

INHALTSVERZEICHNIS

Kapitel 1: Ein guter Start ist der halbe Sieg

Compiler und Entwicklungsumgebungen

Seite 27

Schrödinger steht am Anfang seines neuen Jobs als C#-Entwickler. Sein Problem: Er kann noch gar kein C#. Sein erster Schritt zur Lösung: Er hat sich Hilfe geholt. Und er hat richtig Lust auf die Sache. Beste Voraussetzungen also. Jetzt ist der zweite Schritt an der Reihe: Installieren! Aber was?

| | | | |
|---------------------------------------|-----------|--------------------------|-----------|
| Compiler und Compiler | 28 | Der Spaß geht los! | 37 |
| Hallo Schrödinger | 31 | Theorie und Praxis | 40 |
| Du brauchst eine IDE! | 35 | Was gelernt! | 42 |
| Visual Studio Community Edition | 36 | | |

Kapitel 2: Ein netter Typ

Datentypen und deren Behandlung

Seite 43

Wie heißt es so schön, guten Freunden gibt man doch einen Kaffee – oder? Ich würde sagen, ich stelle dir ein paar freundliche Typen vor, und am Ende trinken wir eine schöne heiße Tasse Kaffee. Das klingt gemütlich, findet Schrödinger und entspannt sich. Gemütlich? Sieben Typen allein für Zahlen? Konvertieren, kompatibel, Kommentare? Doch Schrödinger bleibt locker und lernt dabei sogar noch, mit Kamelen umzugehen.

| | | | |
|-----------------------------------|-----------|---|-----------|
| Dieses Glas für diesen Wein | 44 | Das ständige Hin und Her zwischen | |
| Grundlagen im Kamelreiten | 48 | ja und nein | 58 |
| Übungen für den Barkeeper | 50 | Gut kommentieren! | 59 |
| Rechnen mit Transvestiten | 51 | Kommentare im Einsatz | 60 |
| Ja oder nein? | 56 | Andere für sich denken lassen | 60 |
| Was gibt's zu essen? | 57 | Compiler-Spiele | 61 |
| | | Viele neue Freunde | 62 |

Kapitel 3: Alles unter Kontrolle

Bedingungen und Schleifen

Seite 63

Schrödinger findet zwar Datentypen ganz in Ordnung, jedoch dämmert ihm, dass er mit ein paar Variablen und der Ausgabe auf der Konsole wohl noch keine Spieleentwicklerkarriere starten kann. Beim Spielen gibt es so viele Entscheidungen zu treffen. Angreifen oder abhauen? Stehen oder gehen? Diese Entscheidungen müssen doch auch mit C# programmiert werden können! In diesem Kapitel gewinnt Schrödinger die Kontrolle über alle Abläufe im Code.

| | | | |
|--|-----------|---|-----------|
| Bedingungen | 64 | Die ganze Welt ist Mathematik und aller | |
| In der Kürze liegt die Würze | 67 | guten Dinge sind drei vier | 81 |
| Durch Variationen bleibt es interessant | 68 | Schau's dir an mit dem Debugger | 82 |
| Der Herr der Fernbedienung | 70 | Solange du nicht fertig bist, weitermachen | 83 |
| Ist noch Bier da? | 72 | Ich habe es mir anders überlegt | 84 |
| Einer von vielen | 73 | Oder mach doch weiter... | 85 |
| Zwillinge | 75 | Zurück zu den Schuhsschränken | 86 |
| Ein Schuhschrank muss her | 78 | Wenn aus einem Schuhschrank | |
| Arbeiten in den Tiefen des Schuhschranks – von Kopf bis Fuß | 79 | eine Lagerhalle wird | 87 |
| | | Wiederholung, Wiederholung! | 89 |
| | | Code muss man auch lesen können | 90 |

Kapitel 4: Sexy Unterwäsche – von kleinen Teilen bis gar nichts

Strings, Characters und Nullable Types

Seite 93

Selbst das kleinste Programm arbeitet mit Zeichenketten. Es wird Zeit, dass Schrödinger den Datentyp dafür kennenlernt: string. Aber wer den Buchstaben nicht ehrt, ist des Strings nicht wert. Schrödinger schaut sich auch einzelne Zeichen ganz genau an. Was er nicht gedacht hätte: Es gibt einen engen Zusammenhang zu Zahlen. Jetzt schaut er noch genauer hin und lernt sogar, mit nichts umzugehen, rein gar nichts – mit Variablen, die rein gar keinen Wert haben.

