

Index

Symbols

# protected	28
*	39
– private	28
+ public	28
{abstract}	34
«anonymous»	40
«C-API»	34
«create»	39
«delegate»	40
«extends»	67
«function»	34
«generalization»	67
«interface»	5, 33
«local»	40
«parameter»	40
«persistent»	34
«struct»	34
«utility»	33, 34
«weak»	41

Numerics

0..*	39
0..1	39
1..*	39
3.141592653589793238462643383279502 (I know PI to 110 decimal places!)	34

A

abstract	
method	34
abstract class	27, 34
shorthand	35

abstraction	
imaginary	134
separation from details	147
acceptance tests	96
action	8, 116
activation	7, 46, 53
active object	59, 111
actor	46, 63, 65, 66
secondary	63, 65
tertiary	63
Acyclic Dependencies Principle	105
ADAPTER	19
adornments	21, 28, 33
ADP	105
aerospace	9
aggregation	18, 36, 37
dropped from UML 2.0	36
algorithm depiction	53
analysis	91
angled lines	53
Ann Marie Martin (my lovely wife)	xxiv
anonymous inner class	40, 189
architecture	96
high-level	104
arrow	4
dashed	30
direction of	29
message	7
name	29
number	29
return	46
star	29
with circle	7, 46
association	4, 18, 29, 36, 137
classes	41
horizontal	32
name	4
qualifier	42

- asynchronous message 55, 145
 - possible implementation 56
- AsynchronousLogger 60
- ATM 31

- B**

- Beck, Kent 99, 159, 240
 - Good object words.* xxiv
- behavior 64
 - first 16
 - hidden 64
 - partitioning 133
- big picture 113
- blank paper 63
- blueprints 10
- bold lines 113
- bold rectangle 113
- Booch, Grady xxiii, 34, 43
- Brooks, Alexis Josephine
 - (My granddaughter.)* v
- Brooks, Angela. (My daughter.) v
- Brooks, Matt. (My son-in-law.) v
- bubble sort 12
- bull pen 98
- business analyst 92
- business value 93, 94, 96
- button 16

- C**

- calling hierarchy 7
- card 92
 - index 63
 - use case 65
- CASE xxi, 22
- castles in the air 20
- CCP 104
- cellular phone 16
- chocolate candy 69

- class 28, 46
 - abstract 34
 - compartments 5
 - concrete 27
 - fat 87
 - god 135
 - helper 183
 - in sequence diagram 47
- class diagram 4, 18, 27, 31
- clean code 96
- clunky 53
- Cockburn, Alistair 66
- code
 - clean 96
 - expressive and readable 49
 - generation 23, 199
 - improving 98
 - never let the sun set on bad 98
 - seeing the. 20, 32
- coffee maker 129, 130
- collaboration 17, 18, 46
 - diagrams 7, 16
- colon 5, 28, 59
- COM 30
- Common Closure Principle 104
- Common Reuse Principle 105
- Comparable 5
- compartments 28
- component 103
 - icon 103
- COMPOSITE 40
- composition 18, 36, 37
- conceptual diagrams 1
- concrete class 27
- conditions in sequence diagram 53
- construction 7
 - arguments 7
- Construction phase 95
- contains relationship 102
- continual integration 98
- Coplien, James O. (Cope) 240
- coupling 72
- courage 98

- cover page 67
 - Craftsman 185
 - creation 47
 - crossed circle 40
 - crossed wires 136
 - CRP 105
 - customer 92, 93, 95, 96, 98
 - customer tests 96
 - CVS 98
 - cycles of dependencies 105
 - cyclomatic complexity 208
- D**
- dashed line 46
 - data token 7, 46
 - DATA TRANSFER OBJECT 172
 - database 73
 - facade 73
 - DECORATOR 40
 - deep copy 37
 - defect rate 96
 - degenerate method 86, 196
 - dependency
 - cycles 27
 - management 70, 158
 - relationship 40, 103
 - structure 27
 - Dependency Inversion Principle 86, 106, 141
 - depiction of algorithms 53
 - design 91
 - heuristics 129
 - mistakes 129
 - phase 22
 - principles 69
 - when to apply them. 89
 - quality 69
 - smells 70
 - destruction 37, 47
 - diagram
 - class 4, 18, 27, 31
 - conceptual 1
 - implementation 1
 - object 5, 109
 - package 107
 - physical 3
 - specification 1
 - state 7, 115
 - static 3
 - use case 66
 - diagrams
 - discarding 22
 - dynamic 3, 18, 20
 - incomprehensible 45
 - redundant and wasteful 49
 - simple and clean 21
 - useful 14
 - when not to draw. 22
 - when to draw. 22
 - Dialer 16
 - DIP 86, 106, 141
 - disciplines 91
 - documentation 22, 23, 45, 91, 97, 99
 - back-end 13
 - design 13
 - Martin's first law of 99
 - the value of. 24
 - dog 2
 - Don't Panic v
 - dot structure 7
 - downcasts 84
 - duct tape 70
 - dX* 91, 99
 - dynamic diagram 3, 18, 20
 - dynamic model 45
- E**
- Elaboration phase 95
 - embedded real-time system 132
 - engineers
 - aerospace 9

