



walter DOBERENZ  
thomas GEWINNUS

# Visual C# 2015

**GRUNDLAGEN**  
**PROFIWISSEN**  
**REZEPTE**

// C#-Grundlagen  
// LINQ, OOP, ADO.NET  
// App-Entwicklung  
// Über 150 Praxisbeispiele



**EXTRA: 700 Seiten Bonuskapitel**  
zu WPF und Windows Forms

HANSER

Doberenz/Gewinnus

## Visual C# 2015 Grundlagen, Profiwissen und Rezepte

### Bleiben Sie auf dem Laufenden!



Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter



[www.hanser-fachbuch.de/newsletter](http://www.hanser-fachbuch.de/newsletter)



**Hanser Update** ist der IT-Blog des Hanser Verlags mit Beiträgen und Praxistipps von unseren Autoren rund um die Themen Online Marketing, Webentwicklung, Programmierung, Softwareentwicklung sowie IT- und Projektmanagement. Lesen Sie mit und abonnieren Sie unsere News unter



[www.hanser-fachbuch.de/update](http://www.hanser-fachbuch.de/update)





Walter Doberenz  
Thomas Gewinnus

# **Visual C# 2015**

Grundlagen, Profiwissen  
und Rezepte

HANSER

*Die Autoren:*

*Professor Dr.-Ing. habil. Walter Doberenz, Wintersdorf*

*Dipl.-Ing. Thomas Gewinnus, Frankfurt/Oder*

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autoren und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso übernehmen Autoren und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.



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.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdrucks und der Vervielfältigung des Buches, oder Teilen daraus, sind vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2015 Carl Hanser Verlag München

<http://www.hanser-fachbuch.de>

Lektorat: Sieglinde Schärl

Herstellung: Irene Weilhart

Satz: Ingenieurbüro Gewinnus

Sprachlektorat: Walter Doberenz

Umschlagdesign: Marc Müller-Bremer, [www.rebranding.de](http://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: 978-3-446-44381-5

E-Book-ISBN: 978-3-446-44606-9

# Inhaltsverzeichnis

<b>Vorwort</b> .....	<b>45</b>
 <b>Teil I: Grundlagen</b>	
<b>1 Einstieg in Visual Studio 2015</b> .....	<b>51</b>
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 C#-Programm .....	53
1.2.1 Vorbereitungen .....	53
1.2.2 Quellcode 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 .....	57
1.3.2 Objekt- und ereignisorientierte Programmierung .....	58
1.3.3 Programmieren mit Visual Studio 2015 .....	59
1.4 Die Entwicklungsumgebung Visual Studio 2015 .....	60
1.4.1 Neues Projekt .....	61
1.4.2 Die wichtigsten Fenster .....	62
1.5 Microsofts .NET-Technologie .....	65
1.5.1 Zur Geschichte von .NET .....	65
1.5.2 .NET-Features und Begriffe .....	67
1.6 Wichtige Neuigkeiten in Visual Studio 2015 .....	74
1.6.1 Entwicklungsumgebung .....	74
1.6.2 Neue C#-Sprachfeatures .....	74
1.6.3 Code-Editor .....	75
1.6.4 NET Framework 4.6 .....	75

1.7	Praxisbeispiele .....	76
1.7.1	Unsere erste Windows Forms-Anwendung .....	76
1.7.2	Umrechnung Euro-Dollar .....	80
<b>2</b>	<b>Grundlagen der Sprache C# .....</b>	<b>89</b>
2.1	Grundbegriffe .....	89
2.1.1	Anweisungen .....	89
2.1.2	Bezeichner .....	90
2.1.3	Schlüsselwörter .....	91
2.1.4	Kommentare .....	91
2.2	Datentypen, Variablen und Konstanten .....	92
2.2.1	Fundamentale Typen .....	92
2.2.2	Werttypen versus Verweistypen .....	93
2.2.3	Benennung von Variablen .....	94
2.2.4	Deklaration von Variablen .....	94
2.2.5	Typsuffixe .....	96
2.2.6	Zeichen und Zeichenketten .....	97
2.2.7	object-Datentyp .....	99
2.2.8	Konstanten deklarieren .....	99
2.2.9	Nullable Types .....	100
2.2.10	Typinferenz .....	101
2.2.11	Gültigkeitsbereiche und Sichtbarkeit .....	102
2.3	Konvertieren von Datentypen .....	102
2.3.1	Implizite und explizite Konvertierung .....	102
2.3.2	Welcher Datentyp passt zu welchem? .....	104
2.3.3	Konvertieren von string .....	105
2.3.4	Die Convert-Klasse .....	107
2.3.5	Die Parse-Methode .....	107
2.3.6	Boxing und Unboxing .....	108
2.4	Operatoren .....	109
2.4.1	Arithmetische Operatoren .....	110
2.4.2	Zuweisungsoperatoren .....	111
2.4.3	Logische Operatoren .....	112
2.4.4	Rangfolge der Operatoren .....	115
2.5	Kontrollstrukturen .....	116
2.5.1	Verzweigungsbefehle .....	116
2.5.2	Schleifenanweisungen .....	119