| | | | |
|--|------------|---|------------|
| Zeichenketten – Strings | 94 | Etwas Besonderes sollte es sein | 101 |
| Kleine Teile – einzelne Zeichen | 95 | Verdrehte Welt | 102 |
| Kleine und große Teile | 96 | Sein oder nicht sein? | 105 |
| Einfacher und schneller | 97 | Nichts im Einsatz | 107 |
| Neuerungen in C# 6 bei der String-Formatierung | 100 | Damit bei so viel null nichts verloren geht | 108 |

Kapitel 5: Eine endliche Geschichte

Enumerationen

Seite 109

Allzu oft kommt es vor, dass Schrödinger nicht nur mit Zahlen und Texten arbeiten möchte, sondern mit bestimmten Auswahlmöglichkeiten, mit einer von vielen. Wie bei einer Ampel, die genau Grün, Gelb und Rot zeigt, oder wie die Völker bei World of Warcraft, wobei die ja immer wieder mal erweitert werden. Wie auch immer, ein Konstrukt muss her, das Auswahlmöglichkeiten erlaubt, ohne sie unbedingt in Zahlen abzubilden. Da gibt es doch etwas... Enumerationen.

| | | | |
|--|------------|--|------------|
| Rot – Gelb – Grün | 110 | WoW-Völker | 116 |
| Tageweise | 112 | Auf wenige Sätze heruntergebrochen | 118 |
| Tell me why I don't like mondays | 115 | | |

Kapitel 6: Teile und herrsche

Methoden

Seite 119

Schrödingers Code wird immer länger, und für ihn wird es auch immer schwieriger, den Überblick zu behalten. Ganz zu schweigen davon, immer gute Variablennamen zu finden, die er selbst nicht schon vergeben hat. Schrödinger wird nun also lernen, wie er seinen Code mithilfe von Funktionen – oder formal etwas richtiger: Methoden – besser strukturieren und Teile davon sogar wiederverwenden kann. Copy & Paste von Codeteilen gehört ab jetzt der Vergangenheit an.

| | | | |
|---|------------|--|------------|
| Teilen statt Kopieren | 120 | Tauschgeschäfte, die nicht funktionieren | 131 |
| Originale und überteuerte Kopien | 123 | Ich will das ganz anders oder auch | |
| Eins ist nicht genug | 126 | gar nicht – Methoden überladen | 132 |
| Ich rechne mit dir | 127 | Das Ganze noch einmal umgerührt | 135 |
| Wenn sich nichts bewegt und alles statisch ist | 128 | Ein knurrender Magen spornt bestimmt | |
| Ich hätte gerne das Original! | 128 | zu Höchstleistungen an | 137 |
| Sommerschlussverkauf – alles muss raus | 129 | Eine kleine Zusammenfassung für dich | 138 |

Kapitel 7: Klassengesellschaft

Objekte, Eigenschaften und Sichtbarkeiten

Seite 139

Schrödinger kann zwar Methoden schreiben und doppelten Code vermeiden, doch so wirklich will sich keine Übersichtlichkeit einstellen. Es muss doch möglich sein, Programmcode auf mehrere Dateien aufzuteilen und Variablen nicht immer »global« definieren zu müssen, egal ob sie nur in bestimmten Bereichen benutzt werden. Das schreit nach Objekten! In Objekten kommt zusammen, was zusammengehört: Eigenschaften und Methoden, gekapselt in einem Objekt und mit ein bisschen Geheimniskrämerei.

| | | | |
|--|-----|---|-----|
| Mein Alter, meine Augenfarbe, mein Geburtsdatum | 140 | Zusammen und doch getrennt | 156 |
| Eine Aufgabe für den Accessor | 144 | Laufen, kämpfen, sterben | 158 |
| Ich sehe was, was du nicht siehst | 145 | Vom Leben und Sterben | 159 |
| Eigenschaften aufpoliert und bereit für die Bühne | 146 | Geburtenkontrolle | 160 |
| Tanzen mit Elvis – wenn keiner da ist, ist keiner da | 148 | Mehrlingsgeburt | 163 |
| Geheimniskrämerei und Kontrollfreak | 149 | Partielle Klassen und Strukturen | 164 |
| Darf ich jetzt oder nicht? | 150 | Pfeile über Pfeile – oder Vektoren, wie es mathematisch heißt | 165 |
| Zusammen was zusammengehört! | 154 | Meine partiellen Daten | 166 |
| | | Seltenes nochmal betrachtet | 167 |
| | | Gelernt ist gelernt! | 170 |

Kapitel 8: Es wird Zeit für Übersicht!

