

Tartalomjegyzék

Köszönetnyilvánítás..... xxix

Bevezetés xxxi

Mi egy csapat vagyunk, az Olvasó és én.....	xxxii
A könyv áttekintése.....	xxxiii
Első kötet.....	xxxiii
1. rész: Bevezetés a C#-ba és a .NET platformba.....	xxxiii
1. fejezet: A .NET filozófiája.....	xxxiii
2. fejezet: C#-alkalmazások készítése.....	xxxiv
2. rész: A C# alapvető építőelemei.....	xxxiv
3. fejezet: A C# alapvető építőelemei, I. rész.....	xxxiv
4. fejezet: A C# alapvető építőelemei, II. rész.....	xxxiv
5. fejezet: Egységbe zárt osztálytípusok definiálása.....	xxxv
6. fejezet: A származtatás és a polimorfizmus.....	xxxv
7. fejezet: Strukturált hibakezelés.....	xxxv
8. fejezet: Az objektumok élelciklusa.....	xxxv
3. rész: Haladó programozási szerkezetek a C#-ban.....	xxxvi
9. fejezet: Interfészek használata.....	xxxvi
10. fejezet: Gyűjtemények és generikus típusok.....	xxxvi
11. fejezet: Metódusreferenciák, események és lambdák.....	xxxvi
12. fejezet: Indexerek, operátorok és mutatók.....	xxxvii
13. fejezet: C# 2008 nyelvi újdonságai.....	xxxvii
14. fejezet: Bevezetés a nyelvbe ágyazott lekérdezésekbe (LINQ).....	xxxvii
4. rész: Programozás .NET-szerelvényekkel.....	xxxvii
15. fejezet: A .NET-szerelvények.....	xxxviii
16. fejezet: Típusreflexió, késői kötés és attribútumalapú programozás.....	xxxviii
17. fejezet: Folyamatok, alkalmazástartományok és objektumkörnyezetek.....	xxxviii
18. fejezet: Többszálú alkalmazások készítése.....	xxxviii
19. fejezet: A köztes nyelv (CIL) és a dinamikus szerelvények.....	xxxix

Második kötet.....	xxxix
5. rész: Bevezetés a .NET alaposztály-könyvtáraiba	xxxix
20. fejezet: Fájlműveletek és elkülönített tárolás	xxxix
21. fejezet: Bevezetés az objektumsorosítás világába	xl
22. fejezet: ADO.NET 1. rész: Az élő kapcsolat	xl
23. fejezet: ADO.NET 2. rész: A bontott kapcsolat	xl
24. fejezet: A LINQ API programozása	xl
25. fejezet: A WCF	xli
26. fejezet: A WF	xli
6. rész: Felhasználói felületek.....	xli
27. fejezet: Windows Forms-programozás.....	xli
28. fejezet: A WPF és az XAML	xlii
29. fejezet: Programozás WPF-vezérlőelemekkel	xlii
30. fejezet: WPF 2D grafikus renderelés, erőforrások és témák.....	xlii
7. rész: Webes alkalmazások fejlesztése ASP.NET segítségével.....	xliii
31. fejezet: ASP.NET weboldalak készítése	xliii
32. fejezet: ASP.NET-vezérlőelemek, témák és mesteroldalak	xliii
33. fejezet: ASP.NET állapotkezelési technikák.....	xliii
8. rész: Függelék.....	xliv
A függelék: A COM és a .NET együttműködése	xliv
B függelék: Platformfüggetlen .NET-fejlesztés a Monóval.....	xliv
Öt szabadon letölthető fejezet – még több információ	xlv
A könyv forráskódjának igénylése.....	xlv
A lehetséges javítások	xlvi
Elérhetőségem.....	xlvi

1. rész: Bevezetés a C#-ba és a .NET platformba

1. A .NET filozófiája..... 3

Az előzmények áttekintése.....	4
Egy C/Win32 API programozó élete	4
Egy C++/MFC-programozó élete	4
Egy Visual Basic 6.0 programozó élete	5
Egy Java/J2EE-programozó élete	5
Egy COM-fejlesztő élete.....	6
Egy Windows DNA programozó élete.....	7

