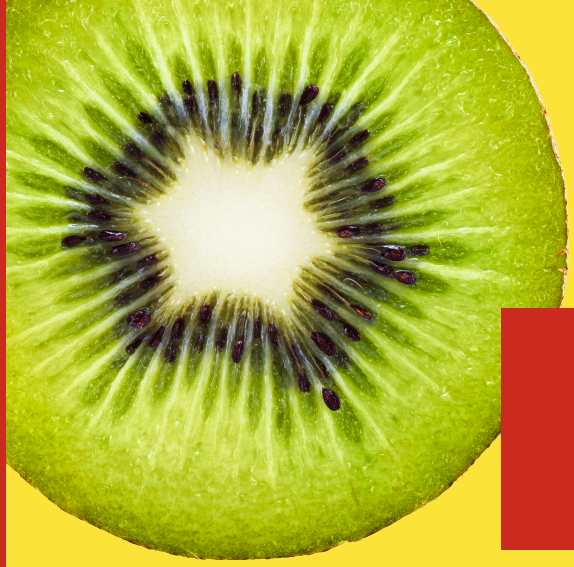

walter DOBERENZ
thomas GEWINNUS

Visual Basic 2015

GRUNDLAGEN
PROFIWISSEN
REZEPTE

// Visual Basic-Grundlagen
// LINQ, OOP, ADO.NET
// App-Entwicklung
// Über 150 Praxisbeispiele

EXTRA: 700 Seiten Bonuskapitel zu WPF und Windows Forms

HANSER

Doberenz/Gewinnus

Visual Basic 2015
Grundlagen, Profiwissen und Rezepte

Walter Doberenz
Thomas Gewinnus

Visual Basic 2015

Grundlagen, Profiwissen und Rezepte

HANSER

Die Autoren:

Professor Dr.-Ing. habil. Walter Doberenz, Wintersdorf
Dipl.-Ing. Thomas Gewinnus, Frankfurt/Oder

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <*http://dnb.d-nb.de*> abrufbar.

Lektorat: Sieglinde Schärl
Herstellung: Irene Weilhart
Satz: Ingenieurbüro Gewinnus
Sprachlektorat: Walter Doberenz
Umschlagdesign: Marc Müller-Bremer, *www.rebranding.de,* München
Umschlagrealisation: Stephan Rönigk
Druck und Bindung: Kösel, Krugzell
Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702
Printed in Germany

Print-ISBN: 978-3-446-44380-8
E-Book-ISBN: 978-3-446-44605-2

Inhaltsverzeichnis

Vorwort **45**

Teil I: Grundlagen

1 Einstieg in Visual Studio 2015 ..... **51**

1.1 Die Installation von Visual Studio 2015 51
1.1.1 Überblick über die Produktpalette 51
1.1.2 Anforderungen an Hard- und Software 52
1.2 Unser allererstes VB-Programm 53
1.2.1 Vorbereitungen 53
1.2.2 Programm schreiben 55
1.2.3 Programm kompilieren und testen 55
1.2.4 Einige Erläuterungen zum Quellcode 56
1.2.5 Konsolenanwendungen sind out 57
1.3 Die Windows-Philosophie 57
1.3.1 Mensch-Rechner-Dialog 58
1.3.2 Objekt- und ereignisorientierte Programmierung 58
1.3.3 Windows-Programmierung unter Visual Studio 2015 59
1.4 Die Entwicklungsumgebung Visual Studio 2015 61
1.4.1 Neues Projekt 61
1.4.2 Die wichtigsten Fenster 62
1.5 Microsofts .NET-Technologie 64
1.5.1 Zur Geschichte von .NET 64
1.5.2 .NET-Features und Begriffe 66
1.6 Wichtige Neuigkeiten in Visual Studio 2015 74
1.6.1 Entwicklungsumgebung 74
1.6.2 Neue VB-Sprachfeatures 74
1.6.3 Code-Editor 74
1.6.4 NET Framework 4.6 75