Namespaces

Seite 171

Es gibt Tausende Klassen im .NET-Framework, und auch in Softwareprojekten sind es gerne Hunderte. Daher schadet es nicht, dass Schrödinger das Konzept der Namespaces kennenlernen. Nicht nur um die eigenen Klassen zu strukturieren, sondern auch um zu verstehen, wo er welche Funktionalitäten aus dem Framework suchen muss.

| | | | |
|---|-----|---|-----|
| Eine Ordnung für die Klassen | 172 | Wo sind nur diese Bausteine? | 179 |
| Was ist denn nur in diesem Namespace vorhanden? | 175 | Mathematik für Einsteiger | 181 |
| Vorhandene Systembausteine | 177 | Nochmals finden, was scheinbar nicht da ist | 182 |
| | | Zum Mitnehmen | 182 |

Kapitel 9: Erben ohne Sterben

Objektorientierte Programmierung

Seite 183

Dass er doppelten Code vermeiden soll, ist für Schrödinger schon lange nichts Neues mehr. Er kennt aber längst noch nicht alle Tricks dazu. Jetzt nimmt er sich Vererbung und Polymorphismus vor – wichtige Konzepte der Objektorientierung. Damit kann man doppelten Code vermeiden (ach, was!) und viele Funktionen einfach geschenkt bekommen.

| | | | |
|---|------------|--|------------|
| Geisterstunde | 184 | Geister haben viele Gestalten | 195 |
| Schleimgeister sind spezielle Geister | 186 | Geister, die sich nicht an die Regeln halten | 198 |
| Fünf vor zwölf | 188 | Gestaltwandler unter der Lupe | 199 |
| Geister fressen, Schleimgeister fressen, Kannibalen fressen – alles muss man einzeln machen | 194 | Nochmals drüber nachgedacht | 200 |
| Enterben | 195 | Hier noch ein Merkzettel | 204 |

Kapitel 10: Abstrakte Kunst

Abstrakte Klassen und Interfaces

Seite 205

Jetzt hat Schrödinger Vererbung und Polymorphismus verstanden und darf sich immer noch nicht »Meister der objektorientierten Programmierung« nennen. Warum nicht? Sein Kumpel kommt immer wieder auf das Thema Copy & Paste zu sprechen. Schrödinger soll doppelten Code noch konsequenter vermeiden: Gemeinsamkeiten zusammenfassen und durch Interfaces und abstrakte Klassen abbilden. Erst dann darf er sagen, dass er die Konzepte der objektorientierten Programmierung kennt.

| | | | |
|---|------------|--|------------|
| Unverstandene Künstler | 208 | Kaffeemaschine im Einsatz | 218 |
| Das Meisterwerk nochmals betrachtet | 210 | Eine Cola bitte | 220 |
| Abstrakte Kunst am Prüftisch | 211 | Freundin vs. Chef – Runde 1 | 222 |
| Allgemein ist konkret genug | 213 | Bei perfekter Verwendung... | 223 |
| Fabrikarbeit | 214 | Freundin vs. Chef – Runde 2 | 224 |
| Alles unter einem Dach | 215 | Freundin vs. Chef – Runde 3 | 226 |
| Kaffee oder Tee? Oder doch lieber eine Cola? | 216 | Abstraktion und Interfaces auf einen Blick | 227 |

Kapitel 11: Airbags können Leben retten

Exceptionhandling

Seite 229

Wenn bloß Schrödingers Programme nicht immer gleich abstürzen würden, sobald der Benutzer etwas eingibt, das er nicht sollte! Das wäre schon eine feine Sache. Beim Programmieren gibt es immer Ausnahmezustände, mit denen man rechnen muss. Deshalb heißt es: Mit Fehlern umgehen lernen. Erstens, die Konzepte kennenzulernen, die C# dafür zu bieten hat. Zweitens, dann natürlich dran denken. Also, liebe Leser, lieber Schrödinger: Fehlerbehandlung nicht vergessen!

