

## Index

- abscissa values 339
- absolute value 34
- abstract classes 44, 46–47, 53, 64–65, 517, 742
- Abstract Data Type (ADT) 126, 654
- Abstract Factory* 494, 508
- abstraction 152
- abstract methods 53, 414–415
- access control systems 511
- accrediting IRS 401
- accrual period 169
- accrued interest 159, 169
- accuracy 707
- actions 506
- Activator 284
- adapters 409, 494
- additive process 221
- ADE *see* Alternating Direction Explicit method
- ad-hoc polymorphism 510
- ADT *see* Abstract Data Type
- advanced lattices 242–270
- AggregateException 701
- aggregate systems 511
- aggregation 22, 494, 739–740
- aggregation/composition 53, 82, 511
- aggregation methods 526, 533
- Akima method 335, 338, 342, 348–349
- ALGLIB 756–758
- algorithmic trading systems 709
- algorithms 495, 501
- Alternating Direction Explicit (ADE) method 241, 255–268, 504, 765–788
- Amdahl's law 708
- American options 160, 264
  - ADE method 262–263
- amortising 797–798
- amortising cap 463–464
- amortising IRS 401
- analogical reasoning 215
- annuity 162
- anonymity 388
- anonymous methods 77
- API *see* application programming interface
- AppDomain 308
- application domains 304–309
- application manifest 273
- application programming interface (API) 277
  - code generation 298–304
- applications 707
- approximation, function derivatives 341–342
- arbitrage-free conditions 335
- arbitragers 369
- Arithmetic Mean Method 341
- arithmetic operators 13–14
- ArrayGenerator 73, 75
- ArrayList 123
- arrays 97–101, 131–135
- Array<T> 131, 135
- ASCII characters 12
- assemblies 273–276, 304, 309–310
- AssemblyInfo.cs 275–276
- assembly manifest 273
- asset price 241–246
- assets 371
- asset swap 429, 455
- asset swap spread 455
- assignment operators 13
- association 494, 740
- associative arrays 142–144, 152–155
- associative containers 142
- associative matrices 144
- asymptotic accuracy 168, 335
- asynchronous delegates 651, 679–681
- asynchronous events 708
- asynchronous methods 710
- ATM *see* at the money
- ATMF *see* at the money forward
- at the money (ATM) 461, 485–487, 503
- at the money forward (ATMF) 466
- AtmStrikeBuilder 503
- atomicity 652

- at par 397
- attributes 275, 284, 285, 304
  - see also* custom attributes
- automated trading systems 709
- automatic garbage collection 728–730
- automatic property 550
- Automation 567–568
- automation client 581
- automation server 581
- AutoResetEvent 676
- 
- background threads 684–685
- BackgroundWorker 685
- backing store classes 186–187
- backing store streams 185, 187–189
- back office requirements 388
- backslash 12
- backward induction 226, 233, 243
- backward Kolmogorov equation 269
- Backward in Time Centred in Space (BTCS) 254
- bag 535, 695
- banded matrices 99
- Barakat-Clark scheme 256
- barbershop problem 705
- barrier 698–699
- barrier options 786
- BaseBond 313, 320
- base class abstract method 411–412
- base class constraint 129
- base class constructor 411
- base (generalised) class 44
- BaseOneDimensionalInterpolator 409, 439
- base priority 637
- basic level 64, 745
- basis 433–434, 467
- basis swap 401
- BBA *see* British Bankers Association
- Beep() 14
- behaviour 748
- behavioural patterns 495, 500–501
- benchmark bond 330
- Bermudan option 160
- best fit methods 405, 502
- bilinear interpolation 335, 352–355, 364–366
- binary association 740–741
- binary formatter 201
- BinaryReader 191
- binary semaphore 674
- binary serialisation 203
- BinaryWriter 191
- binding 513
- BinomialLatticeStrategy 223
- binomial method 215–240, 241, 574
- BinomialMethod 226
- binomial parameters 219–228
- 
- Bisection method 317, 463
- BitArray 114
- bitwise operators 13
- black boxes 512
- Black formula 459
- Black model 374
- Black-Scholes formula 35–36, 83, 246–247
  - implementing ADE 258–262
- blocking 695
- block matrices 712
- blocks 646
- Bond 178
- bond data management 321–324
- bond description 329
- bond details 328–329
- BondDictionary 322
- bond functionality 165–166, 313–317
- bondholder *see* lender
- bond indenture 159
- BondModel 56
- bond models 55–58, 92
- bond portfolios 145
- BondPricer 178
- bond pricing 311–333
- bond rating 328
- bonds 159–184, 542–543
  - risks 181
  - scenarios 543–544
- bond scheduling 312–313
- bool 13, 114
- bootstrapping 437–438
- bootstrapping volatility 469–474
- borrower *see* issuer
- boundary conditions 769–771, 774
- bounded blocking collection 719
- bounded queues 695
- bounds checking 101
- Box-Muller algorithm 719
- branching constructs 303
- Brennan-Schwarz algorithm 233
- bridge 494
- Bridge design pattern 145, 507
- British Bankers Association (BBA) 371
- broadcaster 87
- BTCS *see* Backward in Time Centred in Space
- buffered streams 187
- Builder design pattern 494, 496–499, 502–503
- BuildingBlock 408
- built-in data types 10, 11
- bullet 401
- bullet bond *see* plain vanilla bond
- business day 400
- 
- C#
  - advanced features 53–95
  - fundamentals 9–24