A .NET-megoldás	8
Bevezetés a .NET platform építőelemeibe (CLR, CTS és CLS).....	9
Az alaposztálykönyvtárak szerepe.....	10
A C# előnyei	11
További .NET-alapú programozási nyelvek.....	13
Élet egy többnyelvű világban.....	14
A .NET-szerelvények áttekintése	15
Egyfájlos és többfájlos szerelvények	17
A közös köztes nyelv (CIL) szerepe	18
A köztes nyelv előnyei.....	20
A köztes nyelv fordítása platformfüggetlen utasítássá	21
A .NET-típusmetaadat szerepe	22
A szerelvény manifesztumának szerepe	23
A közös típusrendszer (CTS)	23
Az egységes típusrendszer osztálytípusai.....	24
Az egységes típusrendszer interfésztípusai	25
Az egységes típusrendszer struktúrátípusai.....	25
Az egységes típusrendszer felsorolt típusai.....	26
Az egységes típusrendszer metódusreferencia-típusai	27
Az egységes típusrendszer típustagjai.....	27
Valódi CTS-adattípusok.....	28
A közös nyelvi specifikáció (CLS).....	29
A CLS-megfeleltetés biztosítása.....	31
A közös nyelvi futtatórendszer (CLR).....	32
A szerelvény, a névtér és a típus megkülönböztetése	33
A Microsoft-névterek szerepe	37
Programozott hozzáférés a névterekhez.....	38
Hivatkozás külső szerelvényekre	39
Szerelvények feltárása az ildasm.exe segítségével.....	40
A köztes nyelvi kód megtekintése.....	41
A típusmetaadat megtekintése.....	42
Szerelvény metaadatának (manifesztumának) megtekintése.....	42
Szerelvények feltárása a Lutz Roeder's Reflector segítségével	43
A .NET-futtatókörnyezet telepítése	44
A .NET platformfüggetlen természete.....	44
Összefoglalás.....	46

2. C#-alkalmazások fordítása	49
A .NET 3.5 keretrendszer fejlesztőkészletének szerepe	49
A Visual Studio 2008 parancssor	50
C#-alkalmazások fordítása csc.exe használatával.....	51
Bemeneti és kimeneti célok megadása	52
Hivatkozás külső szerelvényekre	54
Hivatkozás több külső szerelvényre	55
Több forrásfájl fordítása	55
A C#-reakciófájlok használata	56
.NET-alkalmazás készítése TextPad használatával	59
A C#-kulcsszószínezés bekapcsolása	59
A *.cs fájl szűrő konfigurálása	60
A csc.exe integrálása	61
Futtatási parancsok társítása menüelemekhez	62
.NET-alkalmazás készítése Notepad++ használatával.....	64
Az automatikus kiegészítési lista testreszabása	65
.NET-alkalmazás készítése SharpDevelop használatával.....	66
Egy egyszerű tesztprojekt létrehozása	67
.NET-alkalmazás készítése Visual C# 2008 Express használatával	69
A Visual C# Express néhány jellegzetessége	70
.NET-alkalmazás készítése Visual Studio 2008 használatával	71
A Visual Studio 2008 néhány jellegzetessége.....	72
A .NET-keretrendszer kiválasztása a New Project	
párbeszédablak segítségével	72
A Solution Explorer használata	73
Hivatkozás külső szerelvényekre	74
Projekt tulajdonságok megtekintése	74
A Class View.....	75
Az Object Browser	76
Integrált támogatás a kódújratervezéshez.....	76
Kódbővítések és a környezetbe foglalás technológiája.....	79
A Visual Class Designer.....	81
Object Test Bench.....	84
Az integrált .NET Framework 3.5 dokumentációs rendszer	85
További .NET-fejlesztőeszközök	87
Összefoglalás.....	88