| | | |
|--|------------|------------------------------------|
| Mach's stabil! | 230 | Bezahlung ohne Ware – |
| Einen Versuch war es wert | 232 | ArgumentNullException |
| Nur unter bestimmten Umständen | 235 | Bewusste Fehler |
| Fehler über Fehler | 236 | Selbst definierte Fehler |
| Über das Klettern auf Bäume | 240 | Fehler in freier Wildbahn |
| Klettern auf nicht vorhandene Bäume – | | Das Matruschka-Prinzip |
| NullReferenceException | 240 | Alles noch einmal aufgerollt |
| Auf Sträucher klettern – FormatException | 241 | Dein Fehler-Cheat-Sheet |
| Sträucher im Sägewerk – ArgumentException | 242 | |

Kapitel 12: Ein ordentliches Ablagesystem muss her

Collections und Laufzeitkomplexität

Seite 253

Arrays sind zwar nett, aber oftmals auch sehr unflexibel. Vor allem die fixe Größe gefällt Schrödinger überhaupt nicht und treibt ihn regelmäßig an den Rand des Wahnsinns. Eine Alternative zu Arrays muss her! Eine, die das Leben vereinfacht, in der Schrödinger eine beliebige, bei der Definition noch unbekannte Anzahl von Elementen verwaltung kann. Und schnell soll sie sein, diese Alternative. Sind Schrödingers Anforderungen der Wunsch nach einer Eier legenden Wollmilchsau?

| | | | |
|--|------------|--|------------|
| Je größer der Schuhschrank, desto länger die Suche | 254 | Eine Array-Liste | 267 |
| Komplizierte Laufschuhe | 255 | Ringboxen | 268 |
| Geschwindigkeitsprognosen | 258 | Listige Arrays und ihre Eigenheiten | 269 |
| Es muss nicht immer gleich quadratisch sein | 260 | Listige Arrays und ihre Verwendung | 269 |
| Geschwindigkeitseinschätzung und Buchstabensuppe | 263 | The Need for Speed | 270 |
| Selbstwachsende Schuhschränke | 266 | Es wird konkreter | 271 |
| | | Sortieren bringt Geschwindigkeit – SortedList | 272 |
| | | Listenreiche Arbeit | 274 |

| | | | |
|--|------------|---|------------|
| Es geht noch schneller! | 276 | Schnelles Arbeiten mit Sets | 288 |
| Im Rausch der Geschwindigkeit | 278 | Das große Bild | 290 |
| Dictionary-Initialisierung in C# 6 | 280 | Der große Test, das Geheimnis und die Verwunderung | 293 |
| Wörterbücher in der Anwendung | | Noch einmal durchleuchtet | 298 |
| ... oder was im Regelfall schiefgeht | 281 | Dein Merkzettel rund um die Collections aus Laufzeiten | 303 |
| Von Bäumen und Ästen | 285 | | |
| Ein Verwendungsbeispiel | 286 | | |
| Alles eindeutig – das HashSet | 287 | | |

Kapitel 13: Allgemein konkrete Implementierungen

Generizität

Seite 305

Schrödinger hat bereits bei den Collections ein bisschen mit generischen Typen gearbeitet. Doch noch weiß er nicht wirklich, was sich dahinter verbirgt und dass er selbst generische Typen und generische Methoden programmieren kann. Was ihm eine ganz neue und spannende Welt eröffnet.

| | | | |
|--|------------|---------------------------------------|------------|
| Konkrete Typen müssen nicht sein | 306 | Aus allgemein wird konkret | 318 |
| Das große Ganze | 307 | Hier kommt nicht jeder Typ rein. | 319 |
| Mülltrennung leicht gemacht | 308 | Ähnlich, aber nicht gleich! | 320 |
| Der Nächste bitte | 311 | Varianzen hin oder her | 322 |
| Allgemein, aber nicht für jeden! | 313 | Varianzen in der Praxis | 325 |
| Immer das Gleiche und doch etwas anderes | 315 | WoW im Simulator | 328 |
| Fabrikarbeit | 317 | Damit's auch hängen bleibt | 330 |

Kapitel 14: Linke Typen, auf die man sich verlassen kann

LINQ

Seite 331

Algorithmen sind für Schrödinger manchmal ganz schön aufwendig, und wenn es knifflig wird, neigt er zur Ungeduld. Gäbe es doch etwas, das das Suchen von Elementen, das Abgleichen von Listen oder Umwandeln einer Liste von Elementen in andere Typen einfacher macht. Das wäre wirklich vorteilhaft! Und tatsächlich gibt es so etwas. Es heißt LINQ – Language-Integrated Query.