1.7 Praxisbeispiele ... 75
1.7.1 Windows-Anwendung für Einsteiger ... 75
1.7.2 Windows-Anwendung für fortgeschrittene Einsteiger ... 79

2 Einführung in Visual Basic ... 87

2.1 Grundbegriffe ... 87
2.1.1 Anweisungen ... 87
2.1.2 Bezeichner ... 88
2.1.3 Kommentare ... 89
2.1.4 Zeilenumbruch ... 90
2.2 Datentypen, Variablen und Konstanten ... 92
2.2.1 Fundamentale Typen ... 92
2.2.2 Wertetypen versus Verweistypen ... 93
2.2.3 Benennung von Variablen ... 93
2.2.4 Deklaration von Variablen ... 94
2.2.5 Typinferenz ... 97
2.2.6 Konstanten deklarieren ... 97
2.2.7 Gültigkeitsbereiche von Deklarationen ... 98
2.2.8 Lokale Variablen mit Dim ... 98
2.2.9 Lokale Variablen mit Static ... 99
2.2.10 Private globale Variablen ... 99
2.2.11 Public Variablen ... 100
2.3 Wichtige Datentypen im Überblick ... 100
2.3.1 Byte, Short, Integer, Long ... 100
2.3.2 Single, Double und Decimal ... 101
2.3.3 Char und String ... 101
2.3.4 Date ... 102
2.3.5 Boolean ... 103
2.3.6 Object ... 103
2.4 Konvertieren von Datentypen ... 104
2.4.1 Implizite und explizite Konvertierung ... 104
2.4.2 Welcher Datentyp passt zu welchem? ... 105
2.4.3 Konvertierungsfunktionen ... 106
2.4.4 CType-Funktion ... 107
2.4.5 Konvertieren von Strings ... 107
2.4.6 Die Convert-Klasse ... 109
2.4.7 Die Parse-Methode ... 109
2.4.8 Boxing und Unboxing ... 110

2.4.9 TryCast-Operator ... 111
2.4.10 Nullable Types ... 111
2.5 Operatoren ... 112
2.5.1 Arithmetische Operatoren ... 112
2.5.2 Zuweisungsoperatoren ... 113
2.5.3 Logische Operatoren ... 114
2.5.4 Vergleichsoperatoren ... 115
2.5.5 Rangfolge der Operatoren ... 115
2.6 Kontrollstrukturen ... 116
2.6.1 Verzweigungsbefehle ... 116
2.6.2 Schleifenanweisungen ... 119
2.7 Benutzerdefinierte Datentypen ... 120
2.7.1 Enumerationen ... 120
2.7.2 Strukturen ... 121
2.8 Nutzerdefinierte Funktionen/Prozeduren ... 124
2.8.1 Deklaration und Syntax ... 124
2.8.2 Parameterübergabe allgemein ... 126
2.8.3 Übergabe mit ByVal und ByRef ... 127
2.8.4 Optionale Parameter ... 128
2.8.5 Überladene Funktionen/Prozeduren ... 128
2.9 Praxisbeispiele ... 129
2.9.1 Vom PAP zum Konsolen-Programm ... 129
2.9.2 Vom Konsolen- zum Windows-Programm ... 131
2.9.3 Schleifenanweisungen kennen lernen ... 133
2.9.4 Methoden überladen ... 136
2.9.5 Eine Iterationsschleife verstehen ... 138
2.9.6 Anwendungen von C# nach Visual Basic portieren ... 141

3 OOP-Konzepte ... 149

3.1 Strukturierter versus objektorientierter Entwurf ... 149
3.1.1 Was bedeutet strukturierte Programmierung? ... 149
3.1.2 Was heißt objektorientierte Programmierung? ... 150
3.2 Grundbegriffe der OOP ... 151
3.2.1 Objekt, Klasse, Instanz ... 151
3.2.2 Kapselung und Wiederverwendbarkeit ... 152
3.2.3 Vererbung und Polymorphie ... 152
3.2.4 Sichtbarkeit von Klassen und ihren Mitgliedern ... 153
3.2.5 Allgemeiner Aufbau einer Klasse ... 154