- generics 125–127
- implementation 36–39
- properties 26, 50
- string type 117
- C# classes 25–52, 170–174
  - see also* classes
- C++ 9, 289–298
- CAD *see* computer aided design
- calculate() 21, 260
- calculateBC() 260
- calculator 85, 585–589
- Calendar 101
- calibration 395, 418, 445–446
- call feature 160
- call options 37
  - sensitivities 51
- call payoff behaviour 777
- call put parity 466, 794–795
- call risk 181
- capacity 674
- CapBlack 482–483
- CapletBlack 476
- caplet price 481–482
- caplets 387, 459
- caplet volatility 462, 470, 502, 792–794
- cap price matrix 795–797
- caps 459–461
  - multi-curve framework 467–469
- cap stripping 469–473, 476
- captured variables 77
- cap volatility 462–463, 789–792
- CAS *see* compare and swap
- cash-based securities 370
- cash flow 113, 145
  - aggregation 545–546
- cash money market 369, 370
- casting 65–67, 79, 126
- categorisation 745
- causing a delay 683
- CCPs *see* clearing houses
- CCS *see* cross currency swap
- CCW *see* COM Callable Wrapper
- CDS *see* Credit Default Swap
- CEV *see* Constant Elasticity of Variance
- chaining decorators 527
- chaining query operators 527
- chain of responsibility 495
- Change 682
- character literal 117
- characters 12–13
- charm 51
- charts 565
- child objects 205
- child thread 641
- chunks 711
- CIR *see* Cox-Ingersoll-Ross
- classes 9, 53–95
  - abstract 44, 46–47, 53, 64–65, 517, 742
  - Activator 284
  - AppDomain 308
  - ArrayGenerator 73, 75
  - Array<T> 131, 135
  - AtmStrikeBuilder 503
  - AutoResetEvent 676
  - BackgroundWorker 685
  - Barrier 698
  - BaseBond 313, 320
  - BaseOneDimensionalInterpolator 409, 439
  - BinaryReader 191
  - BinaryWriter 191
  - BinomialLatticeStrategy 223
  - BinomialMethod 226
  - BondDictionary 322
  - BuildingBlock 408
  - Connect 599
  - Console 14–15, 273, 280
  - ConsoleEuropeanOptionFactory 17–18
  - Convert 14
  - CountryInfo 550
  - Datasim.DataInterpolator 273
  - Datasim.DataStructures 273
  - Datasim.DataTime 273
  - derived 44, 45, 412, 742
  - DeserializationBinder 324
  - Director 497, 503
  - Directory 197
  - DirectoryInfo 197, 198–199
  - Draw() 45
  - DriveInfo 197
  - DynamicMethod 299
  - EventArgs 88
  - EventHandler 88
  - EventInfo 282
  - FDM 249
  - FieldInfo 282
  - File 196
  - FileInfo 197, 198–199
  - FileSystemInfo 197
  - FloatingRatePlusSpread 332
  - FloatingSchedule 178, 313
  - Formula 474
  - Hashtable<T> 110
  - HashSet<T> 112–114, 140
  - InterpAdapter 409
  - ListedContSpec 383
  - ListedSTFut 384
  - ListedSTFutOption 384
  - locking mechanisms 667–673
  - ManualResetEvent 676
  - Matrix<T> 131, 134, 271

classes (*Continued*)

- Meshor 259
- MethodInfo 282, 299
- Module 277
- Monitor 667, 678
- MonoStrikeBuilder 503
- MonoStrikeCapletVolBuilder 476
- MonoStrikeCapletVolBuilderBestFit 477
- MultiCurveBuilder 441–442, 450–451, 451–452, 503
- NumericMatrix<T> 131, 137, 168, 289
- OneStepFDM 70
- OpCodes 299
- Option 17, 18–20, 38, 42, 217–219, 282
- Parallel 687
- Path 197
- Point 16, 22, 30, 78, 97, 273
- Process 636, 637
- ProcessStartInfo 637
- ProgrammableBase 319
- PropertyInfo 282
- Random 725
- RateSet 407, 408
- ReaderWriterLock 715
- Schedule 178, 311
- Shape 46
- SingleCurveBuilder 406, 410
- SingleCurveBuilderInterpBestFit 413
- SingleCurveBuilderSmoothingFwd 625
- Special Functions 20, 22
- StopWatch 726–727
- System.Reflection 273, 277, 299
- Task 691–692
- Tensor<T> 131, 139–140
- TextReader 191
- TextWriter 191
- ThreadPool 651
- Type 278
- Vector<T> 131, 135–136, 289
- XLBond 319, 321, 324
- class hierarchy 196–199, 313–317, 517
- class hierarchy integration 333
- class instances 25, 735
- class-level binding 74
- class methods 25–28, 30–33
- class variables 30–33
- clean price 160
- clearing houses (CCPs) 400, 432
- client code 121
- clients 26, 68, 82, 210, 509
- clock creation 683
- Clone() 13, 62–63
- closing an open position 388
- CLR *see* Common Language Runtime
- CMS *see* Constant Maturity Swap
- coarse grained locking 670
- coarse-grained parallelism 707
- coarse-grained pattern 711
- code emission *see* code generation
- code generation, API 298–304
- code integration 328–329
- code readability 41, 524
- collateral 400
- collateralisation 432–433
- collateral support annex (CSAs) 400
- collection interfaces 107–109
- collection libraries 97
- collections 525–526
  - ordering 546–547
  - serialisation 206–207
  - synchronising 681–682
- COM *see* Component Object Model
- COM Callable Wrapper (CCW) 569, 582
- command 495
- commodity swap 430
- Common Language Runtime (CLR) 284–285
- comparand 652
- comparative advantage 399
- compare and swap (CAS) 652
- compatibility conditions 255
- compiled code 273
- component diagrams 82
- Component Object Model (COM) 567, 581–583
  - add-ins 595–623
  - interoperability 121–122, 569
  - object model 561–562
  - optional parameters 591–592
- composite pattern 494
- composition 67–68, 82, 125, 494
  - strategies 532–533
- compounding swap 401
- comprehension queries 526, 528–529
- compression streams 187
- ComputableFunction() 73, 75
- computation 17
- computational finance 557
- computer aided design (CAD) 511
- concept levels 745
- concept modelling 743–745
- concepts, explanation-based view 44
- concrete classes 742–743
- concurrency 707
- concurrent data structures 694–701
- Conditional 285
- conditional consistency 776–777
- configurable payoff functions 270
- configuration data 186
- Connect 599
- Console 14–15, 273, 280