| | | | |
|---|------------|--------------------------------------|------------|
| Linke Typen, auf die man sich verlassen kann | 332 | Listen zusammenführen | 339 |
| Shoppen in WoW | 335 | Fix geLINQt statt handverlesen | 347 |
| Gesund oder gut essen? | 338 | Merkzettel | 350 |

Kapitel 15: Blumen für die Dame

Delegaten und Ereignisse

Seite 351

Manchmal wünschen wir uns alle, bestimmte Arbeiten einfach delegieren zu können. Ging es im Leben doch auch nur so einfach wie beim Programmieren. Schrödinger wird bestimmt bald versuchen, die Delegaten, die er nun in C# kennenlernen wird, ins echte Leben zu übertragen und viel Arbeit an seine Freundin abzugeben. Hoffentlich kommen dann nicht ungeahnte Ereignisse auf ihn zu.

| | | | |
|---------------------------------------|------------|--------------------------------------|------------|
| Ein Butler übernimmt die Arbeit | 352 | Eine Runde für alle | 364 |
| Im Strudel der Methoden | 355 | Auf in die Bar! | 365 |
| Die Butlerschule | 358 | Wiederholung, Wiederholung | 369 |
| Ereignisreiche Tage | 361 | Die delegierte Zusammenfassung | 372 |

Kapitel 16: Der Standard ist nicht genug

Extension-Methoden und Lambda-Expressions

Seite 373

Schrödinger hat Gefallen gefunden an LINQ und den anderen Konzepten, die er bis jetzt kennengelernt hat. Aber sein Ausbilder hat ihn besser durchschaut als er sich selbst: Er hätte es gerne kürzer. Noch weniger zu tippen, ja, gar nicht zu tippen, wäre eigentlich am schönsten. Ob es da etwas gibt?

| | | | |
|---|------------|--|------------|
| Extension-Methoden | 374 | Gruppieren | 388 |
| Auf die Größe kommt es an | 378 | Verknüpfen | 389 |
| Erweiterungen nochmals durchschaut | 380 | Gruppieren und Verknüpfen kombiniert | 390 |
| Softwareentwicklung mit Lambdas | 382 | Left Join | 391 |
| Lambda-Expressions auf Collections loslassen | 385 | VerLINQte LAMbdAS | 393 |
| Ein Ausritt auf Lamas | 386 | Lamas im Schnelldurchlauf | 396 |
| Filtrern | 386 | | |

Kapitel 17: Die Magie der Attribute

Arbeiten mit Attributen

Seite 397

»Attribute sind kleine, nette, zusätzliche Elemente, die an Datentypen, Methoden oder Eigenschaften hängen. Sie leben in Symbiose mit deinen Klassen und deren Elementen und wie Fabelwesen sind sie nicht direkt sichtbar, sondern nur unter bestimmten Umständen. Nämlich genau dann, wenn du einen Typ (eine Klasse) selbst ganz genau betrachtest.« Und dann soll es auch noch um Psychologie gehen, und um Magie. Schrödinger ist skeptisch. Ob der Bernhard das alles ernst meint?

| | | | |
|--|------------|--|------------|
| Die Welt der Attribute | 398 | Der Attribut-Meister erstellt eigene Attribute! | 410 |
| Die Magie erleben | 400 | Meine Klasse, meine Zeichen | 412 |
| Das Ablaufdatum-Attribut | 402 | Selbstreflexion | 414 |
| Die Magie selbst erleben | 403 | Die Psychologie lehrt uns: Wiederholung | |
| Eine magische Reise in dein Selbst | 404 | ist wichtig! | 418 |
| In den Tiefen des Kaninchenbaus | 407 | | |

Kapitel 18: Ich muss mal raus

Dateizugriff und Streams

Seite 419

Schrödinger denkt bei jedem Programm an WoW und daran, was ihm an Fähigkeiten noch fehlt, um bei Blizzard punkten zu können. Da fällt ihm auf, dass er noch gar keine Spielstände speichern könnte, da er nicht weiß, wie er auf Dateien zugreift, Daten auf der Festplatte speichert oder auch eine Datei aus dem Internet herunterladen kann. Das sollte doch ganz einfach funktionieren. Nur wie?