2.6	Benutzerdefinierte Datentypen .....	122
2.6.1	Enumerationen .....	122
2.6.2	Strukturen .....	123
2.7	Nutzerdefinierte Methoden .....	125
2.7.1	Methoden mit Rückgabewert .....	126
2.7.2	Methoden ohne Rückgabewert .....	127
2.7.3	Parameterübergabe mit ref .....	128
2.7.4	Parameterübergabe mit out .....	129
2.7.5	Methodenüberladung .....	130
2.7.6	Optionale Parameter .....	131
2.7.7	Benannte Parameter .....	132
2.8	Praxisbeispiele .....	133
2.8.1	Vom PAP zur Konsolenanwendung .....	133
2.8.2	Ein Konsolen- in ein Windows-Programm verwandeln .....	135
2.8.3	Schleifenanweisungen verstehen .....	137
2.8.4	Benutzerdefinierte Methoden überladen .....	139
2.8.5	Anwendungen von Visual Basic nach C# portieren .....	142
<b>3</b>	<b>OOP-Konzepte .....</b>	<b>149</b>
3.1	Kleine Einführung in die OOP .....	149
3.1.1	Historische Entwicklung .....	149
3.1.2	Grundbegriffe der OOP .....	151
3.1.3	Sichtbarkeit von Klassen und ihren Mitgliedern .....	153
3.1.4	Allgemeiner Aufbau einer Klasse .....	154
3.1.5	Das Erzeugen eines Objekts .....	155
3.1.6	Einführungsbeispiel .....	158
3.2	Eigenschaften .....	163
3.2.1	Eigenschaften mit Zugriffsmethoden kapseln .....	163
3.2.2	Berechnete Eigenschaften .....	165
3.2.3	Lese-/Schreibschutz .....	167
3.2.4	Property-Accessoren .....	168
3.2.5	Statische Felder/Eigenschaften .....	168
3.2.6	Einfache Eigenschaften automatisch implementieren .....	171
3.3	Methoden .....	172
3.3.1	Öffentliche und private Methoden .....	172
3.3.2	Überladene Methoden .....	173
3.3.3	Statische Methoden .....	174

3.4	Ereignisse .....	176
3.4.1	Ereignis hinzufügen .....	176
3.4.2	Ereignis verwenden .....	179
3.5	Arbeiten mit Konstruktor und Destruktor .....	182
3.5.1	Konstruktor und Objektinitialisierer .....	182
3.5.2	Destruktor und Garbage Collector .....	185
3.5.3	Mit using den Lebenszyklus des Objekts kapseln .....	188
3.5.4	Verzögerte Initialisierung .....	188
3.6	Vererbung und Polymorphie .....	189
3.6.1	Klassendiagramm .....	189
3.6.2	Method-Overriding .....	190
3.6.3	Klassen implementieren .....	191
3.6.4	Implementieren der Objekte .....	194
3.6.5	Ausblenden von Mitgliedern durch Vererbung .....	196
3.6.6	Allgemeine Hinweise und Regeln zur Vererbung .....	197
3.6.7	Polymorphes Verhalten .....	199
3.6.8	Die Rolle von System.Object .....	201
3.7	Spezielle Klassen .....	202
3.7.1	Abstrakte Klassen .....	202
3.7.2	Versiegelte Klassen .....	204
3.7.3	Partielle Klassen .....	204
3.7.4	Statische Klassen .....	205
3.8	Schnittstellen (Interfaces) .....	206
3.8.1	Definition einer Schnittstelle .....	207
3.8.2	Implementieren einer Schnittstelle .....	207
3.8.3	Abfragen, ob Schnittstelle vorhanden ist .....	208
3.8.4	Mehrere Schnittstellen implementieren .....	209
3.8.5	Schnittstellenprogrammierung ist ein weites Feld .....	209
3.9	Praxisbeispiele .....	209
3.9.1	Eigenschaften sinnvoll kapseln .....	209
3.9.2	Eine statische Klasse anwenden .....	212
3.9.3	Vom fetten zum schlanken Client .....	214
3.9.4	Schnittstellenvererbung verstehen .....	226
3.9.5	Rechner für komplexe Zahlen .....	231
3.9.6	Formel-Rechner mit dem CodeDOM .....	240
3.9.7	Berechnungsergebnisse als Diagramm darstellen .....	248
3.9.8	Sortieren mit IComparable/IComparer .....	252
3.9.9	Einen Objektbaum in generischen Listen abspeichern .....	257