- ConsoleEuropeanOptionFactory 17–18
- console input/output 14–15
- consolidation algorithms 511
- Constant Elasticity of Variance (CEV) 264, 474
- Constant Maturity Swap (CMS) 401, 459
- constraints 129, 145–152
- constructors 25, 226
- consumers 509, 654
- context class hierarchy 58
- context diagram 512
- continuations 694
- continuously compounded rate 336
- contract 125
- contract size 372, 383
- control point 338
- convection-diffusion equation 253
- convection-diffusion-reaction 248, 767
- convection dominance 246, 254
- convection-dominated problems 264
- convection terms 246, 773–774
- convergence 246
- conversion 497
- conversion methods 525
- conversion operators 540
- Convert 14
- convertible bond 160
- convexity 168, 335, 349
- convexity adjustments 372
- convoy problem 717
- coordinator 625
- Copy () 13
- copy constructor 25
- correlation matrices 145
- counterparty risk 371, 373, 388, 400, 432
- counting semaphore 674
- CountryInfo 550
- coupon 159
- coupon rate 159
- coupon types 159, 311
- Cox-Ingersoll-Ross (CIR) model 54, 770
- Crank Nicolson method 69, 256, 504, 507, 766
- creational patterns 496
- Credit Default Swap (CDS) 430
- credit risk *see* default risk
- Credit Support Annex (CSA) 432, 467
- Credit Value Adjustment (CVA) 432
- critical sections 305
- cross currency swap (CCS) 430
- crypto streams 187
- CSA *see* Credit Support Annex
- CSAs *see* collateral support annex
- cubic spline interpolation 335, 338, 343–344, 361–364
- culture 273
- cumulative distribution functions 20
- currency 329, 384
- currency risk *see* exchange-rate risk
- current position 186
- curvature 168
- curve building 335–337, 758–759
- curve construction 403–405
- curve shapes 337–338
- custom attributes 286–289
- custom binding 513
- customisation 388
- CVA *see* Credit Value Adjustment
- databases 523
- data containers 125
- data contract model 199
- DataContractSerializer 199
- data count 159
- data decomposition 510, 746–748
- data flow 497
- data lifecycles 185–186
- data management 185, 209–214
- data parallelism 187
- data presentation 319–321
- data replication 715
- DataSchedule 174–176
- data sharing 307–309
- Datasim.DataInterpolator 273
- Datasim.DataStructures 273
- Datasim.DateTime 273
- Datasim Visualisation tool 419–421
- data source object 83
- data storage devices 186
- datastore 211–212
- data structures 97, 125–158, 182, 572
- data values *see* function values
- data viewpoint 506
- data visualisation 570–578
- date rolling convention 169, 170
- dates 101–105
  - IMM 378–384
- DateTime 101, 170
- DateTimeOffset 101
- day count 159, 311, 400
- day count conventions 168, 169
- day rolling conventions 169
- D-curve 467
- debtor *see* issuer
- Debt Value Adjustment (DVA) 432
- decimal 10
- Decorator pattern 50, 494, 507
- decorators 532
- decorator streams 187
- deep copy 62
- default constructor 25
- default risk 181
- default value 591
- deferred execution 526, 529–531

- defining attribute view 735, 743
- delay causing 683
- delegate instance 72, 80
- delegate mechanisms 53
- delegates 53, 72–76, 87, 124, 517
  - behavioural design patterns 509–510
  - .NET 80–87
  - plug-in methods 82–85
- delegate types 72, 86–87, 301–304
- delta decay *see* charm
- delta hedge portfolio 391
- delta values 380
- denominator 352
- dependent variable 339
- deposits 337, 394
- dequeue 115
- derived (specialised) classes 44, 45, 412, 742
- deserialisation 185, 199, 322
- DeserializationBinder 324
- design patterns 53, 155, 217–232, 384, 493, 494
- deterministic 666
- diagonal rational functions 352
- dictionaries 116, 321
- diffusion term 265
- Director 497, 503
- Directory 197
- directory classes 195–199
- DirectoryInfo 197, 198–199
- Dirichlet boundary condition 257
- dirty price 160
- discontinuous payoff functions 256
- discount bond 336
- discount curve 331
- discount factors 317–319, 394, 453, 540–542
- discounting 162
- discounting curve 437, 467
- discrete derivatives 367
- discrete slopes 339
- display 17
- disposal 727
- Dispose 682
- dividend swap 429
- division by zero 12
- division of labour 685
- DLR *see* Dynamic Language Runtime
- Do () 305
- Dodd-Frank Act 432
- domain architecture 215
- domain decomposition 712
- domain transformation 779–780
- domain truncation 780
- double 10
- Draw () 45
- drift 265
  - see also* convection term
- DriveInfo 197
- dual-currency linked bonds 183
- dual interface 582
- duration 168
- DVA *see* Debt Value Adjustment
- dynamic 514
- dynamic assembly loading 284
- dynamic binding 122, 513
- dynamic instability 254
- Dynamic Language Runtime (DLR) 514
- DynamicMethod 299
- dynamic method invocation 283, 309
- dynamic objects 283–284, 516
- dynamic programming 513–516
- dynamic state 210
- dynamic viewpoint 506
- early binding 582
- EBF *see* European Bankers Federation
- EF *see* Entity Framework
- effective date 400
- effectiveness 389
- efficiency 196, 504
- elapsed time 645
- elasticity factor 265
- element operators 525
- embarrassingly parallel 707, 715
- embedded optionality 160
- empty string 118
- end date 174
- end-of-line character 14
- enqueue 115
- Entity Framework (EF) 549
- entry assembly 277
- enumeration 105–107
- enumeration types 40–41
- EONIA *see* EUR Overnight Index Average
- equality operations 110
- equally spaced 338
- equity pricing, ADE 256–258
- error handling 124
- escape sequence 117
- Euro Interbank Offered Rate 370
- European Bankers Federation (EBF) 371
- European option 160
- European option price *see* option pricing
- EUR Overnight Index Average (EONIA) 403, 433
  - rates replication 547–549
- evaluation stack 299, 300–301
- event 88
- EventArgs 88
- EventHandler 88
- event handler code 601
- EventInfo 282
- event wait handles 676

