Index

©, 230. See also, Superimposition operator.

0-dimensional class constant, 49
0-dimensional signature constant, 50
0-dimensional superimposition term, 50
0-dimensional term, 27
1-dimensional class constant, 85
1-dimensional signature constant, 85
1-dimensional hierarchy constant, 90
1-dimensional superimposition term, 81, 85
2-dimensional superimposition term, 115

Application framework, 20, 133
Appropriate representation, 178
definition, 232
Array type, 49
Assignment, 183
definition, 331
Automated verifiability of Codecharts, 28
Automated verification, 188
AWT, see java.awt.
Axioms of class-based programs, 198, 231

B

Binary relation symbol, 57
Binary relation:
definition, 229
Blueprint metaphor, 9
Bridge pattern, 215
Building-blocks of object-oriented design, 22

C

Call relation symbol, 59
Changeability, 4
Clan, 81
of clans, 115
definition, 230
in a hierarchy, 93
in a set of classes, see Clan.
in a set of hierarchies, 112
Class
individual 49
set of, 75
Class-based programs:
axioms of, 198, 231

Abstraction lemma, 202
Abstract class, 54
Abstract Factory pattern, 154
vs. Factory Method pattern, 156
vs. Iterator pattern, 210
Widget Factory implementation of, 155
Abstract method, 54
Abstract relation symbol, 54
Abstract semantics, 169
Abstract semantics function, 174
Abstract Windowing Toolkit
See java.awt.
Abstraction, 12
in early design, principle of, 32
viz. exponentiation, 207
viz. generalization, 205
viz. information neglect, 202
viz. predicate weakening, 204
Ad-hoc symbols, 161
Adapter pattern, 213
Aggregate relation symbol, 58
All predicate, 97
definition, 231

Codecharts: Roadmaps and Blueprints for Object-Oriented Programs, by Amnon H. Eden
Copyright © 2011 John Wiley & Sons, Inc.
Class-based programming language, 22
Class constant:
  0-dimensional, 49
  1-dimensional, 75
Class Diagrams, see UML vs. Codecharts
Class of dimension 0, 171
definition, 229
Class of dimension 1, 175
definition, 229
Class hierarchy, see Hierarchy.
Class libraries, 19
Class relation, 171
Class-Z, 191
Closeable hierarchy, 24
Closed specification, see Specification, closed
Codecharts:
  history, 43
  motivation for, 3
  overview, 19
  reasoning about, 201
  vs. UML, 37
Composite pattern, 140
  UML vs. codecharts modelling of, 39
Complexity, 3, 13, 25, 104
Concretization, 24
Conformance, 4
Constant, 32. See also, Class constant;
  Signature constant; Hierarchy constant
definition, 230
  vs. variable, 129
Create relation symbol, 61
Data abstraction, 164
Decidability, 11
Decidability question, 11
Decorator pattern, 216
Design description language, 7
Design model, 175
definition, 230
Design motif 20, 127. See also, Design patterns;
  Application frameworks.
Design Navigator, 24. See also, Two-Tier Programming Toolkit
Design pattern, 139, 157
  instance of, 186
  reasoning about, 209
Design recovery, 44
Design verification 179
Dynamic binding, 21
Dynamically bound set of methods, 93

Early design, 159
Elegance, 16
Encapsulation, 21
Enterprise JavaBeans, 135
  UML vs. Codecharts modelling of, 41
Entity of dimension 0, 175
definition, 229
Entity of dimension k, 175
definition, 171
Exclusive predicate, 164
Extends keyword, 57
Facade pattern, 218
Factory Method pattern, 149
  java.util implementation of, 150
  vs. Abstract Factory pattern, 208
  vs. Iterator pattern, 203
Feynman-Tufte Principle, 30
Finite structure, 170
definition, 229
Flyweight pattern, 319
Formal specification languages, see
  Specification language.
Formula, 27
  ground, 231
definition, 231
  predicate, 231
  satisfaction of, 231
Forward relation symbol, 60
Full turing decidability, see Decidability.

Genericity in Codecharts, 32
Global function (C++), 50
Ground formula, 231

Hierarchy, 90
definition, 229
set of, 112
Hierarchy constant:
  1-dimensional, 90
  2-dimensional, 112
Hierarchy of dimension 1, 175
definition, 229
Higher-dimensional term, 27
I

Indirect relation, see Relation, indirect
Implementation minutia, 55
Implements:
  definition, 232
  keyword, 57
  a specification, 180
Inherit relation symbol, 57
Information neglect in Codecharts, 46
Information hiding, 164
Inheritance, 21
  transitive closure, axiom of, 199
Inheritance class hierarchy, see Hierarchy.
Instance of a design pattern, 186
Interface (Java), 54, 56
Interpretation function, 175
  definition, 230
Invisibility, 4
Isomorphic predicate, 83, 120
  example, 88, 97, 107
  definition, 231
  vs. Total predicate, 98
Iterator pattern, 145
  vs Abstract Factory pattern, 210
  vs. Factory Method pattern, 203