| | | | |
|--|------------|---|------------|
| Daten speichern | 420 | Wenn das Fließband nicht ganz richtig läuft | 445 |
| Rundherum oder direkt rein | 421 | Dem Fließband vorgeschalteter Fleischwolf | 449 |
| Rein in die Dose, Deckel drauf und fertig | 423 | Nutze die Attribut-Magie! | 451 |
| Deine Geheimnisse sind bei mir nicht sicher | 424 | Das Formatter-Prinzip | 452 |
| Das Mysterium der Dateiendungen | 427 | X(M)L entspricht XXL | 453 |
| Das Gleiche und doch etwas anders | 430 | Die kleinste Größe – JSON | 455 |
| Das Lexikon vom Erstellen, Lesen, Schreiben, Umbenennen | 431 | Wir sind viele | 457 |
| Ran an die Tastatur, rein in die Dateien | 436 | Das World Wide Web. Unendliche Weiten | 462 |
| Von der Sandburg zum Wolkenkratzer | 438 | Deine Seite, meine Seite | 464 |
| Fließbandarbeit | 442 | Probe, Probe, Leseprobe | 466 |
| | | Punkt für Punkt für's Hirn | 468 |

Kapitel 19: Sag doch einfach, wenn du fertig bist

Asynchrone und parallele Programmierung

Seite 469

Schrödinger hat bereits gemerkt, dass Dinge, die mit dem Download von Dateien zu tun haben, lange dauern. Aber auch Algorithmen können ihre Zeit brauchen. Gleichzeitig hat sein PC mehrere Kerne, die sich meistens langweilen. Das muss sich doch zusammenbringen lassen? Er hat gehört, dass parallele und asynchrone Programmierung nicht so einfach sind, doch mit den richtigen Tricks ist es plötzlich gar nicht schwer.

| | | | |
|--|------------|---|------------|
| Zum Beispiel ein Download-Programm | 470 | Wenn jeder mit anpackt, dann geht alles schneller | 493 |
| Asynchroner Start mit Ereignis bei Fertigstellung | 472 | Rückzug bei Kriegsspielen | 496 |
| Subjektive Geschwindigkeiten und Probleme mit dem Warten | 474 | async/await/cancel | 498 |
| Auf der Suche nach der absoluten Geschwindigkeit | 477 | Unkoordinierte Koordination | 500 |
| Es geht auch einfacher! | 480 | Anders und doch gleich | 505 |
| Was so alles im Hintergrund laufen kann | 485 | Gemeinsam Kuchen backen | 506 |
| Gemeinsam geht es schneller | 487 | Wenn das Klo besetzt ist | 511 |
| Jetzt wird es etwas magisch | 491 | Das Producer-Consumer-Problem | 511 |
| | | Dein Spickzettel | 517 |

Kapitel 20: Nimm doch, was andere schon gemacht haben

Die Paketverwaltung NuGet

Seite 519

Beim Programmieren ist es doch so, dass die meisten Probleme schon von anderen Entwicklern gelöst wurden. Oftmals gießen diese die Lösung dann in fertige Bibliotheken, die nur darauf warten, genutzt zu werden. Das wird Schrödinger bestimmt gefallen.

| | | | |
|------------------------------|------------|---------------------------------|------------|
| Bibliotheken für Code | 520 | Die Welt ist schon fertig | 526 |
| Fremden Code aufspüren | 523 | | |

Kapitel 21: Die schönen Seiten des Lebens

Einführung in XAML

Seite 527

Auf die Dauer werden Konsolenanwendungen öde. Auch wenn Schrödinger geduldig die Programmierkonzepte von C# gelernt hat, wird es Zeit, sich der Oberflächenprogrammierung zu widmen. Endlich Fenster und Schaltflächen für seine Programme! Die XAML-Technologie scheint da genau das Richtige zu sein! Wird sie doch sowohl für Silverlight, Windows, Windows Phone als auch für die Windows Universal Platform verwendet. Doch wie in diese große neue Welt am besten eintauchen?