3.3 Ein Objekt erzeugen ... 155
3.3.1 Referenzieren und Instanziieren ... 156
3.3.2 Klassische Initialisierung ... 157
3.3.3 Objekt-Initialisierer ... 157
3.3.4 Arbeiten mit dem Objekt ... 157
3.3.5 Zerstören des Objekts ... 158
3.4 OOP-Einführungsbeispiel ... 158
3.4.1 Vorbereitungen ... 158
3.4.2 Klasse definieren ... 159
3.4.3 Objekt erzeugen und initialisieren ... 160
3.4.4 Objekt verwenden ... 160
3.4.5 Unterstützung durch die IntelliSense ... 160
3.4.6 Objekt testen ... 161
3.4.7 Warum unsere Klasse noch nicht optimal ist ... 162
3.5 Eigenschaften ... 162
3.5.1 Eigenschaften kapseln ... 162
3.5.2 Eigenschaften mit Zugriffsmethoden kapseln ... 165
3.5.3 Lese-/Schreibschutz für Eigenschaften ... 166
3.5.4 Statische Eigenschaften ... 167
3.5.5 Selbst implementierende Eigenschaften ... 168
3.6 Methoden ... 169
3.6.1 Öffentliche und private Methoden ... 169
3.6.2 Überladene Methoden ... 170
3.6.3 Statische Methoden ... 171
3.7 Ereignisse ... 172
3.7.1 Ereignise deklarieren ... 172
3.7.2 Ereignis auslösen ... 173
3.7.3 Ereignis auswerten ... 173
3.7.4 Benutzerdefinierte Ereignisse (Custom Events) ... 175
3.8 Arbeiten mit Konstruktor und Destruktor ... 178
3.8.1 Der Konstruktor erzeugt das Objekt ... 178
3.8.2 Bequemer geht's mit einem Objekt-Initialisierer ... 180
3.8.3 Destruktor und Garbage Collector räumen auf ... 181
3.8.4 Mit Using den Lebenszyklus des Objekts kapseln ... 184
3.9 Vererbung und Polymorphie ... 184
3.9.1 Vererbungsbeziehungen im Klassendiagramm ... 184
3.9.2 Überschreiben von Methoden (Method-Overriding) ... 186
3.9.3 Klassen implementieren ... 186

3.9.4 Objekte implementieren ... 191
3.9.5 Ausblenden von Mitgliedern durch Vererbung ... 192
3.9.6 Allgemeine Hinweise und Regeln zur Vererbung ... 194
3.9.7 Polymorphe Methoden ... 195
3.10 Besondere Klassen und Features ... 197
3.10.1 Abstrakte Klassen ... 197
3.10.2 Abstrakte Methoden ... 198
3.10.3 Versiegelte Klassen ... 198
3.10.4 Partielle Klassen ... 199
3.10.5 Die Basisklasse System.Object ... 201
3.10.6 Property-Accessors ... 202
3.10.7 Nullbedingter Operator ... 202
3.11 Schnittstellen (Interfaces) ... 203
3.11.1 Definition einer Schnittstelle ... 203
3.11.2 Implementieren einer Schnittstelle ... 204
3.11.3 Abfragen, ob eine Schnittstelle vorhanden ist ... 205
3.11.4 Mehrere Schnittstellen implementieren ... 205
3.11.5 Schnittstellenprogrammierung ist ein weites Feld ... 205
3.12 Praxisbeispiele ... 206
3.12.1 Eigenschaften sinnvoll kapseln ... 206
3.12.2 Eine statische Klasse anwenden ... 209
3.12.3 Vom fetten zum dünnen Client ... 211
3.12.4 Schnittstellenvererbung verstehen ... 220
3.12.5 Aggregation und Vererbung gegenüberstellen ... 224
3.12.6 Eine Klasse zur Matrizenrechnung entwickeln ... 230
3.12.7 Rechner für komplexe Zahlen ... 236
3.12.8 Formel-Rechner mit dem CodeDOM ... 244
3.12.9 Einen Funktionsverlauf grafisch darstellen ... 249
3.12.10 Sortieren mit IComparable/IComparer ... 253
3.12.11 Objektbäume in generischen Listen abspeichern ... 258
3.12.12 OOP beim Kartenspiel erlernen ... 264