- Excel 216
  - add-ins 319–321, 566–569
  - automation 581–594
  - bond pricing 324–328
  - data visualisation 570–578
  - exporting schedulers 176–177
  - integration 561–579
  - lattice presentation 230–232
  - LINQ queries 549–557
  - real-time data 625–626
  - standalone C# application 564–566
- Excel-DNA 603–615
- ExcelMechanisms 572–575
- exception handling 50, 304, 572–575, 653–654, 701–702
- exchange 652
- exchange-rate risk 181
- exclusive locking 667
- execution path 666
- execution speed 305
- execution time 708
- exemplar-based view 744
- expectation theory 337
- expiration date *see* maturity date
- expiry/tenor 473
- explanation-based view 44, 744–745
- explicit Euler finite method 69, 721
- explicit finite difference methods 241, 246–258
- explicit interfaces 65
- explicitly downcast 79
- explicit serialisation 203
- exponential fitting 246, 254
- exponentially fitted schemes 265
- expression tree 76
- extension methods 38, 40–44, 51, 54, 495
- extensions 508, 735
- facade pattern 494, 499
- factories 16, 55
- Factory method 215, 408, 494, 508
- factory objects 494, 495
- fair rate 396
- FDM 249
- FDM, *see also* Finite Difference Method
- FDMDirector 251
- feature 82
- Fibonacci generator (lagged) 720
- Fichera function 257
- FieldInfo 282
- FIFO *see* First In First Out
- File 196
- file access 188
- file classes 195–199
- FileInfo 197, 198–199
- file mode 188
- file path name 188
- file streams 186–187
- FileSystemInfo 197
- filtering 341, 524
- finalisers 25, 32, 78, 727
- fine grained locking 670
- fine-grained parallelism 707
- Finite Difference Method (FDM) 53, 241–270
  - ADE method 262–263
  - Black-Scholes equation 246–247, 258–263
  - explicit schemes 246–268
  - trinomial models 241–246
- first-exit time for diffusion processes 269–270
- First In First Out (FIFO) data structure 115
- fixed in advance and paid in arrears 371
- fixed income applications 540–549
- fixed rate 400, 401
- fixed rate bond 159
- fixing date 174
- flat results set 535
- Flatten 702
- flat volatility 462
- flat yield curve 338
- float 10
- floating rate 400
- floating rate bond 331
- floating rate note (FRN) 159
- FloatingRatePlusSpread 332
- FloatingRateSchedule 178, 313
- FloorletBlack 476
- floorlets 387
- floors 459
  - multi-curve framework 467–469
- floor volatility 462
- flyweight 494
- following business day 170
- foreground threads 684–685
- foreign exchange (FX) risk *see* exchange-rate risk
- Forex swap 430
- forked 709
- fork and join pattern 709–711, 731
- format 103
- format strings 103
- formatters 201–202
- Formula 474
- forward induction 219, 226, 243
- forwarding curve 437, 467
- Forward Monotone Convex Spline 335
- forward premium 466
- forward rate agreement (FRA) 387, 396–397, 427
- forward rates 335, 336, 372, 394, 419, 436, 467
- forward rates curve 336
- forward start IRS 402
- forward start swaps 421
- forward swap matrix 447–448
- Forward in Time Centred in Space (FTCS) 253
- foundation classes 289

- FP *see* functional programming  
 FRA *see* forward rate agreement  
 fragmented 729  
 frequency 400  
 Fritsch-Butland Method 341–342  
 FRN *see* floating rate note  
 FTCS *see* Forward in Time Centred in Space  
 full matrices 98  
 fully qualified name 272  
 functional decomposition 511  
 functionality 92, 504  
 functional programming (FP) 77, 493  
 functional viewpoint 506  
 function derivatives, approximation 341–342  
 function overloading 510  
 function values 339  
 future rate 372, 397  
 FX *see* foreign exchange  
 FX-linked foreign currency coupon bond 183
- GAC *see* Global Assembly Cache  
 gamma trading strategies 709  
 garbage collection 727–730  
 garbage collector (GC) 728  
 Gaussian distribution 20, 34  
 GBM *see* Geometric Brownian Motion  
 GC *see* garbage collector  
 generalisations 508–509, 743  
 generational mark-and-compact 728  
 generation methods 526  
 generations 729  
 generic classes 410–412, 550  
 generic collections 681  
 generic constraints 129  
 generic delegates 86, 128–129, 157  
 generic methods 128–129, 157, 304, 509, 524  
 generic programming (GP) 493, 737–738  
 generic queue 156–157  
 generic type 126  
 generic vectors 301–304, 515  
 Geometric Brownian Motion (GBM) 242  
 geometric decomposition 711–715  
*Geometric Decomposition* pattern 712  
 German bonds 315  
 GetILGenerator 299  
 get it right phase 51, 241  
 get it working phase 241  
 Global Assembly Cache (GAC) 275, 563  
 global interpolation 336, 339, 350–352, 413–415  
 Globally Unique Identifier (GUID) 581  
 GOF design pattern 494–496  
 government bonds 330  
 GP *see* generic programming  
 the Greeks *see* option sensitivities  
 grid 339  
 grouping 525
- GroupJoin 535–540  
 GUID *see* Globally Unique Identifier
- Hagan-West approach 335, 338, 349–350, 355  
 Handle 702  
 hash code 110  
 Hashtable<T> 110  
 HashSet<T> 112–114, 140  
 heap-based memory management 9  
 heap memory 10  
 hedgers 369  
 hedge sensitivities (greeks) 233  
 HelloWorld 9, 585, 604  
 Hermite cubic interpolant 335, 338, 344  
 hierarchical result set 535  
 higher-dimensional structures 139–140  
 high-frequency trading 709  
 hooks 259, 501  
 horizontal slabs 712  
 host application 581  
 humped yield curve 338  
 Hyman filter 335  
 Hyman quantic interpolation 338
- IAsyncResult 680  
 IBondModel 55  
 IBondPricer 178  
 IBSPde 248  
 IBVPFDM 259  
 ICA *see* Interface Connection Architecture  
 ICalculateDistance 280  
 ICalculator 319  
 ICapletVolMatrixBuilder 503  
 ICloneable 61, 133  
 ICollection<T> 107  
 ICouponProvider 331  
 IDictionary 108, 111–112  
 IDispatch 568  
 IDisposable 105  
 IDTEExtensibility2 596–597  
 IEnumerable 106, 119, 523  
 IEnumerator 105–106  
 IExcelAddIn 608  
 if-else statement 37  
 IFormatter 201  
 IID *see* interface ID  
 IL *see* Intermediate Language  
 IListedInstrument 383  
 IList<T> 108, 109–110  
 IMatrixAccess<T> 149  
 IMM *see* International Monetary Market  
 immutable 12, 117  
 implementation 152  
 implementation details 47  
 implementation inheritance 53  
 implicit Euler method 69, 255

