

Inhaltsverzeichnis

| | |
|--|----|
| Danksagung | 11 |
| Einführung | 13 |
| Systemanforderungen | 13 |
| Wo Sie am besten anfangen | 14 |
| Die Übungsdateien installieren und verwenden | 16 |
| Installation der Übungsdateien | 16 |
| Die Übungsdateien verwenden | 17 |
| Deinstallation der Übungsdateien | 19 |
| Konventionen und Elemente in diesem Buch | 20 |
| Konventionen | 20 |
| Andere Elemente | 21 |
| Korrekturen, Kommentare und Hilfe | 21 |
| Besuchen Sie die Website von Microsoft Press | 22 |
| Lektion 1 | |
| Wie Sie Ihr erstes objektorientiertes Programm schreiben | 23 |
| Bücher lesen: Ihr erstes objektorientiertes Programm | 24 |
| Die Klassenansicht benutzen | 40 |
| Das ReadBooks-Programm erstellen | 48 |
| Schnellüberblick | 51 |
| Lektion 2 | |
| Klasseninstanzen mit Konstruktoren erstellen | 55 |
| Konstruktoren | 55 |
| Bücher lesen: Eine alternative Implementierung | 56 |
| Konstruktoren in der Book-Klasse erstellen | 56 |
| Die Konstruktoren benutzen | 61 |
| Mit dem Konstruktor eine Instanz von Book erstellen | 62 |
| Ein Array mit Book-Instanzen generieren | 63 |
| Auf die Auswahl im ListBox-Steuerelement reagieren | 63 |
| Arrays deklarieren und initialisieren | 65 |
| Schnellüberblick | 68 |

| | | |
|-----------|---|-----|
| Lektion 3 | Felder und Eigenschaften erstellen | 71 |
| | Überlegungen zum Design der Eigenschaften | 72 |
| | Code-Analyse: Viele Eigenschaften | 73 |
| | Die SourceFile-Klasse entwerfen | 73 |
| | Die Benutzeroberfläche erstellen | 75 |
| | Die SourceFile-Klasse erstellen | 76 |
| | Der Benutzeroberfläche Code hinzufügen | 83 |
| | Das Programm testen | 87 |
| | Code-Analyse: Indexer und Standardeigenschaften verwenden | 88 |
| | Die AClass-Klasse erstellen | 88 |
| | Die Benutzeroberfläche erstellen | 94 |
| | Die DisplayClasses-Methode schreiben | 95 |
| | Die Browse-Schaltfläche programmieren | 96 |
| | Das Programm testen | 97 |
| | Schnellüberblick | 98 |
| Lektion 4 | Mit Methoden arbeiten | 101 |
| | Ein Kartenspiel | 101 |
| | Die Card-Klasse erstellen | 104 |
| | Die Hand-Klasse erstellen | 107 |
| | Die Hand-Klasse testen | 113 |
| | Die Deck-Klasse erstellen | 117 |
| | Die Testapplikation schreiben | 123 |
| | Schnellüberblick | 133 |
| Lektion 5 | Spezialisierte Klassen mithilfe der Vererbung erstellen | 137 |
| | Vererbung: ein Überblick | 137 |
| | BankAccount: Ein einfaches Beispiel | 139 |
| | Die BankAccount-Basisklasse erstellen | 139 |
| | Die abgeleitete SavingsAccount-Klasse erstellen | 143 |
| | Die abgeleitete CheckingAccount-Klasse erstellen | 152 |
| | Die abgeleiteten Klassen polymorph verwenden | 155 |
| | Die Benutzeroberfläche erstellen | 156 |
| | Die Konten erstellen | 157 |
| | Eine Transaktion durchführen | 158 |
| | Den Objekttyp ermitteln | 160 |
| | Von einem Steuerelement erben: Die RoundButton-Klasse | 162 |
| | Die RoundButton-Klasse erstellen | 162 |
| | Die Klasse verwenden | 164 |
| | Starten der Applikation | 165 |
| | Schnellüberblick | 167 |

| | | |
|------------------|--|------------|
| Lektion 6 | Basisklassen als abstrakte Klassen entwerfen | 171 |
| | Abstrakte Klassen | 171 |
| | Noch einmal die BankAccount-Klasse | 172 |
| | Das Design mit UML beschreiben | 173 |
| | Die abstrakte Klasse erstellen | 175 |
| | Die SavingsAccount-Klasse schreiben | 178 |
| | Die CheckingAccount-Klasse schreiben | 181 |
| | Die Klassen testen | 184 |
| | Eine typisierte Auflistungsklasse | 185 |
| | Die Library-Klasse neu gestalten | 186 |
| | Das Projekt erstellen | 188 |
| | Die Book-Klasse neu gestalten | 188 |
| | Die BookCollection-Klasse erstellen | 190 |
| | Die Klasse testen | 192 |
| | Schnellüberblick | 195 |
| Lektion 7 | Mit Ereignissen und Ausnahmen auf Veränderungen reagieren | 197 |
| | Feuer auf den Gleisen! Eine ereignisgesteuerte Applikation | 197 |
| | Die Track-Klasse implementieren | 200 |
| | Das CaughtOnFire-Ereignis erstellen | 205 |
| | Das Feuer auf dem Gleis anordnen | 210 |
| | Die Train-Klasse implementieren | 212 |
| | Die Benutzeroberfläche implementieren | 216 |
| | Das Programm testen | 219 |
| | Ereignismethoden ohne den Designer einrichten | 219 |
| | Ausnahmen – Wenn Probleme auftreten | 222 |
| | Eine Ausnahme generieren | 223 |
| | Eigene Ausnahmeklassen schreiben | 224 |
| | Schnellüberblick | 229 |
| Lektion 8 | Komponenten für Entwickler | 233 |
| | Das Memory-Spiel | 233 |
| | Das Spiel entwerfen | 235 |
| | Die Klassenbibliothek des Spiels | 236 |
| | Das GamesLibrary-Projekt erstellen | 237 |
| | Den Namespace ändern | 237 |
| | Das Card-Steuerelement erstellen | 238 |
| | Das Projekt für das Memory-Spiel erstellen | 246 |
| | Einen Verweis auf die Spielbibliothek hinzufügen | 247 |
| | Der Toolbox das Card-Steuerelement hinzufügen | 247 |
| | Die Deck-Komponente erstellen | 249 |