2. rész: A C# alapvető építőelemei

3. A C# alapvető építőelemei, I. rész 91

Egy egyszerű C#-program anatómiája.....	91
Variációk Main() metódusra.....	93
Alkalmazás-hibakód megadása.....	94
A parancssori argumentumok feldolgozása.....	96
Parancssori paraméterek megadása Visual Studio 2008-ban.....	98
Érdekes kitérő: a System.Environment osztály néhány további tagja.....	99
A System.Console osztály.....	100
A Console osztály alapvető bemenete és kimenete.....	101
A konzolkimenet formázása.....	102
Numerikus adatok formázása.....	103
Numerikus adatok formázása a parancssori alkalmazásokon túl.....	105
A System-adattípusok és gyorsazonosítók a C#-ban.....	106
Változódeklarálás és inicializálás.....	108
A beépített adattípusok példányosítása.....	110
Az adattípusok hierarchiája.....	110
A numerikus adattípusok tagjai.....	112
A System.Boolean tagjai.....	113
A System.Char tagjai.....	114
Értékek értelmezése sztringadatokból.....	114
A System.String típus.....	115
Egyszerű sztringkezelés.....	116
Sztringösszefűzés.....	117
Vezérlőkarakterek.....	118
Szó szerinti sztringek definiálása.....	119
A sztringek és az egyenlőség.....	120
A sztring megváltoztathatatlan.....	121
A System.Text.StringBuilder típus.....	123
System.DateTime és System.TimeSpan.....	125
Szűkítő és bővítő adattípus-konverziók.....	126
Hibacsapda állítása a szűkítő adatkonverzióknak.....	128
Projekt szintű túlcsoportolás-ellenőrzés beállítása.....	131
Az unchecked kulcsszó.....	132
A System.Convert szerepe.....	132

Ciklus készítése a C#-ban.....	133
A for ciklus.....	133
A foreach ciklus.....	134
A while és a do/while ciklusszerkezetek.....	135
Elágazó szerkezetek és a reláció/egyenlőség-operátorok.....	136
Az if/else utasítás.....	136
A switch utasítás.....	138
Összefoglalás.....	139
4. A C# alapvető építőelemei, II. rész.....	141
Metódusok és paramétermódosítók.....	141
Az alapértelmezett paraméterátadási viselkedés.....	142
Az out módosító.....	144
A ref módosító.....	145
A params módosító.....	147
A tag túlterhelésének megértése.....	149
Tömbök kezelése a C#-ban.....	151
A C# tömbinicializáló szintaxisa.....	152
Objektumtömb definiálása.....	153
Munka a többdimenziós tömbökkel.....	154
Tömbök mint paraméterek (és visszatérési értékek).....	156
A System.Array osztály.....	156
Az Enum típus.....	159
Az enum alapjául szolgáló tároló vezérlése.....	160
Felsorolások deklarációja és használata.....	160
A System.Enum típus.....	162
Felsorolás név/érték párjainak dinamikus feltárása.....	162
A struktúrátípus.....	165
Struktúraváltozók létrehozása.....	167
Érték- és referenciatípusok.....	169
Érték- és referenciatípusok, valamint az értékadó operátor.....	170
Referenciatípusokat tartalmazó értéktípusok.....	172
Referenciatípusok átadása értéként.....	174
Referenciatípusok átadása referenciaként.....	176
Érték- és referenciatípusok: további részletek.....	177
Nullázható típusok a C#-ban.....	179
Munka a nullázható típusokkal.....	180
A ?? operátor.....	181
Összefoglalás.....	182

5. Beágyazott osztálytípusok definiálása	183
Bevezetés a C# osztálytípusaiba.....	183
Objektumok memóriaterületének lefoglalása	
a new kulcsszóval	186
A konstruktor	187
Az alapértelmezett konstruktor szerepe.....	188
Egyedi konstruktor definiálása.....	189
Az újradefiniált alapértelmezett konstruktor	190
A this kulcsszó szerepe.....	191
Konstruktorhívások láncolása a this kulcsszóval.....	193
A konstruktorok lefutásának sorrendje	196
A static kulcsszó	198
Statikus metódusok (és mezők) definiálása	199
Statikus adatok definiálása	200
Statikus konstruktorok definiálása.....	203
Statikus osztályok definiálása	205
Az objektumorientált programozás alappillérei	207
Az egységbe zárás.....	208
A származtatás	209
A polimorfizmus.....	210
Hozzáférés-módosítók a C#-ban.....	212
Az alapértelmezett hozzáférés-módosítók.....	213
Hozzáférés-módosítók és beágyazott típusok	214
Az első pillér: az egységbe zárás a C#-ban	215
Egységbe zárás tradicionális lekérdezők és	
módosítók használatával	216
Egységbe zárás típustulajdonságok használatával	218
A tulajdonságok belső reprezentációja	220
A get/set utasítástulajdonságok láthatósági	
szintjeinek vezérlése	222
Csak olvasható és csak írható tulajdonságok.....	223
Statikus tulajdonságok	224
Konstans adatok	226
Írásvédett mezők	227
Statikus írásvédett mezők.....	228
Részleges típusok	229