| | | | |
|--|-----|---|-----|
| Oberflächenprogrammierung | 528 | Alles schön am Raster ausrichten | 557 |
| Hinzufügen der Komponenten für die Universal | | Das sieht doch schon aus wie eine | |
| Windows Platform Apps in Visual Studio | 530 | Anwendung | 559 |
| Diese X-Technologien | 532 | Ein Layout für eine App | 561 |
| Tabellen über Tabellen | 536 | Auf in die (App)Bar | 564 |
| Hallo Universal App | 539 | Die Ecken und Winkel in der Bar | 565 |
| Die App soll »Hallo« sagen | 540 | Einfach und wirksam | 566 |
| Schrödingers kreative Katze | 544 | Das ist alles eine Stilfrage | 568 |
| Buttons und Text ausrichten | 547 | Von der Seite in die Anwendung | 570 |
| Von Tabellen, Listen und Parkplätzen | 549 | Do you speak English, Koreanisch oder so? | |
| VariableSizedWrapGrid und RelativePanel – | | Schrödinger, I do! | 571 |
| zwei gute Teilnehmer | 552 | Die Welt der Sprachen | 573 |
| Die Mischung macht's! | 554 | Honey, I do! | 576 |
| Das gemischte Layout | 555 | Oberflächenprogrammierung auf einen Blick | 578 |

Kapitel 22: Models sind doch schön anzusehen

Das Model-View-ViewModel-Entwurfsmuster

Seite 579

Die Trennung von Code und Design ist ein wichtiges Konzept, so viel ist klar. Dies kann bereits mit den Code-Behind-Dateien erreicht werden. Aber für große Anwendungen mit WPF und XAML gibt es etwas Besseres: Das Entwurfsmuster Model-View-ViewModel, MVVM. Sieht schon mal schön symmetrisch aus. View steht bestimmt für Design. Das braucht Schrödinger also unbedingt, sonst kann er sich nicht Profi nennen.

| | | | |
|--|-----|--|-----|
| Einführung in MVVM | 580 | Eine Eigenschaft für alle Infos | 591 |
| Mein erstes eigenes Model | 584 | Wenn nur jeder wüsste, was er zu tun hätte | 593 |
| Datenbindung noch kürzer – als Seitenressource | 588 | Los geht's! Notify-Everybody | 596 |
| Eine Technik, sie alle zu binden! | 589 | Ein Laufsteg muss es sein! | 599 |

| | | | |
|---|------------|--|------------|
| Über Transvestiten und Bindungsprobleme | 606 | Die Größe der Kaffeetasse | 614 |
| Über Bindungsprobleme und deren Lösungen | 607 | Auf mein Kommando | 619 |
| Alleine oder zu zweit? | 608 | Kommandierende Butler | 621 |
| Aus Klein mach Groß und zurück | 609 | Dem Zufall das Kommando überlassen | 625 |
| Klein aber fein | 610 | MVVM Punkt für Punkt | 630 |

Kapitel 23: Stereotyp Schönheit

Universal Windows Platform Apps

Seite 631

Microsoft lässt nicht jede beliebige App in den Windows Store. Wie schon zu Zeiten von Windows 95 sollen auch bei Windows Store Apps die Anwendungen alle ähnlich zu bedienen sein. Nur haben sich die Regeln für die Designerstellung signifikant verändert. Hierzu gibt es Designrichtlinien, an die sich Schrödinger zu halten hat. Doch Microsoft sei Dank gibt es da etwas, das ihm schon viel Arbeit abnimmt.

| | | | |
|--|------------|---|------------|
| Heute dreht sich alles um Apps | 632 | Navigation einfach gemacht | 658 |
| Universal Windows Platform Apps – Planung | 633 | Navigation und App-Lebenszyklus | 659 |
| Ran an den Code – die Wetter-App | 638 | Einstellungen, fix und fertig | 660 |
| Visuelle Zustände als einfache Lösung | 644 | Bitte warten – please hold the line | 663 |
| Apps ohne Ende | 647 | Menüführung | 665 |
| Etwas mehr Komfort darf schon sein! | 649 | Deine Template10-Übung | 666 |
| Viele tolle Sachen am Template10 | 653 | Das ganze Layout auf einen Blick | 670 |

Kapitel 24: Toast-Notifications

Der Einsatz von Toast-Notifications

Seite 671

Ob Erinnerungen, kurze Benachrichtigungen oder Benutzerentscheidungen: Toast-Notifications werden eingesetzt, wenn es etwas zu sagen gibt, das sich den Nutzern über alle Anwendungen hinweg aufdrängen darf – und soll.