| | |
|--|------------|
| Das Steuerelement testen | 256 |
| Das Memory-Steuerelement erstellen | 258 |
| Die Spiel-Features des Spiels implementieren | 263 |
| Das Spiel implementieren | 264 |
| Die Projektmappe erstellen | 269 |
| Die Memory-Spielapplikation | 270 |
| Der Toolbox die LotsOfFun.Games-Steuerelemente hinzufügen | 270 |
| Die Benutzeroberfläche erstellen | 271 |
| Das Spiel programmieren | 271 |
| Schnellüberblick | 273 |
| Lektion 9 | |
| Dienste mithilfe von Schnittstellen zur Verfügung stellen | 275 |
| Die IMoveable-Schnittstelle | 275 |
| Die IMoveable-Schnittstelle definieren | 277 |
| Die IMoveable-Schnittstelle in der Pawn-Klasse implementieren | 278 |
| Die IMoveable-Schnittstelle testen | 282 |
| .NET Framework-Schnittstellen | 283 |
| Die IComparable-Schnittstelle implementieren | 285 |
| Die Schnittstelle testen | 288 |
| Die Schnittstellen IEnumerable und IEnumerator implementieren | 290 |
| Die Schnittstellen testen | 297 |
| Die IFormattable-Schnittstelle implementieren | 297 |
| Die IFormattable-Schnittstelle testen | 301 |
| Zwei weitere Verwendungsmöglichkeiten für Schnittstellen | 302 |
| Schnellüberblick | 304 |
| Lektion 10 | |
| Klassen mithilfe der Polymorphie austauschbar verwenden | 307 |
| Der Muster-Editor | 307 |
| Das Design des Muster-Editors | 308 |
| Die Muster- und Editor-Klassen entwerfen | 308 |
| Die Basisklassen | 311 |
| Die Pattern-Klasse erstellen | 312 |
| Die PatternEditor-Klasse erstellen | 313 |
| Die abgeleiteten Klassen | 314 |
| Die DrawnPattern-Klasse erstellen | 314 |
| Die DrawnPatternEditor-Klasse erstellen | 317 |
| Die BitmapPattern-Klasse erstellen | 321 |
| Die BitmapPatternEditor-Klasse erstellen | 323 |

| | |
|--|------------|
| Die Benutzeroberfläche | 327 |
| Die Elemente der Benutzeroberfläche erstellen | 327 |
| Die Vorlageninstanzen erstellen | 330 |
| Die neuen Muster bearbeiten und speichern | 331 |
| Die Applikation testen | 333 |
| Schnellüberblick | 334 |
| Lektion 11 | 337 |
| Gemeinsame und statische Member verwenden | 337 |
| Gemeinsam genutzte und statische Member | 337 |
| Ein etwas interessanterer Punkt | 338 |
| Die SortablePoint-Klasse erstellen | 339 |
| Die SortablePoint-Klasse testen | 343 |
| Die Applikation starten | 346 |
| Eine effizientere Karte | 347 |
| Die Card-Klasse implementieren | 347 |
| Die Card-Klasse testen | 352 |
| Das Singleton-Muster | 355 |
| Das Singleton-Muster implementieren | 355 |
| Die Singleton-Klasse testen | 357 |
| Überlegungen zum Design | 359 |
| Schnellüberblick | 361 |
| Lektion 12 | 363 |
| Operatoren mit Visual C# überladen | 363 |
| Eine kurze Lektion zum Thema Vektoren | 363 |
| Die Vektoralgebra-Applikation | 365 |
| Die Vector-Klasse implementieren | 366 |
| Die Vektoralgebra-Applikation implementieren | 370 |
| Die Logik hinzufügen | 374 |
| Die Applikation testen | 376 |
| Schnellüberblick | 377 |
| Lektion 13 | 379 |
| Instanzdaten speichern | 379 |
| Serialisierung | 379 |
| Die binäre Serialisierung implementieren | 381 |
| Die Benutzeroberfläche erstellen | 385 |
| Die Serialisierung definieren | 389 |
| Die Applikation starten und testen | 393 |
| Die XML-Serialisierung implementieren | 394 |
| Daten laden und speichern | 397 |
| Datasets | 397 |
| Ein nicht typisiertes Dataset implementieren | 399 |
| Ein typisiertes Dataset implementieren | 402 |
| Schnellüberblick | 408 |

| | | |
|-------------------|---|------------|
| Lektion 14 | Die Komplexität durch das Design verringern | 409 |
| | Die .NET-Entwurfsrichtlinien | 409 |
| | Objekte benennen | 410 |
| | Klassen-Member entwerfen | 413 |
| | Refactoring | 418 |
| | Das Refactoring zum Verschieben von Feldern in übergeordnete Klassen implementieren | 420 |
| | Entwurfsmuster | 424 |
| | Observer | 425 |
| | Schnellüberblick | 427 |
| Anhang A | Zusätzliche Informationsquellen | 429 |
| | Bücher | 429 |
| | Organisationen | 430 |
| | Stichwortverzeichnis | 431 |
| | Der Autor | 437 |