- implicitly upcast 79
- implicit serialisation 203
- implied rate 167, 372
- implied volatility 463, 731
- IMultiRateCurve 438
- independent variable 339
- index sets 557
- inflation risk 181
- inflation swap 430
- inheritance 44–46, 51, 82, 93, 494, 501
- Initial Boundary Value Problem 259
- inner queries 531
- input interpolation 487
- Insert () 119
- instantaneous forward curve 336
- instantiating objects 304
- integer counter 77, 674
- integers 12
- integrating factoring method 767
- intension 735
- Interbank Offered Rate Euribor 370
- interest *see* coupon
- interest rate applications, interpolation methods 335–368
- InterestRateCalculator 160
- Interest Rate Curve (IRC) 331, 394
  - building blocks 395–397
  - code design 406–418
- Interest Rate Multi-Curve (IRMC) 437
- interest rate options 459
- interest rate risk 181, 370
- interest rates derivatives valuation 431–436
- Interest Rate Single Curve (IRSC) 437
- interest rate swap (IRS) 397
  - cash flow 398–399
  - contract specification 399–402
- Interface Connection Architecture (ICA) 26, 82
- interface ID (IID) 581
- interface inheritance 53
- interfaces 17, 46–50, 53–54, 87, 94–95, 517
  - collection 107–109
  - explicit 65
  - generic parameters 130
  - IAAsyncResult 680
  - IBondModel 55
  - IBondPricer 178
  - IBSPde 248
  - IBVPFDM 259
  - ICalculateDistance 280
  - ICalculator 319
  - ICapletVolMatrixBuilder 503
  - ICloneable 61, 133
  - ICollection 107
  - ICouponProvider 331
  - IDictionary 108, 111–112
  - IDispatch 568
  - IDisposable 105
  - IDExtensibility2 596–597
  - IEnumerable 106, 119, 523
  - IEnumerator 105–106
  - IExcelAddIn 608
  - IFormatter 201
  - IList 108, 109–110
  - IListedInstrument 383
  - IMatrixAccess 149
  - implementing 69–72
  - IMultiRateCurve 438
  - IOptionFactory 18
  - IRateCurve 439, 476
  - ISingleRateCurve 406, 439
  - ITwoFactorPayoff 47, 53
  - IXmlSerializer 204, 207–209
  - model problem 68–69
  - .NET 61–67
    - object casting 65–66
    - properties 63–64
    - pure discount bonds 55
    - standardisation 145–152
  - interlocked instructions 652
  - Intermediate Language (IL) code 273
  - Internal Rate of Return (IRR) 167
  - International Monetary Market (IMM) 369, 378–384
  - International Swaps and Derivatives Association (ISDA) 399
  - interoperability 61, 289–293
  - InterpAdapter 409
  - interpolant 339
  - interpolation 405, 577
    - methods 335–368
  - interpolators 338–339, 409
  - interpreted queries 523, 533
  - Interpreter design pattern 495, 499
  - into 532
  - intrinsic shape 340
  - invariant code 221
  - invariants 501
  - inverted yield curve 338
  - investor *see* lender
  - IOptionFactory 18
  - IRateCurve 439, 476
  - IRC *see* Interest Rate Curve
  - IRMC *see* Interest Rate Multi-Curve
  - IRR *see* Internal Rate of Return
  - IRS *see* interest rate swap
  - IRSC *see* Interest Rate Single-Curve
  - ISDA *see* International Swaps and Derivatives Association
  - ISIN code 328
  - ISingleRateCurve 406, 439
  - isolated storage file streams 187
  - issue amount 329

- issuer 159
- issuer name 329
- Italian bonds 316
- Italian Government bonds 333
- iteration variable 528
- iterative methods 502
- iterative scheme 752
- iterators 107, 495
- ITwoFactorPayoff 47, 53
- IXmlSerializer 204, 207–209
  
- Jacobi iteration 709
- jagged arrays 98–101
- Java 9
- JIT *see* Just-In-Time
- join 525
- Join 535–540
- joined 709
- Just-In-Time (JIT) compiler 298
  
- keywords
  - delegate 72, 77
  - dynamic 122, 514
  - into 532
  - lock 667
  - namespace 271
  - 'new' 33
  - 'this' 28
  - var 80, 524
  - volatile 644
- Kruger Method 341
  
- label 383
- lagged Fibonacci generator 720
- Lagrange interpolation 338
- lambda expression 76, 527
- lambda functions 53, 72–77, 693
- lambda queries 526
- Language Integrated Query (LINQ) 212, 299, 393, 523–559
  - advanced queries 531–533
  - aggregation methods 533
  - Excel interoperability 549–557
  - fixed income applications 540–549
  - queries 526–531
  - query operators 524–526
  - set operations 535
- Large Object Heap (LOH) 729
- Last In First Out (LIFO) data structure 114, 126
- late binding 122, 582
- late binding clients 584
- lattice presentation 230–232, 575
- Lawson extrapolation technique 256
- layers pattern 499–500
- lazy initialisation 529
- legal risk *see* political risk
- legs 400, 401
- Leisen-Reimer method 239
- lender 159
- Levenberg-Marquardt method 393, 405, 438, 477, 758
- leverage 372, 388
- liability 370
- Libor credibility 337
- lifecycle systems 511
- LIFO *see* Last In First Out
- lightweight 638
- linear spline interpolation 335, 338, 342–343
- LineSegment 22, 23
- LINQ *see* Language Integrated Query
- liquidity 388
- liquidity premium theory 337
- liquidity risk 181
- listed instruments 383–384
- load 300
- load balancing 712
- LoadBondFixedCoupon 325
- loading 613–614
- locally concave 339
- locally convex 339
- locally monotonic 339, 349
- local methods 339
- local object collections 523
- local query 523
- local variables 303
- lock 667
- lock-free 719
- lock granularity 669
- locking mechanisms 665, 667–673
- locks 305
  - see also* mutex
- logarithm 34
- logical operators 14
- logical units 273
- LOH *see* Large Object Heap
- long 12
- long-end 337, 338
- lookup tables 152–155
- loop-level parallelism 712
- loose couplings 17, 495
- lower triangular matrices 98
- LU matrix decomposition 709
  
- Macauley duration 167–168
- Main() 9
- maintainability 196, 504, 524
- main thread 641
- major client 85, 510
- managed memory leaks 727, 730
- management information systems 511
- ManualResetEvent 676
- manufacturing domain architecture 497, 511