A C#-forráskód dokumentálása XML segítségével	231
Az XML-fájl generálása	234
XML-megjegyzések átalakítása az NDoc-on keresztül.....	235
Az elvégzett munka eredményének láthatóvá tétele.....	237
Összefoglalás.....	238
6. A származtatás és a polimorfizmus.....	239
A származtatás alapvető szerkezete	239
Az osztálytípus szülőosztálya	240
Több őosztály	242
A sealed kulcsszó	243
Osztálydiagramok készítése a Visual Studio 2008-ban	245
A második pillér: a származtatás	246
Őosztály konstruktorának meghívása a base kulcsszóval	249
Családi titok megtartása: a protected kulcsszó.....	251
Lezárt osztály hozzáadása	252
A tartalmazás és a delegálás	254
A beágyazott típus definiálása	255
A harmadik pillér: a polimorfizmus támogatása	258
A virtual és override kulcsszavak.....	259
Virtuális tagok felülbírálnása a Visual Studio 2008 segítségével.....	261
A virtuális tagok lezárása	262
Az absztrakt osztályok	263
Polimorfikus interfész építése	265
Tagok eltakarása	270
Az őosztály, a leszármazott osztály és a típusmódosítási szabályok	272
Az <i>as</i> kulcsszó.....	274
Az <i>is</i> kulcsszó	274
A fő szülőosztály: a System.Object.....	275
A System.Object.ToString() felüldefiniálása	279
A System.Object.Equals() felüldefiniálása	280
A System.Object.GetHashCode() felüldefiniálása	281
A módosított Person osztály tesztelése.....	282
A System.Object statikus tagjai	283
Összefoglalás.....	284

7. A strukturált kivételkezelés	285
Hibák, programhibák és kivételek	285
A kivételkezelés szerepe a .NET-ben.....	286
A .NET-kivételkezelés	288
A System.Exception őssztály.....	288
A lehető legegyszerűbb példa.....	290
Általános kivétel dobása	292
Kivételek elkapása	294
A kivétel tulajdonságainak beállítása	295
A TargetSite tulajdonság.....	295
A StackTrace tulajdonság.....	297
A HelpLink tulajdonság.....	297
A Data tulajdonság	298
Rendszerszintű kivételek (System.SystemException)	301
Alkalmazásszintű kivételek (System.ApplicationException).....	302
Egyedi kivétel készítése, 1. rész	302
Egyedi kivétel készítése, 2. rész	305
Egyedi kivétel készítése, 3. rész	306
Többszörös kivételek feldolgozása	308
Általános catch utasítások	310
Kivételek továbbdobása	311
Belső kivételek.....	311
A finally blokk	313
Ki mit dob?	314
A kezeletlen kivételek eredménye	315
Kezeletlen kivételek hibakeresése a Visual Studio használatával	315
Összefoglalás.....	317
8. Az objektumok élettartama.....	319
Osztályok, objektumok és referenciák.....	319
Az objektumok élettartama	321
A new kulcsszó köztes nyelve.....	322
Objektumreferenciák null értékre állítása	323
Az alkalmazás gyökérelemének szerepe.....	325
Az objektumgenerációk.....	327
A System.GC típus	328
A szemétygyűjtés kikényszerítése	330

Véglegesíthető objektumok készítése	333
A System.Object.Finalize() felüldefiniálása	334
A véglegesítő folyamat részletei	337
Eldobható objektumok készítése	337
A C# using kulcsszó egy újabb alkalmazása	340
Véglegesíthető és eldobható típusok készítése	342
A formalizált felszabadítási minta	343
Összefoglalás	346

