensive Programming 209
3.1 Definition of Exceptions 210
3.1.1 Predefined exception classes 211
3.1.2 Self-defined exception classes 213
3.1.2.1 The root class of throwables 213
3.1.2.2 The class of illegal amount exceptions 215
3.2 Extended Method Specifications 219
3.2.1 Structure of method specifications 220
3.2.2 Extended method signature 227
3.2.3 Defensive specification of bank accounts 230
3.2.4 Semantics of throws clauses 235
3.3 Throwing Exceptions 237
3.4 Catching Exceptions 247
3.4.1 General structure of try-catch statements 247
3.4.2 Semantics of try-catch statements 249
3.4.3 Order of catchers 251
3.4.4 Nesting of try-catch statements 252
3.4.4.1 Statically nested try-catch statements 252
3.4.4.2 Dynamically nested try-catch statements 255
3.4.5 The class of bank accounts: Catching exceptions 258
3.5 The Finally Clause 261
3.5.1 Semantics 262
3.5.2 Application 265
3.6 Verification 266
3.7 Dependencies between Properties 270
3.7.1 Checkers 271
3.7.2 Initialization 272
Chapter 4 Value Classes 279
4.1 Objects versus Values 280
4.1.1 Objects 280
4.1.1.1 State of objects 280
4.1.1.2 Behavior of objects 281
4.1.1.3 Identity of objects 282
4.1.1.4 Lifetime of objects 283
4.1.2 Values 283
4.2 Definition of Value Classes 284
4.2.1 Algebras 284
4.2.2 Money amounts 285
4.2.3 Basic inspectors 286
4.2.4 Constructors 286
4.2.5 Operators 287
4.2.6 Methods inherited from the root class of objects 288
4.2.7 Definition of the class of money amounts 291
4.3 Enumerations 300
4.4 Bank Accounts 305

Part II Associations 317

Chapter 5 Associations with Restricted Multiplicity 321

5.1 Class Diagrams 323
5.1.1 Classes 323
5.1.1.1 Graphical representation of classes 324
5.1.1.2 Static classification 325
5.1.1.3 Disjoint classification 328
5.1.2 Associations 329
5.1.3 Object diagrams 333

5.2 Unidirectional Associations 334
5.2.1 Representation 334
5.2.2 Manipulation 336

5.3 Bidirectional Associations 339
5.3.1 Representation 340
5.3.2 Manipulation 341
5.3.2.1 Definition of the class of bank cards 342
5.3.2.2 Definition of the class of bank accounts 354

5.4 Memory Management 359
5.4.1 Run-time stack 360
5.4.2 Object memory 363
5.4.3 Recovery of object memory 364
5.4.4 Finalization 368

5.5 Destructors 369

5.6 Packages 371
5.6.1 Packages 371
5.6.2 Nested packages 372
5.6.3 Packages and directories 374
5.6.4 Package statement 375
5.6.5 Import statement 376
5.6.6 Access rights 380

5.7 The Banking System Revisited 383

Chapter 6 Associations with Unrestricted Multiplicity 387

6.1 Detailed Design 388
6.1.1 Classification 389
6.1.2 Relations 389
6.1.3 Attributes 391
6.1.4 Methods 391

6.2 Specification 392
6.2.1 Introduction to interfaces 392
6.2.1.1 Non-generic interfaces 393
6.2.1.2 Generic classes and interfaces 394
6.2.2 Lists 397
6.2.2.1 Number of elements 397
6.2.2.2 Accessors 398
6.2.2.3 Adders and removers 398

6.2.2.4 Inspectors 400
6.2.2.5 Constructors 401
6.2.2.6 Destructor 401
6.2.3 Sets 401
6.2.3.1 Basic inspector 402
6.2.3.2 Checker 402
6.2.3.3 Adders and removers 403
6.2.3.4 Inspectors 404
6.2.3.5 Constructors and destructor 404
6.3 Representation 411
6.3.1 Data structures 412
6.3.1.1 Lists 412
6.3.1.2 Sets 414
6.3.1.3 Maps 415
6.3.1.4 Representation of bank accounts 416
6.3.2 Representation invariants 418
6.3.2.1 Semantics of representation invariants 419
6.3.2.2 Specification of representation invariants 420
6.4 Implementation 422
6.4.1 Constructor 422
6.4.2 Savings accounts: Basic methods 423
6.4.3 Grantees: Basic methods 427
6.4.4 Additional methods 434
6.4.5 Destructor 437
6.4.6 The class of savings accounts 437
6.5 Verification 441
6.5.1 Basics of white box testing 441
6.5.1.1 Program graphs 441
6.5.1.2 Decision-to-decision paths 443
6.5.1.3 Path testing 444
6.5.1.4 Data flow testing 445
6.5.2 Practice of white box testing 447
6.5.3 Black & white testing 450
Part III Inheritance 453
Chapter 7 Extension Inheritance 457
7.1 Taxonomies 458
7.1.1 Semantics of inheritance 458
7.1.2 Design notation 459
7.2 Subclass Definition 461
7.2.1 Inheritance of instance features 462
7.2.2 Inheritance of static features 463
7.2.3 Access rights 465
7.2.4 Constructors of subclasses 467
7.2.5 Extending subclasses 471
7.2.6 Overloading revisited 474
7.2.7 Open-closed principle 475

7.3 Abstract Classes 477
7.3.1 Definition of abstract classes 478
7.3.2 Testing 480
7.4 Polymorphism 482
7.4.1 Variables with reference semantics 482
7.4.2 Variables with value semantics 487
7.5 Run-Time Type Information 488
7.5.1 The operator "instanceof" 489
7.5.2 Type casts 490
7.5.3 Reflection 491
7.5.4 Generalization 493
Chapter 8 Specialization Inheritance 497
8.1 Overriding Instance Methods 498
8.1.1 Comprehensive method definitions 499
8.1.2 Overriding abstract methods 501
8.1.3 Dynamic binding 504
8.1.4 Overriding concrete methods 505
8.1.5 Final qualifications 508
8.1.6 White box testing 510
8.1.7 Dynamic binding versus RTTI 513
8.2 Static Binding and Class Methods 518
8.2.1 Hiding class methods 519
8.2.2 Semantics of static binding 522
8.2.3 Class methods versus instance methods 524
Chapter 9 Liskov Substitution Principle 529
9.1 Basic Semantics 530
9.2 Class Invariants 533
9.2.1 Additional class invariants 534
9.2.2 Strengthening class invariants 535
9.2.3 Strengthening restrictions on class properties 537
9.3 Postconditions 539
9.3.1 Additional postconditions 540
9.3.2 Inertia axiom 542
9.3.3 Strengthening postconditions 543
9.3.4 Effect clauses 547
9.4 Preconditions 549
9.5 Return Clauses 555
9.6 Exceptions 567
9.6.1 List of exceptions 568
9.6.2 Conditions for throwing exceptions 570
9.7 Return Types 579
9.8 Formal Arguments 582
9.9 Access Rights 587
9.10 Annotations 589

9.11 Black Box Testing 592