- manufacturing systems 511
- marching scheme 261
- margin 401
- market inputs 394
- market price 394
- market segmentation theory 338
- mark-to-market 436, 448–450
- MarshalByRefObject 308
- master 625
- master thread 685
- Master-Worker parallel design pattern* 642
- Math 21
- mathematical functions 34
- matrices 131–135, 301–304, 343
- Matrix<T> 131, 134, 271
- maturity date 159, 311, 328, 387, 400
  - see also* termination date
- maximum 34
- maximum principle 253
- Mediator* pattern 17, 21, 216, 228–230, 495
- Memento* design pattern 495
- memory leaks 730
- memory management 10
- memory streams 187, 213
- Mersenne-Twister generator 720
- Mesh 259
- message passing 494, 665, 748
- metadata 276–289
- MethodInfo 282, 299
- Method of Lines (MOL) 777
- method overloading 92
- methods
  - calculate() 21, 260
  - calculateBC() 260
  - CapBlack 482–483
  - CapletBlack 476
  - Clone() 13, 62–63
  - ComputableFunction() 73, 75
  - Copy() 13
  - extension 38, 40–44, 51, 54, 495
  - Flatten 702
  - FloorletBlack 476
  - Handle 702
  - Join 535–540
  - modifier 25
  - Read() 715
  - selector 25
  - ThreadStart 645
  - Write 14, 715
  - WriteLine 14
- method of steepest ascent 752
- minimal functionality 132
- minimum 34
- Minimum Tick 372
- minor client 510
- mixed syntax queries 528
- mixin methods 42
- M-matrix theory 254
- modified following business day 170
- modified previous business day 170
- modifier methods 25
- modular programming 22, 67–68
- modules 277
- modulo operators 13
- MOL *see* Method of Lines
- money, time value 160–166
- money market transactions, risks 370–371
- Monitor 667, 678
- monomorphic methods 510
- MonoStrikeBuilder 503
- mono strike caplet volatilities 479–481
- MonoStrikeCapletVolBuilder 476
- MonoStrikeCapletVolBuilderBestFit 477
- monotonicity 335, 340
- Monte Carlo method 510–511, 709, 719–726
- Monte Carlo software engine 497
- multicast delegates 85–86
- multi-core processors 635
- MultiCurveBuilder 441–442, 450–451, 503
- multi-curve building 431–458, 615
- multi-curve framework 435
- multi-dimensional binomial method 233–236
- multi-dimensional data structures 139–140
- multi-method 513
- multiple application domains 305–307
- multiple dispatch 513
- multiple readers 715–719
- multiple strikes 502, 503
- multiple target methods 85
- multiple writers 715–719
- multiplicative binomial method 219
- multiplicative binomical method 219
- multi-processor computers 635
- multiset 535
- MultiStrikeBuilder 503
- multi-strike cap 463, 797–798
- multi-strike floor 464
- multi-tasking 645
- multi-threaded applications 707–732
- multi-threaded data structures 654–659
- multi-threaded timers 682
- multi-threading 437, 635–663, 665–706
- multi-variable optimisation 751–753
- mutex 667, 673–676
- mutual exclusion 665
- mutual-exclusion lock 667
- name collisions 271
- named parameters 121
- name give up 395

- namespaces 271–273
- natural habitat hypothesis 399
- $n$ -dimensional integrals 709
- nested locking 669–672
- .NET
  - application domains 304–309
  - COM interoperability 121–122
  - COM object model 561–562
  - data structures 97
  - dates 101–103
  - delegates 80–87, 495–496
  - disposal 727–728
  - dynamic programming 513–516
  - interfaces 61–67
  - multi-threaded timers 682
  - named parameters 121
  - optional parameters 120–121
  - parallel programming 687–691
  - Regular Expression 299
  - serialisation engines 199–203, 212
  - standard event pattern 87–91
  - unit of deployment 273
- NetDataContractSerializer 201
- net present value (NPV) 402
- .Net Stream Architecture 185
- network streams 187
- Neville's algorithm 351
- 'new' 33
- Newton-Raphson method 317, 463, 484, 752, 755
- non-business date 170
- noncollection 525–526
- non-conservative form 253
- non-destructive peek 114
- non-deterministic 665
- nonlinear least-squares minimisation 751–764
- nonlinear problems 775
- nonlinear programming 751–753
- nonlinear regression 753–754
- non-negativity 335
- non plain vanilla swap 397
- non-polymorphic copy 63
- nonuniform meshes 781
- no overshoot 254
- normal yield curve 338
- notification 495, 676–679
- notional amount 400
- NPV *see* net present value
- $n$ -tuple 155
- null reference 32, 66, 97
- numerator 352
- numerical quadrature formulas 712
- numeric matrices 135–139
- NumericMatrix<T> 131, 137, 168, 289
- numeric type unification 514–516
- numPoints 32
- object-based collections 107–109, 681
- object cloning 62
- Object Connection Architecture (OCA) 26, 82
- object creation 495
- object implementation 210
- object-level binding 74
- object lifecycle 495
- object model 561
- object-oriented programming (OOP) 9, 493, 735–737
- objects 9, 33, 735
  - casting 65–67
  - copying 62–63
  - locking mechanisms 667–673
- observer 495
  - see also* subscriber
- Observer design pattern 86, 87, 496
- Obsolete 285
- OCA *see* Object Connection Architecture
- ODE *see* ordinary differential equation
- off-balance products 373
- off market swap 401
- OIS *see* Overnight Index Swap
- on-balance sheet 372
- OnConnection() 599
- one-factor problems 766–767
- one-sided difference approximations 233
- OneStepFDM 70
- one-step marching scheme 255
- OOP *see* object-oriented programming
- opcode 299, 300
- operator overloading 124, 301–304
- operators 13–14
- Option 17, 18–20, 38, 42, 217–219, 282
- optional parameters 120–121
- option pricer 605–608
- option pricing 33–40, 58–61, 232–233, 709
  - application design 16–21
  - finite difference method 247
- option sensitivities (the Greeks) 33
  - ADE 264
  - analytic formulae 36
  - see also* hedge sensitivities
- order 338
- order class 92
- ordering 525
- ordering collections 546–547
- ordinary differential equation (ODE) 257
- OTC *see* Over-The-Counter
- outer queries 531
- outer variables 77
- out-of-range 101
- overflow 12
- overlapping I/O 708
- Overnight Index Swap (OIS) 401, 403, 432–433
  - discounting 436–437