3. rész: Haladó programozási szerkezetek a C#-ban

9. Interfészek használata 349

Az Interface típus	349
Az interfésztípusok és absztrakt őssosztályok összehasonlítása	351
Saját interfészek definiálása	354
Az interfész implementálása	356
Interfésztagok hívása az objektumon	358
Az interfészreferenciák megszerzése: az as kulcsszó	360
Az interfészreferenciák megszerzése: az is kulcsszó	360
Interfészek mint paraméterek	361
Interfészek mint visszatérési értékek	364
Interfésztípusok tömbje	364
Interfészimplementálás a Visual Studio 2008 segítségével	366
Névütközés feloldása explicit interfészimplementációval	367
Interfészhierarchia tervezése	371
Többszörös öröklés interfésztípusokkal	372
Felsorolható típusok készítése (IEnumerable és IEnumerator)	375
Iterátor metódus készítése a yield kulcsszóval	379
Nevesített iterátor létrehozása	380
Az iterátormetódusok belső reprezentációja	381
Klónozható objektumok készítése (ICloneable)	383
Egy összetettebb klónozási példa	385
Összehasonlítható objektumok készítése (Comparable)	389
Többszörös rendezési sorrend megadása (Comparable)	392
Egyedi tulajdonságok, egyedi rendezési típusok	394
A Callback interfész	395
Összefoglalás	399

10. Gyűjtemények és generikus típusok	401
A System.Collections névtér interfészei	401
Az ICollection interfész	403
Az IDictionary interfész	403
Az IDictionaryEnumerator interfész	404
Az IList interfész	404
A System.Collections névtér osztálytípusai	405
Az ArrayList típus használata	406
A Queue típus használata	408
A Stack típus használata	409
A System.Collections.Specialized névtér	410
A bedobozolás, kidobozolás és a System.Object kapcsolata	412
A dobozolóval járó probléma	415
A típusbiztonság és az erősen típusos gyűjtemények	416
Egyedi gyűjtemény készítése	417
Az erősen típusos gyűjtemények és a dobozolás	419
A System.Collections.Generic névtér	421
A List<T> típus vizsgálata	423
Saját generikus metódusok létrehozása	426
A típusparaméterek kikövetkeztetése	427
Generikus struktúrák és osztályok létrehozása	429
A default kulcsszó a generikus kódban	430
Saját generikus gyűjtemény készítése	432
A saját generikus gyűjtemény korlátai	433
Típusparaméter korlátozása a where kulcsszó használatával	435
Az operátorkorlátozások hiánya	438
Generikus ősosztály létrehozása	439
Generikus interfész létrehozása	440
Összefoglalás	442
11. Metódusreferenciák, események és lambdák	443
A.NET-metódusreferencia	443
A metódusreferencia definiálása C#-ban	445
A System.MulticastDelegate és System.Delegate alaposztályok	448
A legegyszerűbb metódusreferencia	449
A metódusreferencia-objektum vizsgálata	452

A Car típus utólagos módosítása metódusreferenciákkal	453
Többes küldés engedélyezése	457
Cél eltávolítása a metódusreferencia hívási listájáról	459
Egy összetettebb metódusreferencia	460
Metódusreferenciák mint paraméterek	461
Az átirányító kód elemzése	464
Metódusreferenciák kovarianciája	465
Generikus metódusreferencia készítése	468
A generikus metódusreferencia szimulációja generikus típusok nélkül	469
C#-események	471
Az event kulcsszó	472
Eseménykezelés a felszín alatt	474
A bejövő események figyelése	476
Esemény beregisztrálásának egyszerűsítése a Visual Studio 2008 használatával	477
Egy „korrekt” esemény	478
A generikus EventHandler<T> metódusreferencia	480
Névtelen metódusok a C# -ban	481
„Külső” változók elérése	483
A metóduscsoport átalakítása	484
A C# 2008 lambda operátora	486
A lambda kifejezések részletezése	489
Argumentumok feldolgozása több utasításon keresztül	490
A CarDelegate példa áttervezése lambda kifejezések használatával	492
Lambda kifejezések több paraméterrel (vagy paraméterek nélkül)	494
Összefoglalás	496

12. Indexelők, operátorok és mutatók 497

Az indexelő metódus	497
Objektumok indexelése sztringértékekkel	500
Az indexelő metódusok túlterhelése	501
Az indexelő metódusok belső ábrázolása	502
Többdimenziós indexelők	503
Indexelő definíciók interfésztípusokon	503