| | | | |
|--|------------|---|------------|
| Ich habe etwas Wichtiges zu sagen! | 672 | Die Verwendung deines individuellen Templates | 678 |
| Das Betriebssystem wird es schon richten | 672 | Das Feinste vom Feinen | 679 |
| Einfache Toast-Notifications | 673 | Kleine Änderungen mit großer Wirkung | 682 |
| Templates verzögert anzeigen | 675 | Minütlich grüßt die Notification | 683 |
| Eigene Notification-Templates | 676 | Deine Toast-Zusammenfassung | 686 |

Kapitel 25: Live is Live

Die Verwendung von Live-Kacheln

Seite 687

Viele Windows-Store-Anwendungen besitzen nicht nur Kacheln, sondern sogar Live-Kacheln. Diese zeigen für den Benutzer relevante Informationen an und führen dazu, dass er die Anwendung häufiger benutzt. Hört sich an, als würde sich der Aufwand auch für Schrödinger lohnen, Live-Kacheln einzubauen.

| | | | |
|-------------------------------------|------------|-------------------------------------|------------|
| Innovation Live-Kacheln | 688 | Live-Kacheln mit C# erstellen | 693 |
| Klein, mittel, groß | 688 | Gona Catch'em all | 699 |
| Die Do's und Dont's | 690 | Deine Live-Zusammenfassung | 702 |
| Live-Tiles mit XML definieren | 691 | | |

Kapitel 26: Ich will alles rausholen

Datenzugriff über die Windows API

Seite 703

Prinzipiell weiß Schrödinger bereits, wie er auf Dateien zugreifen kann. Aber Universal Windows Platform Apps sind ja sehr abgeschottet, Sicherheit wird ganz großgeschrieben. Deshalb wird es dort bestimmt nicht ganz so einfach. Alles neu lernen für die UWP-Apps muss er aber nicht. Nur ein bisschen was dazulernen. Aber muss man das nicht sowieso immer?

| | | | |
|--|------------|--|------------|
| Dateizugriff nur mit Erlaubnis | 704 | Besser als Raumschiff Enterprise – ein Logbuch ... | 711 |
| Verhandlungstechnik 1: Dateiauswahl | 707 | Energie! Die Oberfläche der App | 712 |
| Verhandlungstechnik 2: Ordner auswählen | 708 | Der Sourcecode | 713 |
| Verhandlungstechnik 3: Anwendungsdaten speichern, ohne benutzergewählten Speicherort | 708 | Das ist doch alles dasselbe | 718 |
| | | Deine Kurzliste mit den wichtigsten Infos | 720 |

Kapitel 27: Funktioniert das wirklich?

Unit-Testing

Seite 721

Es ist immer wieder das Gleiche mit Fehlern. Kaum hat man einen gefunden, wurden zwei neue produziert, oder es tauchen Fehler an einer ganz anderen Stelle auf. Dann heißt es, alles noch einmal testen und immer wieder von vorn, langweilig und lästig. Wäre das nicht etwas für einen Computer? Könnte Schrödinger nicht ein Programm schreiben, das seine Programme testet?

| | | | |
|--|------------|--|------------|
| Das Problem: Testen kann lästig werden | 722 | Unit-Tests sind nicht alles | 729 |
| Die Lösung: Unit-Tests – Klassen, die Klassen testen | 723 | Testgetriebene Softwareentwicklung – oder wie du Autofahren lernst | 730 |
| Das Testprojekt erstellen | 726 | Darfst du schon fahren? | 731 |
| Die Ausführung ist das A und O! | 728 | Let's do it! | 736 |
| Spezielle Attribute | 729 | Dein Test-Merkzettel | 737 |

Kapitel 28: Auf ins Kaufhaus!

Das Publizieren im Windows Store

Seite 739

Das Wichtigste ist, dass die App auch in den Windows Store geladen wird. Denn ansonsten steht sie der Welt nicht zur Verfügung. Und jetzt ist es so weit, sich darum zu kümmern. Und doch gibt es auch hier ein paar Fallstricke, die Schrödinger berücksichtigen muss, damit die Annahme seiner App im Store nicht verweigert wird.

| | | | |
|--|------------|-----------------------------------|------------|
| Registriere einen Account, und es kann losgehen! | 740 | Die Zertifizierung startet! | 744 |
| So kommt dein Produkt in den Store | 742 | Auf ein Wiedersehen! | 748 |

| | |
|--------------------|------------|
| Index | 749 |
|--------------------|------------|