4 Arrays, Strings, Funktionen ... 269

4.1 Datenfelder (Arrays) ... 269
4.1.1 Ein Array deklarieren ... 269
4.1.2 Zugriff auf Array-Elemente ... 270
4.1.3 Oberen Index ermitteln ... 270
4.1.4 Explizite Arraygrenzen ... 270

4.1.5 Arrays erzeugen und initialisieren ... 270
4.1.6 Zugriff mittels Schleife ... 271
4.1.7 Mehrdimensionale Arrays ... 272
4.1.8 Dynamische Arrays ... 273
4.1.9 Zuweisen von Arrays ... 274
4.1.10 Arrays aus Strukturvariablen ... 275
4.1.11 Löschen von Arrays ... 276
4.1.12 Eigenschaften und Methoden von Arrays ... 276
4.1.13 Übergabe von Arrays ... 279
4.2 Zeichenkettenverarbeitung ... 280
4.2.1 Strings zuweisen ... 280
4.2.2 Eigenschaften und Methoden eines Strings ... 280
4.2.3 Kopieren eines Strings in ein Char-Array ... 283
4.2.4 Wichtige (statische) Methoden der String-Klasse ... 283
4.2.5 Die StringBuilder-Klasse ... 285
4.3 Reguläre Ausdrücke ... 288
4.3.1 Wozu braucht man reguläre Ausdrücke? ... 288
4.3.2 Eine kleine Einführung ... 289
4.3.3 Wichtige Methoden der Klasse Regex ... 289
4.3.4 Kompilierte reguläre Ausdrücke ... 291
4.3.5 RegexOptions-Enumeration ... 292
4.3.6 Metazeichen (Escape-Zeichen) ... 293
4.3.7 Zeichenmengen (Character Sets) ... 294
4.3.8 Quantifizierer ... 295
4.3.9 Zero-Width Assertions ... 296
4.3.10 Gruppen ... 300
4.3.11 Text ersetzen ... 300
4.3.12 Text splitten ... 301
4.4 Datums- und Zeitberechnungen ... 302
4.4.1 Grundlegendes ... 302
4.4.2 Wichtige Eigenschaften von DateTime-Variablen ... 303
4.4.3 Wichtige Methoden von DateTime-Variablen ... 304
4.4.4 Wichtige Mitglieder der DateTime-Struktur ... 305
4.4.5 Konvertieren von Datumstrings in DateTime-Werte ... 306
4.4.6 Die TimeSpan-Struktur ... 306
4.5 Vordefinierten Funktionen ... 308
4.5.1 Mathematik ... 308
4.5.2 Datums- und Zeitfunktionen ... 310

4.6 Zahlen formatieren ... 312
4.6.1 Die ToString-Methode ... 313
4.6.2 Die Format-Methode ... 314
4.6.3 Stringinterpolation ... 316
4.7 Praxisbeispiele ... 316
4.7.1 Zeichenketten verarbeiten ... 316
4.7.2 Zeichenketten mittels StringBuilder addieren ... 319
4.7.3 Reguläre Ausdrücke testen ... 323
4.7.4 Fehler bei mathematischen Operationen behandeln ... 325
4.7.5 Methodenaufrufe mit Array-Parametern ... 328
4.7.6 String in Array kopieren und umgekehrt ... 331
4.7.7 Ein Byte-Array in einen String konvertieren ... 333
4.7.8 Strukturvariablen in Arrays einsetzen ... 335