J

Java.awt:
  implementation of the composite pattern, 141, 207
  verifying conformance to the composite pattern, 188
Java.io, 24, 86. See also, Closeable hierarchy.
Java.util, 47, 72, 110, 112, 114
  implementation of the Iterator pattern, 146
  implementation of the Factory Method pattern, 150
JavaUtil design model example, 176
Java 3D, 103
  in UML, 14
Java RMI, see Remote Method Invocation.
Java SDK, 111, see also, Java.awt, Java.io, Java.util.
Java Servlets, see, Servlets.
JDOM, 101
JGraph, 16
JUnit, 136

K

Keyboard Safari Syndrome, 17

L

LeftExclusive predicate, see Exclusive predicate.
LePUS3. See also, Codecharts.
  history of, 44
Logic, LePUS3 in classical, 195

M

Mathematical logic, 195
Member relation, 171
Member relation symbol, 57
Method, 50
  dynamically bound set of, 93
  set of, see Clan; Tribe.
Method of dimension 0, 171
  definition, 229
Method of dimension 1, 175. See also, Clan, Tribe
Method, dynamically bound set of.
  definition, 229
Method signature, see Singature
Minimality of Codecharts, 33
Modelling, 7
Modelling language, 7, 12. See also,
  Design description language.
Modularity, 19
Motif, see Design motif.
Multiple-dispatch method, 53

O

Object-orientation in Codecharts, 19
Object-oriented application framework.
  See Application framework
Object-oriented programming languages, 22
Observer pattern, 220
OutputStream hierarchy, 25
Ontological question, 15
Open specification See Specification, open

P

Pattern, see Design pattern.
Pattern mining, 186
Portable Widget Factory. See Widget Factory.
Practice vs. theory, 9
Predicate, see All Predicate, Isomorphic Predicate, Total predicate.
Predicate formula, 231
Index

Primitive type, 49
Produce relation symbol, 61
Program Modelling process, 178
Programs:
  - modelling industry-scale, 109
  - modelling large, 71
  - modelling small, 47
Properties:
  - of a class, 53
  - of a method, 53
  - of sets, 97. *See also*, All predicate.
Proxy pattern, 221

R
Recursive composite pattern, *see*, Composite pattern.
Relation, 171
  - binary, 229
  - indirect, 64
  - simple, 56
  - unary, 171
  - transitive, *see* Relation, indirect.
  - turing-computable, *see* decidability.
Remote Method Invocation (RMI), 128
Return relation symbol, 61
RightExclusive predicate, *see* Exclusive predicate.
Rigour of Codecharts, 26
RMI. *See* Remote Method Invocation.
Roadmap metaphor, 8

S
Satisfaction, 185
  - ground formula, *definition*, 231
  - predicate formula, *definition*, 231
Satisfaction under assignment, 185
  - *definition*, 232
Satisfies relation, 180
  - *definition*, 232
Scalability, 13
  - of Codecharts, 30
Scaling question, 13
Schema, 191
Semantically entails, 201
Semantics, *see* Abstract semantics.
Servlets, 116
Signature, 50
  - set of, 85
Signature constant:
  - 0-dimensional, 50
  - 1-dimensional, 85
Signature of dimension 0, 171
  - *definition*, 229
Signature of dimension 1, 81
  - *definition*, 229
Signature-of relation, 171–172
Software design specifications, *see* specifications.
Software evolution:
  - first law of, 4
  - second law of, 4
Software modelling, *see* Modelling.
Software’s chronic crisis, 3
  - closed, 180
  - open, 183
Specification language, 7. *See also*, Design description language.
State pattern, 223
Strategy pattern, 223
Submodel, 175
  - *definition*, 232
Subtyping, 66
Superimposition operator, *definition*, 230
Superimposition term:
  - 0-dimensional, 50
  - 1-dimensional, 81, 85
  - 2-dimensional, 115

T
Template Method pattern, 224
Term, 27
Theory vs. practice, 9
Throw relation symbol, 61
Tool support in automated verification, *see* Automated verification.
Total predicate, 77, 118
  - *definition*, 231
  - example, 99, 102, 103
  - vs. Isomorphic, 98
Transitive relation symbol 64
Transitive closure. *See also*, Relation, indirect.
  - *definition*, 229
Transitive relation, *see* Relation, indirect
Tribe, 85
  - *definition*, 230
Tribe of classes, 115
Tribe in a class, 85
Tribe in a set of hierarchies, 115
Truth conditions, 181
TTP Toolkit, *see* Two-Tier Programming toolkit.
Turing-decidability, see Decidability.

Two-Tier Programming (TTP)
   Toolkit, ix, 16, 29, 44
   automated verification with, 188
   design recovery with, 24
   visualization with, 24

U

UML, 10, 14, 17
   shortcomings of, 134
UML quick reference, 233
UML vs. Codecharts, 37
Unary relation, 171
   definition, 229
Unary relation symbol, 53
Unified Modeling
   Language see UML
Universe, 171, 175
   definition, 229, 230

V

Variable, 27, 130
Variable vs. constant, 129
Verification, 179
   automated see Automated verification
Verification question, 11
Visitor pattern, 225
Visualization in Codecharts, 23

W

Well-formed formula, see Formula.
Widget Factory, 121
   implementation of the Abstract Factory pattern, 155