3.9.10	OOO beim Kartenspiel erlernen .....	263
3.9.11	Eine Klasse zur Matrizenrechnung entwickeln .....	267
<b>4</b>	<b>Arrays, Strings, Funktionen .....</b>	<b>273</b>
4.1	Datenfelder (Arrays) .....	273
4.1.1	Array deklarieren .....	273
4.1.2	Array instanziiieren .....	274
4.1.3	Array initialisieren .....	274
4.1.4	Zugriff auf Array-Elemente .....	275
4.1.5	Zugriff mittels Schleife .....	276
4.1.6	Mehrdimensionale Arrays .....	277
4.1.7	Zuweisen von Arrays .....	279
4.1.8	Arrays aus Strukturvariablen .....	280
4.1.9	Löschen und Umdimensionieren von Arrays .....	281
4.1.10	Eigenschaften und Methoden von Arrays .....	282
4.1.11	Übergabe von Arrays .....	284
4.2	Verarbeiten von Zeichenketten .....	285
4.2.1	Zuweisen von Strings .....	285
4.2.2	Eigenschaften und Methoden von String-Variablen .....	286
4.2.3	Wichtige Methoden der String-Klasse .....	288
4.2.4	Die StringBuilder-Klasse .....	290
4.3	Reguläre Ausdrücke .....	292
4.3.1	Wozu werden reguläre Ausdrücke verwendet? .....	293
4.3.2	Eine kleine Einführung .....	293
4.3.3	Wichtige Methoden/Eigenschaften der Klasse Regex .....	294
4.3.4	Kompilierte reguläre Ausdrücke .....	296
4.3.5	RegexOptions-Enumeration .....	297
4.3.6	Metazeichen (Escape-Zeichen) .....	297
4.3.7	Zeichenmengen (Character Sets) .....	298
4.3.8	Quantifizierer .....	300
4.3.9	Zero-Width Assertions .....	301
4.3.10	Gruppen .....	304
4.3.11	Text ersetzen .....	305
4.3.12	Text splitten .....	306
4.4	Datums- und Zeitberechnungen .....	307
4.4.1	Die DateTime-Struktur .....	307
4.4.2	Wichtige Eigenschaften von DateTime-Variablen .....	308
4.4.3	Wichtige Methoden von DateTime-Variablen .....	309

4.4.4	Wichtige Mitglieder der DateTime-Struktur .....	309
4.4.5	Konvertieren von Datumstrings in DateTime-Werte .....	310
4.4.6	Die TimeSpan-Struktur .....	311
4.5	Mathematische Funktionen .....	312
4.5.1	Überblick .....	312
4.5.2	Zahlen runden .....	313
4.5.3	Winkel umrechnen .....	313
4.5.4	Potenz- und Wurzeloperationen .....	313
4.5.5	Logarithmus und Exponentialfunktionen .....	314
4.5.6	Zufallszahlen erzeugen .....	314
4.6	Zahlen- und Datumsformatierungen .....	315
4.6.1	Anwenden der ToString-Methode .....	315
4.6.2	Anwenden der Format-Methode .....	317
4.6.3	Stringinterpolation .....	318
4.7	Praxisbeispiele .....	319
4.7.1	Zeichenketten verarbeiten .....	319
4.7.2	Zeichenketten mit StringBuilder addieren .....	322
4.7.3	Reguläre Ausdrücke testen .....	325
4.7.4	Methodenaufrufe mit Array-Parametern .....	327
<b>5</b>	<b>Weitere Sprachfeatures .....</b>	<b>331</b>
5.1	Namespaces (Namensräume) .....	331
5.1.1	Ein kleiner Überblick .....	331
5.1.2	Einen eigenen Namespace einrichten .....	332
5.1.3	Die using-Anweisung .....	333
5.1.4	Namespace Alias .....	334
5.1.5	Namespace Alias Qualifizierer .....	334
5.2	Operatorenüberladung .....	335
5.2.1	Syntaxregeln .....	335
5.2.2	Praktische Anwendung .....	336
5.3	Collections (Auflistungen) .....	337
5.3.1	Die Schnittstelle IEnumerable .....	337
5.3.2	ArrayList .....	339
5.3.3	Hashtable .....	341
5.3.4	Indexer .....	341
5.4	Generics .....	343
5.4.1	Klassische Vorgehensweise .....	344
5.4.2	Generics bieten Typsicherheit .....	345