Az operátor-túlterhelés.....	504
Kétooperandusú operátorok túlterhelése.....	506
Mi a helyzet a += és -= operátorokkal?.....	508
Egyoperandusú operátorok túlterhelése.....	509
Az egyenlőségvizsgáló operátorok túlterhelése.....	510
Összehasonlító operátorok túlterhelése.....	511
A túlterhelt operátorok belső ábrázolása.....	512
Zárszó az operátor-túlterhelésről.....	514
Egyedi típuskonverziók.....	515
A numerikus konverzió.....	515
Konverzió az egymásból származó osztálytípusok között.....	516
Egyedi konverziós rutinok készítése.....	517
A Square típus további explicit konverziói.....	520
Implicit konverziós rutinok definiálása.....	521
Az egyedi konverziós rutinok belső ábrázolása.....	523
Mutatótípusok használata.....	524
Az unsafe kulcsszó.....	526
A * és az & operátor.....	528
A nem felügyelt (és felügyelt) cserefüggvény.....	528
Mezőelérés mutatókkal (a -> operátor).....	530
A stackalloc kulcsszó.....	530
Típuspéldány rögzítése a fixed kulcsszóval.....	531
A sizeof kulcsszó.....	532
A C# előfordítói direktívái.....	533
Kódrégiók megadása.....	534
Feltételes kód fordítás.....	535
Összefoglalás.....	537
13. A C# 2008 nyelv szolgáltatásai.....	539
Implicit típusú lokális változók.....	539
A var használata foreach szerkezetekben.....	541
Az implicit típusú változókra vonatkozó korlátozások.....	542
Implicit típusú lokális tömbök.....	543
Az implicit típusú adatok erősen tipizáltak.....	544
Az implicit típusú lokális változók előnye.....	545
Automatikus tulajdonságok.....	546
Interakció automatikus tulajdonságokkal.....	548
Az automatikus tulajdonságok elérésének korlátozása.....	549
Az automatikus tulajdonságok és az alapértelmezett értékek.....	549

Bővítő metódusok.....	551
A bővítő metódusok definiálása	552
A bővítő metódusok meghívása példányszinten	553
A bővítő metódusok statikus meghívása	555
A bővítő metódus hatóköre	555
A bővítő metódusokat definiáló típusok importálása	557
IntelliSense a bővítő metódusokhoz.....	558
Bővítő könyvtárak létrehozása és használata	559
Interfésztípusok kiterjesztése bővítő metódusokkal	560
Részleges metódusok	562
A részleges metódusok első pillantásra.....	563
A részleges metódusok használata.....	565
Az objektuminicializáló szintaxis.....	566
Egyedi konstruktor meghívása az inicializációs szintaxissal	568
A belső típusok inicializálása	569
Gyűjteményinicializálók	571
Névtelen típusok	572
A névtelen típusok belső ábrázolása	573
A ToString() és a GetHashCode() megvalósítása	575
Az egyenlőség szemantikája névtelen típusoknál.....	576
Névtelen típusokat tartalmazó névtelen típusok	578
Összefoglalás.....	579

14. A LINQ. Bevezetés 581

A LINQ szerepe	581
A LINQ-kifejezések jellemzői (típzáltság és kiterjeszthetőség)	583
Alapvető LINQ-szerelvények	583
Bevezetés a LINQ lekérdező kifejezésekbe	585
A LINQ és az implicit típusú lokális változók.....	588
A LINQ és a bővítő metódusok	590
A késleltetett futtatás szerepe.....	591
Az azonnali futtatás szerepe	592
A LINQ és a generikus gyűjtemények.....	593
A LINQ-kifejezés alkalmazása	594
A LINQ és a nem generikus gyűjtemények.....	595
Adatok szűrése az OfType<T>() használatával.....	596