- overshoot problems 341  
 Over-The-Counter (OTC) 370, 397
- PAC *see* Presentation-Abstractor-Control  
 Pad[e] rational approximations 236  
 PadLeft () 119  
 PadRight () 119  
 pairs trading 709  
 Parallel 687  
 parallel applications 707–732  
 parallel design patterns 707, 709  
 Parallel Framework (PFX) 687  
 Parallel LINQ (PLINQ) 687, 699–701  
 Parallel Pattern Language (PPL) 719  
 parallel programming 635  
   .NET 687–691  
 parallel shift 438  
 parameterless constructor constraint 129  
 parametric polymorphism 509  
 par bond 167  
 parsing 103  
 par swap 464  
 par swap rate 51, 397, 400, 402, 434–436  
 partial class 295  
 partial differential equations (PDEs) 247, 712  
 partition 342  
 par volatility 462  
 par yield curve 167  
 passed by value 10  
 Path 197  
 patterns  
   *Geometric Decomposition* 712  
   *Producer-Consumer* 657–659, 695–698, 719–726  
   *Shared Data* 715  
 paw swap rate 402  
 payer IRS 402  
 payer swaption 51  
 payment date 174  
 payment frequency 311, 401  
 payoff 47, 216  
 PCA *see* Principal Component Analysis  
 PDB *see* pure discount bond  
 PdeIBSPdeBS 248  
 PDS *see* Persistent Data Service  
 perfectly linear 708  
 performance 707  
 Period 408  
 persistency 194  
 Persistent Data Service (PDS) 210  
 Persistent Identifier (PID) 210  
 Persistent Object Manager (POM) 210  
 Persistent Object Service (POS) 210  
 persistent state 210  
 PFX *see* Parallel Framework  
 PIA *see* Primary Interop Assemblies  
 PID *see* Persistent Identifier  
 PiecewiseLinear (PWL) model 474  
 piecewise polynomials 340  
 pipeline 657–659  
 pipe streams 187  
 placeholders 126  
 plain vanilla 159, 397  
 plug-in methods 82–85  
 Point 16, 22, 30, 78, 97, 273, 286  
 poles 352  
 policy classes 501  
 policy-free 85  
 political risk 181  
 polylines 570–571  
 polymorphism 63, 94, 501  
 polynomial interpolation 351  
 POM *see* Persistent Object Manager  
 POS *see* Persistent Object Service  
 positivity 335  
 positivity-preserving rational cubic interpolation  
   335, 338, 344–348  
 postprocessing 497  
 powers 34  
 PPL *see* Parallel Pattern Language  
 pre-confirmation 399  
 preemptive 645  
 preferred habitat theory 337  
 prefetches 635  
 Presentation-Abstractor-Control (PAC) 499  
 present value (PV) 162, 402, 418, 427  
 previous business day 170  
 Price () 39  
 price 317–319  
 price control systems 511  
 pricing bonds 178–180  
 pricing libraries 504  
 Primary Interop Assemblies (PIA) 563  
 principal amount 159  
 Principal Component Analysis (PCA) 337  
 principle of substitutability 63  
 print method 93  
 private assemblies 275  
 private data 26  
 procedural programming model 735, 738  
 Process 636, 637  
 process control system 511  
 processes 636, 641  
 processing 497  
 processor registers 644  
 ProcessStartInfo 637  
*Producer-Consumer* pattern 657–659, 695–698, 719–726  
 producers 654, 719  
 ProgrammableBase 319  
 progressive query construction 532  
 projection 525

- properties 28–30, 93
- PropertyInfo 282
- protocol 53, 64, 80
- Prototype* 63, 84, 494, 508, 744
- proxy 494
- public data 26
- publisher 87, 495
  - see also* broadcaster
- purchasing power risk *see* inflation risk
- pure behaviour 64
- pure discount bond (PDB) 55
- pure risk premium theory 337
- put provision 160
- PV *see* present value
- PWL *see* PiecewiseLinear model
  
- quadratic Bernstein polynomials 338
- quantifiers 526
- quanto swap 430
- quantum 638
- quarterly contract months 372
- query construction 529
- query execution 529
- querying 523
- Queue<T> 115
- quick reference card 524
- quotations 400
- quote perturbation 428
- quotes 389, 394
  
- race conditions 665, 666
- Random 725
- ranges 590
- Rannacher Method* 507
- RAT *see* Resource Allocation and Tracking
- rate calibration 759–763
- rate curves 608–613
- RateSet 407, 408
- rational functions 347, 350
- rational interpolation 352
- raw materials 497
- RCW *see* Runtime Callable Wrapper
- reaction terms 246
- Read() 715
- ReaderWriterLock 715
- reader/writer locks 715–719, 730
- ReadLine 14
- read-only 28
- read-write 28
- real-time data (RTD) server 625–633
- real-time scheduling 709
- receiver 495
- receiver IRS 402
- receiver swaption 51, 464
- rectangular arrays 98–101, 134
- rectangular control mesh 365
- recurism 718–719
- recursive Gaussian quadrature 709
- redemption amount 159, 311, 336
- reduction variable 719
- refactoring 499
- reference element 364
- reference rate 371, 387
- reference-type constraint 129
- reference types 9, 10, 97
- reflection 276–289, 309, 516
- registration 613–614
- reinvestment rate 181
- reinvestment risk 181
- relevant fixing 464
- reliability 41, 196
- reminder programs 683
- Remote Procedure Call (RPC) 201
- remoting 299, 308
- Remove() 119
- Replace() 119
- reset date 400
- Resource Allocation and Tracking (RAT) 508, 511
- resources 273
- responsibility 217
- responsiveness 635
- reusability 61
- reverse convertible bond 182
- rho 51
- Richardson extrapolation 256, 507
- RiskComponent 331
- risk effect 436–437
- risk management 388–389
- roller-coaster IRS 401
- Rolle's theorem 342
- rounding 34
- RPC *see* Remote Procedure Call
- RTD *see* real-time data
- running 645
- Runtime Callable Wrapper (RCW) 561
- run-time performance 132
  
- SABR model 474
- safety 142
- Saul'yev scheme 264
- scalar-valued function 72
- scaleability 715
- scenarios 543–545
- Schedule 178, 311
- schedulers 168
  - exporting 176–177
- SDE *see* stochastic differential equations
- sealed class 46
- secured cash 370
- security 195, 285
- seek 186

- selector methods 25
- self-describing 273
- self-tuning 729
- semaphore 673–676
- semi-discretisation process 777
- sender 495
- sensitivities 421–426, 438–439
  - see also* option sensitivities
- sensitivity analysis 380–383
- separation-of-concerns 43, 58, 719
- serial equivalence 701, 715
- serialisation 185, 199, 214, 285, 322, 665
- serialisation engines 199–203
- Serializable 285
- serial value 170
- server name 631
- servers 26, 68, 82, 509
- set abstraction 112
- set operations 535
- set operators 525
- sets 140–142
- settlement date 387
- settlement sum 387
- shallow copy 62
- Shape 46
- shape parameters 348
- Shared Add-in Wizard 603
- shared assemblies 275
- shared data 715–719
- Shared Data* pattern 715
- shared queue 654
- shifted curve array 438
- shifting curves 439, 702–704
- shift operators 13, 14
- shim 568
- short-end 337, 338
- short term interest rate (STIR) 369–391, 397
  - futures 371–374
  - options 374–377
- side-effects 25
- sign 34
- signalling 676–679
- signature 80
- SIMD *see* Single Instruction Multiple Data
- simple bond class 164–165
- simulations 709
- simultaneous nonlinear equations 754
- simultaneous processing 635
- SingleCurveBuilder 406, 410
- SingleCurveBuilderInterpBestFit 413
- SingleCurveBuilderSmoothingFwd 659
- single-curve building 393–430
- single dispatching 513
- single framework 435
- single inheritance 46
- Single Instruction Multiple Data (SIMD) 691
- Single Responsibility Principle (SRP) 16, 495, 499
- singleton 494
- sleep 645–646
- slices 135
- Smoother 50
- smoothness 335, 341, 415
- software architecture 493–521
- solution vectors 260
- sorted dictionaries 116
- SortedDictionary 116
- SortedList 116
- sparsely spaced 338
- Special Functions 20, 22
- specialisation 742
- speculators 369
- speedup 635, 708
- SpinWait 675
- spot date 400
- spot rate 55
- spot starting IRS 402
- spot volatility 462
- spot zero-coupon rate 167
- spread 400, 401
- spurious oscillations 256
- square root 34
- SRP *see* Single Responsibility Principle
- stability 246
- Stack 114–115
- stack memory 10
- standard collection interfaces 107–109
- standardisation 143–152
- standard query operators 526
- start date 174, 402
- state 495
- statement block 76
- states 506
- static constructors 30, 77–78
- static data 93
- static extension methods 523
- static instability 254
- static methods 15, 20, 30, 50
- static variables 305
- steep yield curve 338
- step up IRS 401
- STIR *see* short term interest rate
- stochastic differential equations (SDE) 55–58, 219, 242, 508, 517, 721
- stopping time 269
- stopwatch 683
- StopWatch 726–727
- strategy algorithm 73
- Strategy* design pattern 215, 219–228, 268, 279, 495, 501
- stream adapters 187, 191–195