- structural 9
 - essence of OOD 148
 - estimation
 - calibration 95
 - use cases 65
 - user stories 92
 - event 7, 116
 - custom 117
 - entry 117
 - exit 117
 - use case 64
 - event-based system 115
 - evolution 20
 - evolving diagrams 20
 - exploration 91
 - Extreme Programming 91, 99
- F**
- FACADE 73, 207
 - Feathers, Michael 89
 - feedback 95
 - FileCarrier 183
 - finite state machine 7, 115
 - fitness, see also: <http://fitness.org> 97
 - FLIP-FLOP 83
 - folder 102
 - the force 67
 - Fowler, Martin 1, 8, 99, 240
 - Who has a music note on his head* xxiv
 - Fragility** 70, 71
 - Francis, Alan
 - (LSF)* xxiv
 - function
 - argument 28
 - declaration 28
 - functional decomposition 104
- G**
- Gamma, Erich 43, 240
 - garbage collector 37, 47
 - generalization relationship 2
 - generation, code 23
 - glitzy UI 94
 - god classes 135
 - goodies 45
 - grandchildren v
 - Grenning, James. (A good friend.) 45
 - guard 7, 51, 53
 - GUI 71, 73
 - isolation 74
 - guillemet 33
 - guilty 22
- H**
- half-arrowheads 55
 - has 29
 - HASA 29
 - Helm, Richard 43
 - heuristics
 - design 129
 - package 104
 - hidden behavior 64
 - hideous 18, 132
 - HTTPRequest 11
 - HTTPResponse 11
- I**
- ICE 123
 - icon
 - state 115
 - if statement 53
 - Illustrator. See Brooks, Angela v
 - imaginary abstraction 134
 - Immobility** 70
 - implementation 91
 - implementation diagrams 1
 - import 103
 - impossible task 63
 - impunity 98
 - Inception phase 95

incomprehensible diagram	45	
index card	63	
inheritance	30	
relationship	2	
vertical	32	
initial pseudo state	116	
inner classes	40	
anonymous	40	
instance variables	29	
instanceof	84, 86	
integration		
continual	98	
interface	5, 30, 33	
in sequence diagram	60	
shorthand	34	
Interface Segregation Principle	87, 105	
ISA	29	
ISP	87, 105	
italics	34	
iteration midpoint	94	
iteration planning	93	
iterative	20	
Iterative Development	91	
iterator	51	
J		
jar	101, 103	
Javadocs	24	
JDepend. See also: <i>www.clarkware.com</i>	105	
JDOM	71, 205	
Johnson, Ralph	43	
just in time requirements	63	
Justin Michael Martin (<i>will invent cold-fusion</i>)	xxiv	
K		
key	42	
kitchen garbage	70	
Koss, Bob (<i>Young Bob</i>)	xxiv	
L		
laboratory	98	
library paste	70	
lifeline	46, 47	
lifetime responsibility	37	
LinearObject	13	
link	5, 7, 18	
Liskov Substitution Principle	84	
List	29	
local variable	40	
Logger	60	
LoginServlet	11	
lollipop	30	
on lifeline	61	
loop condition	48	
loops in sequence diagrams	48	
LSP	84	
M		
Martin, Angelique Thouvenin (<i>My daughter-in-law.</i>)	v	
Martin, Gina (<i>My Daughter</i>)	xxiv	
(<i>Singer Extraordinaire</i>)	15	
Martin, Luka Jean. (<i>My grandson.</i>)	v	
Martin, Micah. (<i>My son.</i>)	v, 185	
Martin's First Law of Documentation	99	
Math	34	
McBreen, Pete	xxii	
meat	69	
rotting	69	
member functions	27	
member variables	27	
message	7, 46	
arguments	46	
arrow	7	
asynchronous	55	
label	7, 46	
synchronous	55	
that take time	53	
to interfaces	60	

- method 28
 - abstract 34
 - degenerate 86, 196
 - midpoint of iteration 94
 - minimalism 21
 - modeling 9
 - MODEL-VIEW-CONTROLLER 73
 - multiple threads 59
 - multiplicity 38
 - multithreaded systems 55, 111, 145
 - music note on Martin Fowler's head. ...xxiv
 - MVC 73
 - variation 75
- N**
- names, underlined 46
 - namespace 101
 - Needless Complexity** 70
 - Needless Repetition** 70
 - nitty gritty details 49
 - NULOBJECT 189
- O**
- object 46
 - creation 47
 - destruction 47
 - diagram 5, 109
 - name 5, 46
 - OCP 73, 84, 106
 - office
 - open 98
 - OOD
 - the essence of 148
 - training 129
 - OOPL 70
 - OOverkill 158
 - Opacity** 70
 - open office 98
 - Open-Closed Principle 73, 84, 106
 - organic chemistry 187
 - oval 67
 - overdocumenting 161
- P**
- package 101
 - abstractness 106
 - dependencies 103
 - dependency diagrams 107
 - design principles 104
 - icon 102
 - nesting 102
 - pair programming 96
 - perfect programming days 92
 - Persistable 72
 - phone call 54
 - physical diagrams 3
 - PI 34
 - plan 91
 - iteration 93
 - release 93
 - Point 13
 - Pollable 145
 - polling 145
 - PolyLine 13
 - polymorphism 70
 - postconditions 63, 65
 - practices 91
 - preconditions 63, 65
 - primary course 64
 - principles 20
 - design 69
 - OOD 69
 - package 104
 - priorities 92
 - private 28
 - process 22, 91, 99
 - productivity 96
 - properties 35
 - protected 28
 - PROXY 40
 - pseudo code 51
 - public 28