5.4.3	Generische Methoden .....	346
5.4.4	Iteratoren .....	347
5.5	Generische Collections .....	348
5.5.1	List-Collection statt ArrayList .....	348
5.5.2	Vorteile generischer Collections .....	349
5.5.3	Constraints .....	349
5.6	Das Prinzip der Delegates .....	349
5.6.1	Delegates sind Methodenzeiger .....	350
5.6.2	Einen Delegate-Typ deklarieren .....	350
5.6.3	Ein Delegate-Objekt erzeugen .....	350
5.6.4	Delegates vereinfacht instanziiieren .....	352
5.6.5	Anonyme Methoden .....	353
5.6.6	Lambda-Ausdrücke .....	354
5.6.7	Lambda-Ausdrücke in der Task Parallel Library .....	356
5.7	Dynamische Programmierung .....	358
5.7.1	Wozu dynamische Programmierung? .....	358
5.7.2	Das Prinzip der dynamischen Programmierung .....	358
5.7.3	Optionale Parameter sind hilfreich .....	361
5.7.4	Kovarianz und Kontravarianz .....	362
5.8	Weitere Datentypen .....	362
5.8.1	BigInteger .....	362
5.8.2	Complex .....	365
5.8.3	Tuple<> .....	365
5.8.4	SortedSet<> .....	366
5.9	Praxisbeispiele .....	367
5.9.1	ArrayList versus generische List .....	367
5.9.2	Generische IEnumerable-Interfaces implementieren .....	371
5.9.3	Delegates, anonyme Methoden, Lambda Expressions .....	374
5.9.4	Dynamischer Zugriff auf COM Interop .....	378
<b>6</b>	<b>Einführung in LINQ .....</b>	<b>381</b>
6.1	LINQ-Grundlagen .....	381
6.1.1	Die LINQ-Architektur .....	381
6.1.2	Anonyme Typen .....	382
6.1.3	Erweiterungsmethoden .....	384
6.2	Abfragen mit LINQ to Objects .....	385
6.2.1	Grundlegendes zur LINQ-Syntax .....	385
6.2.2	Zwei alternative Schreibweisen von LINQ Abfragen .....	386

6.2.3	Übersicht der wichtigsten Abfrage-Operatoren .....	387
6.3	LINQ-Abfragen im Detail .....	388
6.3.1	Die Projektionsoperatoren Select und SelectMany .....	389
6.3.2	Der Restriktionsoperator Where .....	390
6.3.3	Die Sortierungsoperatoren OrderBy und ThenBy .....	391
6.3.4	Der Gruppierungsoperator GroupBy .....	392
6.3.5	Verknüpfen mit Join .....	395
6.3.6	Aggregat-Operatoren .....	395
6.3.7	Verzögertes Ausführen von LINQ-Abfragen .....	397
6.3.8	Konvertierungsmethoden .....	398
6.3.9	Abfragen mit PLINQ .....	398
6.4	Praxisbeispiele .....	401
6.4.1	Die Syntax von LINQ-Abfragen verstehen .....	401
6.4.2	Aggregat-Abfragen mit LINQ .....	404
6.4.3	LINQ im Schnelldurchgang erlernen .....	407
6.4.4	Strings mit LINQ abfragen und filtern .....	409
6.4.5	Duplikate aus einer Liste oder einem Array entfernen .....	410
6.4.6	Arrays mit LINQ initialisieren .....	413
6.4.7	Arrays per LINQ mit Zufallszahlen füllen .....	415
6.4.8	Einen String mit Wiederholmuster erzeugen .....	417
6.4.9	Mit LINQ Zahlen und Strings sortieren .....	418
6.4.10	Mit LINQ Collections von Objekten sortieren .....	419
6.4.11	Ergebnisse von LINQ-Abfragen in ein Array kopieren .....	422

## Teil II: Technologien

<b>7</b>	<b>Zugriff auf das Dateisystem .....</b>	<b>425</b>
7.1	Grundlagen .....	425
7.1.1	Klassen für den Zugriff auf das Dateisystem .....	426
7.1.2	Statische versus Instanzen-Klasse .....	426
7.2	Übersichten .....	427
7.2.1	Methoden der Directory-Klasse .....	427
7.2.2	Methoden eines DirectoryInfo-Objekts .....	428
7.2.3	Eigenschaften eines DirectoryInfo-Objekts .....	428
7.2.4	Methoden der File-Klasse .....	428
7.2.5	Methoden eines FileInfo-Objekts .....	429
7.2.6	Eigenschaften eines FileInfo-Objekts .....	430