5 Weitere Sprachfeatures ... 339

5.1 Namespaces (Namensräume) ... 339
5.1.1 Ein kleiner Überblick ... 339
5.1.2 Die Imports-Anweisung ... 341
5.1.3 Namespace-Alias ... 341
5.1.4 Namespaces in Projekteigenschaften ... 342
5.1.5 Namespace Alias Qualifizierer ... 343
5.1.6 Eigene Namespaces einrichten ... 343
5.2 Überladen von Operatoren ... 344
5.2.1 Syntaxregeln ... 344
5.2.2 Praktische Anwendung ... 345
5.2.3 Konvertierungsoperatoren überladen ... 346
5.3 Auflistungen (Collections) ... 347
5.3.1 Beziehungen zwischen den Schnittstellen ... 347
5.3.2 IEnumerable ... 348
5.3.3 ICollection ... 349
5.3.4 IList ... 349
5.3.5 Iteratoren ... 349
5.3.6 Die ArrayList-Collection ... 350
5.3.7 Die Hashtable ... 351
5.4 Generische Datentypen ... 352
5.4.1 Wie es früher einmal war ... 352
5.4.2 Typsicherheit durch Generics ... 354
5.4.3 List-Collection ersetzt ArrayList ... 355

5.4.4 Über die Vorzüge generischer Collections ... 356
5.4.5 Typbeschränkungen durch Constraints ... 357
5.4.6 Collection-Initialisierer ... 358
5.4.7 Generische Methoden ... 358
5.5 Delegates ... 359
5.5.1 Delegates sind Methodenzeiger ... 359
5.5.2 Delegate-Typ definieren ... 360
5.5.3 Delegate-Objekt erzeugen ... 361
5.5.4 Delegates vereinfacht instanziieren ... 361
5.5.5 Relaxed Delegates ... 362
5.5.6 Anonyme Methoden ... 362
5.5.7 Lambda-Ausdrücke ... 363
5.5.8 Lambda-Ausdrücke in der Task Parallel Library ... 364
5.6 Dynamische Programmierung ... 366
5.6.1 Wozu dynamische Programmierung? ... 366
5.6.2 Das Prinzip der dynamischen Programmierung ... 366
5.6.3 Kovarianz und Kontravarianz ... 370
5.7 Weitere Datentypen ... 371
5.7.1 BigInteger ... 371
5.7.2 Complex ... 373
5.7.3 Tuple(Of T) ... 374
5.7.4 SortedSet(Of T) ... 374
5.8 Praxisbeispiele ... 376
5.8.1 ArrayList versus generische List ... 376
5.8.2 Delegates und Lambda Expressions ... 379
5.8.3 Mit dynamischem Objekt eine Datei durchsuchen ... 382

6 Einführung in LINQ ... 387

6.1 LINQ-Grundlagen ... 387
6.1.1 Die LINQ-Architektur ... 387
6.1.2 LINQ-Implementierungen ... 388
6.1.3 Anonyme Typen ... 388
6.1.4 Erweiterungsmethoden ... 390
6.2 Abfragen mit LINQ to Objects ... 391
6.2.1 Grundlegendes zur LINQ-Syntax ... 391
6.2.2 Zwei alternative Schreibweisen von LINQ-Abfragen ... 392
6.2.3 Übersicht der wichtigsten Abfrage-Operatoren ... 394