Q

QA	96	reuse	70, 104
manager	97	granule	104
qualifier	42	rework	96
quality assurance	96	Rigidity	70

R

race condition	54	RMI	83
Rational	xxiii	road maps	13
RDD	136	rocket science	184
realizes relationship	31	ROPE	185
real-time system	132	Ruby	185
rectangle	4, 28, 102	Runnable	113
bold	59	RUP	95
skinny. See activation.			
recurrence expression	51		
refactoring	98, 144		
relationship			
aggregation	18, 36, 37		
dropped from UML 2.0	36		
association	4, 18, 29, 36, 137		
composition	18, 36, 37		
contains	102		
dependency	40, 103		
generalization	2		
inheritance	2		
realizes	31		
use case	67		
release			
granule	104		
planning	93		
Release/Reuse Equivalency Principle ...	104		
REP	104		
requirements	91, 92, 96		
analysis	63		
document	97		
just in time	63		
responsibility	71		
driven design	136		
return value	28		

S

SAP	106
SAX	71
SCCS	98
scenarios	
failure	65
SDP	105
SELF SHUNT	79
separation of concerns	71, 147
sequence diagram	6, 7, 16, 45
angled lines	53
condition	53
crossed lines	54
high-level vs. low-level	49
interfaces	60
lollipop	61
loops	48
advanced	51
scenarios	49
the kind to avoid	49
thread identifier	59
sequence number	7
becoming unworkable	187
dot structure	7
Servlet	11
short cycles	20
Single Responsibility Principle	71, 104
SMC	121, 161
socket	111
SocketService	111, 186

- software architects 22
- Software Craftsmanship xxii
- source code control system 98
- source code dependencies 27, 30
- Space 13
- spaghetti code 70
- Sparky 2
- specification diagrams 1
- spike 93
- SQL 205
- SRP 71, 104
- Stable Abstractions Principle 106
- Stable Dependencies Principle 105
- star 16, 51
- STATE 122
- state
- final pseudo 119
 - icon 115
 - initial pseudo 116, 119
 - super 116
- state diagram 7, 115
- State Machine Compiler 121, 161
- state transition diagram 115
- state transition table 120
- static diagrams 3
- static variables 33
- stereotypes 33
- custom 34
 - listed in first page of index
- stick figure 6, 46
- stimulus 64
- story points 92, 94
- subway turnstile 7, 120
- superstate 116, 118
- synchronous message 55
- system behavior 64
- system boundary diagram 66
- T**
- tab 102
- tangle 13
- task 94
- team 98
- sitting and spinning 63
- test 91
- acceptance (See acceptance tests)
 - unit (See unit tests)
- test-first 83, 97
- Thread 113
- thread 55, 56
- identifier in sequence diagram 59
 - internal 59
- THREE-LEVEL FSM 196
- thrill xxiii
- transition 8, 116
- reflexive 117
- Transition phase 95
- tricky things 134
- turnstile 7
- U**
- UI
- glitzy 94
- UML xxiii, 1
- language lawyer 42
 - method, not 99
 - misuse of 9
 - not using it 161
 - overuse of 9
 - perversity 55
- UML Distilled xxi, 8
- Uncle Bob xxiii
- underline 5, 46
- underlying problem, the 136
- Unified Process 95
- unit tests 73, 78, 83, 89, 97, 162
- use case 63, 92, 96, 137
- alternate course 65
 - capturing them all 63
 - card 63, 65
 - details 63
 - diagrams 64, 66
 - estimation 65
 - failure scenarios 65

name	65	Mitchell, William H.	
postconditions	65	(Formerly known as "Red".)	xxiv
preconditions	65	wind tunnels	9
primary course	64	wires	
relationships	67	crossed	136
simplicity	63	wisdom	49
tomorrow they change	63	www.objectmentor.com	71
vastly overcomplicated	63		
writing	64		
user story	92, 96		
estimating	92		
merging	93		
points	92		
splitting	93		
utility	34		

X

X on lifeline	47
XML	71, 205
XP	91, 99

Z

zone	32
------------	----

V

vapor classes	134
variable	28
declaration	28
static	33
Vector	29
velocity	93, 95
Viscosity	70
VISITOR	176, 193
Vlissides, John	43
Vogel, Billy	xxiii

W

war room	98
waste of time	45
expensive	61
huge	49
weak reference	41
Welter, Lance (<i>sales creature</i>)	xxiv
whatchamajiggers	42
white board	13, 14, 20, 62, 94, 99, 129
whole/part	36
wiki	24