7.3	Operationen auf Verzeichnisebene .....	430
7.3.1	Existenz eines Verzeichnisses/einer Datei feststellen .....	430
7.3.2	Verzeichnisse erzeugen und löschen .....	431
7.3.3	Verzeichnisse verschieben und umbenennen .....	431
7.3.4	Aktuelles Verzeichnis bestimmen .....	432
7.3.5	Unterverzeichnisse ermitteln .....	432
7.3.6	Alle Laufwerke ermitteln .....	432
7.3.7	Dateien kopieren und verschieben .....	433
7.3.8	Dateien umbenennen .....	434
7.3.9	Dateiattribute feststellen .....	434
7.3.10	Verzeichnis einer Datei ermitteln .....	436
7.3.11	Alle im Verzeichnis enthaltenen Dateien ermitteln .....	436
7.3.12	Dateien und Unterverzeichnisse ermitteln .....	436
7.4	Zugriffsberechtigungen .....	437
7.4.1	ACL und ACE .....	437
7.4.2	SetAccessControl-Methode .....	438
7.4.3	Zugriffsrechte anzeigen .....	438
7.5	Weitere wichtige Klassen .....	439
7.5.1	Die Path-Klasse .....	439
7.5.2	Die Klasse FileSystemWatcher .....	440
7.6	Datei- und Verzeichnisdialoge .....	441
7.6.1	OpenFileDialog und SaveFileDialog .....	442
7.6.2	FolderBrowserDialog .....	443
7.7	Praxisbeispiele .....	444
7.7.1	Infos über Verzeichnisse und Dateien gewinnen .....	444
7.7.2	Eine Verzeichnisstruktur in die TreeView einlesen .....	448
7.7.3	Mit LINQ und RegEx Verzeichnisbäume durchsuchen .....	450
<b>8</b>	<b>Dateien lesen und schreiben .....</b>	<b>455</b>
8.1	Grundprinzip der Datenpersistenz .....	455
8.1.1	Dateien und Streams .....	455
8.1.2	Die wichtigsten Klassen .....	456
8.1.3	Erzeugen eines Streams .....	457
8.2	Dateiparameter .....	457
8.2.1	FileAccess .....	457
8.2.2	FileMode .....	457
8.2.3	FileShare .....	458

8.3	Textdateien .....	458
8.3.1	Eine Textdatei beschreiben bzw. neu anlegen .....	458
8.3.2	Eine Textdatei lesen .....	460
8.4	Binärdateien .....	461
8.4.1	Lese-/Schreibzugriff .....	461
8.4.2	Die Methoden ReadAllBytes und WriteAllBytes .....	462
8.4.3	Erzeugen von BinaryReader/BinaryWriter .....	462
8.5	Sequenzielle Dateien .....	463
8.5.1	Lesen und schreiben von strukturierten Daten .....	463
8.5.2	Serialisieren von Objekten .....	464
8.6	Dateien verschlüsseln und komprimieren .....	465
8.6.1	Das Methodenpärchen Encrypt/Decrypt .....	465
8.6.2	Verschlüsseln unter Windows Vista/7/8/10 .....	465
8.6.3	Verschlüsseln mit der CryptoStream-Klasse .....	466
8.6.4	Dateien komprimieren .....	467
8.7	Memory Mapped Files .....	468
8.7.1	Grundprinzip .....	468
8.7.2	Erzeugen eines MMF .....	469
8.7.3	Erstellen eines Map View .....	470
8.8	Praxisbeispiele .....	471
8.8.1	Auf eine Textdatei zugreifen .....	471
8.8.2	Einen Objektbaum persistent speichern .....	474
8.8.3	Ein Memory Mapped File (MMF) verwenden .....	481
8.8.4	Hex-Dezimal-Bytes-Konverter .....	483
8.8.5	Eine Datei verschlüsseln .....	487
8.8.6	Eine Datei komprimieren .....	490
8.8.7	Echte ZIP-Dateien erstellen .....	492
8.8.8	PDFs erstellen/exportieren .....	494
8.8.9	Eine CSV-Datei erstellen .....	497
8.8.10	Eine CSV-Datei mit LINQ lesen und auswerten .....	500
8.8.11	Einen korrekten Dateinamen erzeugen .....	503
<b>9</b>	<b>Asynchrone Programmierung .....</b>	<b>505</b>
9.1	Übersicht .....	505
9.1.1	Multitasking versus Multithreading .....	506
9.1.2	Deadlocks .....	507
9.1.3	Racing .....	507