

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>9</b>
1.1. C#: Eine Programmiersprache mit vielfältigen Anwendungsmöglichkeiten	10
1.2. Die Entwicklungsgeschichte von C#	11
1.3. C# und .NET Framework	13
1.4. Vom Code zum fertigen Programm: die Kompilierung	14
<b>2. Visual Studio: die passende IDE für die Entwicklung mit C#</b>	<b>17</b>
2.1. Visual Studio installieren	21
2.2. Ein neues Projekt mit Visual Studio erstellen	23
<b>3. Das erste Programm in C#</b>	<b>26</b>
3.1. Eine Ausgabe mit C# erzeugen	26
3.2. So ist ein C#-Programm aufgebaut	29
3.3. Den Programmcode kompilieren und ausführen	31
3.4. Kommentare erleichtern das Verständnis des Codes	34
3.5. Übungsaufgabe: Das erste Programm erweitern	35
<b>4. Variablen</b>	<b>39</b>
4.1. Die Verwendung von Variablen in der Informatik	39
4.2. Variablentypen: wichtig für das Programmieren in C#	42
4.3. Variablen in C# verwenden	44
4.4. Berechnungen mit arithmetischen Operatoren durchführen	47
4.5. Eingaben des Anwenders in einer Variablen aufnehmen	52
4.6. Übungsaufgabe: Programme mit Variablen erstellen	56
<b>5. Datenstrukturen in C#</b>	<b>60</b>
5.1. Arrays für eine feste Anzahl an Werten	61
5.2. Mehrdimensionale Arrays	65
5.3. Listen für eine variable Anzahl an Werten	67
5.4. Listen für identische Datentypen	69

5.5. Listen für unterschiedliche Datentypen .....	72
5.6. Übungsaufgabe: mit Datenstrukturen arbeiten .....	75
<b>6. If-Abfragen: mehrere Alternativen in das Programm einfügen.....</b>	<b>79</b>
6.1. Der Aufbau der if-Abfrage.....	79
6.2. Vergleichsoperatoren .....	81
6.3. Logische Operatoren.....	85
6.4. Weitere Optionen zur if- Abfrage hinzufügen.....	88
6.5. Übungsaufgabe: Programme mit Verzweigungen erstellen .....	91
<b>7. Schleifen: Befehle automatisch wiederholen lassen .....</b>	<b>97</b>
7.1. Den Ablauf mit einer while-Schleife steuern .....	97
7.2. Do-while-Schleife: die fußgesteuerte Alternative .....	103
7.3. Die for-Schleife für eine feste Anzahl an Wiederholungen .....	105
7.4. Foreach-Schleifen für Arrays und andere Datenstrukturen .....	106
7.5. Übungsaufgabe: eigene Schleifen erstellen .....	110
<b>8. Grundzüge der objektorientierten Programmierung in C#.....</b>	<b>116</b>
8.1. Was bedeutet objektorientierte Programmierung? .....	116
8.2. Die Klasse: grundlegende Struktur für objektorientierte Programme .....	119
8.3. Ein Objekt als Instanz einer Klasse erzeugen .....	121
8.4. Übungsaufgabe: objektorientierte Programme schreiben .....	124
<b>9. Methoden in C# verwenden.....</b>	<b>128</b>
9.1. Welche Vorteile bieten Methoden? .....	128
9.2. Der Aufbau einer Methode in C# .....	130
9.3. Methoden mit Übergabewerten .....	135
9.4. Methoden mit Rückgabewerten .....	138
9.5. Funktionen und Methoden: Worin bestehen die Unterschiede? ....	142
9.6. Methoden wie klassische Funktionen verwenden.....	144
9.7. Übungsaufgabe: Methoden im Programm verwenden.....	146
<b>10. Weiterführende Prinzipien der objektorientierten Programmierung ...</b>	<b>154</b>

10.1. Konstruktoren für die Instanziierung der Objekte verwenden .....	154
10.2. Daten kapseln: besserer Schutz für die Attribute der Objekte ....	157
10.3. Vererbung: Klassen von anderen Klassen ableiten.....	162
10.4. Klassen in Module auslagern .....	169
10.5. Übungsaufgabe: fortgeschrittene objektorientierte Programme..	172
<b>11. Vorgefertigte Funktionen im Programm verwenden .....</b>	<b>176</b>
11.1. Eine Übersicht über die verschiedenen Möglichkeiten .....	177
11.2. Eine vorgefertigte Funktion im Programm verwenden.....	180
11.3. Übungsaufgaben: passende Funktionen für das Programm suchen .....	184
<b>12. Fehler im Programm beseitigen .....</b>	<b>189</b>
12.1. Syntax-, Laufzeit- und Semantikfehler: Worin bestehen die Unterschiede?.....	189
12.2. Syntaxfehler beheben .....	192
12.3. Ausnahmen für Laufzeitfehler erstellen .....	193
12.4. Semantikfehler durch das Debugging finden .....	200
<b>13. Daten dauerhaft abspeichern.....</b>	<b>206</b>
13.1. Verschiedene Möglichkeiten für die Datenspeicherung.....	206
13.2. Was ist eine Datenbank? .....	208
13.3. Verschiedene Organisationsformen für Datenbanken.....	210
13.4. Ein passendes Datenbankmanagementsystem für C#-Programme .....	212
13.5. SQL-Befehle: wichtig für die Arbeit mit Datenbanken.....	216
13.6. Eine Datenbank erzeugen .....	217
13.7. Tabellen, Spalten und Felder erstellen .....	221
13.8. Werte aus der Tabelle auslesen und verändern .....	224
13.9. Eine Verbindung zwischen dem Programm und der Datenbank herstellen .....	228
13.10. SQL-Befehle im Programm verwenden .....	231

13.11. Übungsaufgabe: Daten in Datenbanken speichern .....	238
<b>14. GUI: moderne Programme mit einzelnen Fenstern erstellen.....</b>	<b>244</b>
14.1. Verschiedene Techniken für die Erstellung von User Interfaces .....	245
14.2. Das erste Fenster mit WPF erstellen .....	247
14.3. Buttons im Fenster verwenden .....	253
14.4. Ein neues Fenster öffnen .....	259
14.5. Inhalte in einem Fenster dynamisch anzeigen lassen.....	262
14.6. Übungsaufgabe: ein kleines Quiz erstellen .....	269
<b>15. Anwendungsbeispiel: ein kleines Spiel programmieren.....</b>	<b>273</b>
15.1. Das Hauptfenster gestalten .....	274
15.2. Das Spiel beginnen .....	286
15.3. Das Spielfeld aktualisieren .....	294
15.4. Einen Zug durchführen .....	300
15.4. Den Spielstand speichern und laden .....	315
15.5 Überblick: Alle Klassen und Dateien des Spiels .....	317
<b>16. Ausblick: die Programmierkenntnisse weiter ausbauen.....</b>	<b>336</b>