6.3 LINQ-Abfragen im Detail ... 395
6.3.1 Die Projektionsoperatoren Select und SelectMany ... 396
6.3.2 Der Restriktionsoperator Where ... 398
6.3.3 Die Sortierungsoperatoren OrderBy und ThenBy ... 398
6.3.4 Der Gruppierungsoperator GroupBy ... 400
6.3.5 Verknüpfen mit Join ... 401
6.3.6 Aggregat-Operatoren ... 402
6.3.7 Verzögertes Ausführen von LINQ-Abfragen ... 404
6.3.8 Konvertierungsmethoden ... 405
6.3.9 Abfragen mit PLINQ ... 405
6.4 Praxisbeispiele ... 408
6.4.1 Die Syntax von LINQ-Abfragen verstehen ... 408
6.4.2 Aggregat-Abfragen mit LINQ ... 411
6.4.3 LINQ im Schnelldurchgang erlernen ... 413
6.4.4 Strings mit LINQ abfragen und filtern ... 416
6.4.5 Duplikate aus einer Liste oder einem Array entfernen ... 417
6.4.6 Arrays mit LINQ initialisieren ... 420
6.4.7 Arrays per LINQ mit Zufallszahlen füllen ... 422
6.4.8 Einen String mit Wiederholmuster erzeugen ... 423
6.4.9 Mit LINQ Zahlen und Strings sortieren ... 425
6.4.10 Mit LINQ Collections von Objekten sortieren ... 426
6.4.11 Ergebnisse von LINQ-Abfragen in ein Array kopieren ... 428

Teil II: Technologien

7 Zugriff auf das Dateisystem ... 431
7.1 Grundlagen ... 431
7.1.1 Klassen für Verzeichnis- und Dateioperationen ... 432
7.1.2 Statische versus Instanzen-Klasse ... 432
7.2 Übersichten ... 433
7.2.1 Methoden der Directory-Klasse ... 433
7.2.2 Methoden eines DirectoryInfo-Objekts ... 434
7.2.3 Eigenschaften eines DirectoryInfo-Objekts ... 434
7.2.4 Methoden der File-Klasse ... 434
7.2.5 Methoden eines FileInfo-Objekts ... 435
7.2.6 Eigenschaften eines FileInfo-Objekts ... 436

7.3 Operationen auf Verzeichnisebene ... 436
7.3.1 Existenz eines Verzeichnisses/einer Datei feststellen ... 436
7.3.2 Verzeichnisse erzeugen und löschen ... 437
7.3.3 Verzeichnisse verschieben und umbenennen ... 438
7.3.4 Aktuelles Verzeichnis bestimmen ... 438
7.3.5 Unterverzeichnisse ermitteln ... 438
7.3.6 Alle Laufwerke ermitteln ... 439
7.3.7 Dateien kopieren und verschieben ... 440
7.3.8 Dateien umbenennen ... 441
7.3.9 Dateiattribute feststellen ... 441
7.3.10 Verzeichnis einer Datei ermitteln ... 443
7.3.11 Alle im Verzeichnis enthaltene Dateien ermitteln ... 443
7.3.12 Dateien und Unterverzeichnisse ermitteln ... 443
7.4 Zugriffsberechtigungen ... 444
7.4.1 ACL und ACE ... 444
7.4.2 SetAccessControl-Methode ... 445
7.4.3 Zugriffsrechte anzeigen ... 445
7.5 Weitere wichtige Klassen ... 446
7.5.1 Die Path-Klasse ... 446
7.5.2 Die Klasse FileSystemWatcher ... 447
7.6 Datei- und Verzeichnisdialoge ... 449
7.6.1 OpenFileDialog und SaveFileDialog ... 449
7.6.2 FolderBrowserDialog ... 451
7.7 Praxisbeispiele ... 452
7.7.1 Infos über Verzeichnisse und Dateien gewinnen ... 452
7.7.2 Die Verzeichnisstruktur in eine TreeView einlesen ... 455
7.7.3 Mit LINQ und RegEx Verzeichnisbäume durchsuchen ... 457

8 Dateien lesen und schreiben ... 463

8.1 Grundprinzip der Datenpersistenz ... 463
8.1.1 Dateien und Streams ... 463
8.1.2 Die wichtigsten Klassen ... 464
8.1.3 Erzeugen eines Streams ... 465
8.2 Dateiparameter ... 465
8.2.1 FileAccess ... 465
8.2.2 FileMode ... 465
8.2.3 FileShare ... 466