- stream architecture 186–187
- stream decorators 189–191
- streams 186
- StrictMapping 552–553
- strike 473
- StringBuilder 117–120
- string name 275
- strings 12, 117–120
- strong name 275
- structs 15, 22
- structural patterns 494, 495, 499–500
- structural relationships 738–743
- subclasses 205–206
- subcontractors 498
- subject 87
- subordinate level 65, 745
- subqueries 531
- subscribers 87, 495
- Substring() 119
- subtype polymorphism 510
- sum-of-squares functions 754
- superordinate level 64, 745
- suppliers 509
- swap in arrears 401
- swap rate 398
- swaps 51, 337, 397
- swaptions 51, 459–491, 464, 798–800
  - multi-curve framework 467–469
- swaption straddle 800–803
- swap valuation 402–403
- symmetric matrices 99
- synchronising collections 681–682
- synchronising constraints 665
- syntax 9
- system level programming 195
- system management 271
- System.Reflection 273, 277, 299
- Task 691–692
- task decomposition 510
- task group 693
- task parallelism 687
- task parallel library (TPL) 691–694
- tear-down method 727
- telescopic 470, 471
- template function 737
- Template Method patterns 259, 262, 495, 501, 507
- templates 568
- template specialisation 298
- tenor 464
- tensors 100
- Tensor<T> 131, 139–140
- termination date 400
- term-sheet 399
- term structure 335
- term structure movements 337
- TextReader 191
- TextWriter 191
- theoretical price 394
- third-degree polynomial 348
- 'this' 28
- thread lifecycle 661
- thread-local storage 725
- thread-local variable 642
- thread pool 680
- ThreadPool 651
- threads 638, 641
  - dependency 676
  - execution barrier 698
  - interrupt 648–650
  - joining 646–647
  - notification 665
  - pooling 651–652, 692
  - priority 650
  - safety 666–667, 688
  - synchronisation 665
- thread-specific variable 642
- ThreadStart 645
- ThreadState 675
- thread states 644–650
- ticks 726
- tick size 383
- tick value 383
- TimeInfo 170
- timers 682–683
- times 101–105
- time sharing 645
- time slice 638
- TimeSpan 101
- time value of money 160–166
- time zones 101–105
- ToLower() 119
- topic class 629–631
- topic parameter 627
- ToString() 103
- Total Returns Swap (TRS) 429
- ToUpper() 119
- TPL *see* task parallel library
- tracking systems 511
- tradable papers 370
- trade date 400
- traditional bootstrapping method 336, 404, 412–413
- traditional kitchen timer 683
- transactional locking mechanisms 717
- transitions 506
- transparency 388
- tridiagonal matrix system 343
- trigonometric functions 34
- TrimEnd() 119
- TrimStart() 119

- trinomial method 241–246
  - alternative probabilities 268
- try-throw-catch 50
- tuples 155–156, 157–158
- two-dimensional shape data 15
- two-factor payoff hierarchies 46–50
- two-sided difference approximations 233
- Type 278
- type libraries 582
- types 304
- type safety 126
- UDF *see* User Defined Function; user-defined functions
- UML *see* Unified Modeling Language
- UML sequence diagram 326
- unary association 741–742
- unblocked 646
- Uncertain Volatility Models (UVM) 263
- undeterministic 666
- Unicode characters 12
- Unified Modeling Language (UML) 738
- unit of deployment 273
- unsecured cash 370
- unsecured deposit 395
- up-front 401
- upgradeable locks 717, 718–719
- upper triangular matrices 98
- upwinding 246, 254
- usability 142
- use cases 195
- user-defined aggregation methods 533, 534–535
- user-defined data structures 125–158
- user-defined functions (UDF) 519
- User Defined Function (UDF) 566, 583, 592
- user-defined structs 15–16
- user input, validation 13
- user interface 635
- user settings 15
- utility code 597–599
- UVM *see* Uncertain Volatility Models
- valuation methods 709
- value-type constraint 129
- value types 9, 10, 97
- var 80, 524
- varargs method 299
- variance swap 429
- variants 501
- variations 495
- Vasicek model 54, 57
- vectors 135–139
- Vector<T> 131, 135–136, 289, 357
- vega 51
- versioning 590
- version tolerance 199
- vertical slabs 712
- Visitor design pattern 42, 50, 53, 58, 357, 495, 501
- Volatile() 591
- volatile 641, 644
- volatile methods 590–591
- volatility 55, 473
- volatility bootstrapping 483
- volatility calculation 502
- volatility matrix 503
- volatility optimisation 487–490
- volatility risk 181
- volatility surface 321, 731
- volatility swap 429
- volatility table 350
- vomma 51
- von Neumann stability analysis 253
- VSTO 568–569
- Whole-Part pattern 499, 745–748
- Windows Forms 600–601
- wrapping queries 532
- Write 14, 715
- WriteLine 14
- write-only 28
- XLA 567
- XLBond 319, 321, 324
- XLL 567
- XML data 523
- XML formatter 201
- XML serialisation 204–209
- XmlSerializer 204
- year fraction 169
- year inflation swap 430
- yield 107, 166–167, 317–319
- yield curve 337
- yield return statement 107
- yield to maturity (YTM) 167
- zero-coupon bonds (ZCB) 23, 159, 336
  - see also* discount bonds
- zero coupon yield curve 336

<http://www.pbookshop.com>