A LINQ-lekérdezőoperátorok belső ábrázolása	597
Lekérdező kifejezések létrehozása lekérdezőoperátorok segítségével. Ismétlés	598
Lekérdező kifejezések létrehozása az Enumerable típus és a lambdák használatával.....	599
Lekérdező kifejezések létrehozása az Enumerable típus és a névtelen metódusok használatával	601
Lekérdező kifejezések létrehozása az Enumerable típus és a nyers metódusreferenciák használatával.....	602
A LINQ-lekérdezőoperátorok	603
Elemek számának meghatározása az Enumerable segítségével.....	604
Egyszerű tesztprojekt létrehozása	605
A kiválasztó szintaxis alapjai.....	606
Adatok egy részhalmazának kinyerése.....	608
Új adattípusok leképezése.....	609
Az eredményhalmaz-elemek sorrendjének megfordítása	610
Kifejezések rendezése	610
Különbségek keresése.....	611
LINQ-lekérdezők: szigetek a szigeteken?	612
Lekérdezőeredmények transzformálása Array típusúvá	614
Összefoglalás.....	615

4. rész: Programozás .NET-szerelvényekkel

15. A .NET-szerelvények 619

Egyedi névtér definiálása	619
A típus teljesen meghatározott neve	621
Definiálás álnevek használatával.....	623
Beágyazott névterek létrehozása	624
A Visual Studio 2008 „alapértelmezett névtere”	625
A .NET-szerelvények szerepe	626
A szerelvények a kód újrafelhasználását segítik	626
A szerelvények típushatárokat definiálnak.....	627
A szerelvények verzióval rendelkező egységek	627
A szerelvények önleírók	627
A szerelvények konfigurálhatók.....	628

A .NET-szerelvények felépítése.....	628
A Win32-fejléc	629
A CLR-fejléc.....	630
A köztes nyelvi kód, a típusmetaadat és a szerelvénymanifesztum	631
Opcionális szerelvény-erőforrások.....	631
Egyfájlos és többfájlos szerelvények	632
Egy állományból álló szerelvény készítése és használata.....	634
A manifesztum	638
A köztes nyelvi kód	639
A típusmetaadat.....	640
C#-ügyfélalkalmazás készítése	641
Visual Basic ügyfélalkalmazás készítése	643
A nyelvek közti származtatás	644
Több állományból álló szerelvény készítése és használata.....	645
Az ufo.netmodule fájl.....	647
Az airvehicles.dll fájl	647
A többfájlos szerelvény használata	648
Privát szerelvények.....	649
A privát szerelvény azonosítója.....	650
A keresési sorrend (probing) megadása	651
Privát szerelvények beállítása	651
Konfigurációs fájlok és a Visual Studio 2008.....	654
A .NET Framework konfiguráló segédprogramja	655
Megosztott szerelvények	657
Erős nevek megértése.....	658
A CarLibrary.dll ellátása erős névvel.....	660
Erős nevek hozzárendelése a Visual Studio 2008 segítségével.....	662
Megosztott szerelvények telepítése/eltávolítása a globális szerelvénytárba.....	662
A megosztott szerelvények felhasználása.....	664
A SharedCarLibClient manifesztuma	665
A megosztott szerelvények konfigurálása	666
Az aktuális megosztott szerelvény befagyasztása.....	667
A megosztott szerelvény 2.0.0.0 verziójának elkészítése.....	668
Dinamikus átirányítás a megosztott szerelvények meghatározott verzióiba	670
A .NET Framework konfiguráló programja – ismét	672

A GAC belső felépítése	673
Kibocsátó házirenddel rendelkező szerelvény	674
A kibocsátó házirend letiltása	676
A <codeBase> elem	677
A System.Configuration névtér	679
A gépi szintű konfigurációs fájl.....	680
Összefoglalás.....	681

16. Típusreflexió, késői kötés és attribútumalapú programozás..... 683

A metaadatok szükségessége.....	683
Az EngineState felsorolás (részleges) metaadata	685
A Car típus (részleges) metaadata.....	686
A TypeRef	687
A definiáló szerelvény leírása	687
A hivatkozott szerelvények leírása.....	688
Sztringliterálok leírása.....	688
A reflexió	689
A System.Type osztály	690
Típusreferencia megszerzése a System.Object.GetType() segítségével.....	691
Típusreferencia megszerzése a System.Type.GetType() segítségével.....	692
Típusreferencia készítése a typeof() segítségével	693
Egyedi metaadat-nézegető létrehozása	693
A metódusok reflexiója	693
Mezők és tulajdonságok reflexiója.....	694
A típus által megvalósított interfészek reflexiója	695
Egyéb érdekességek megjelenítése	695
A Main() metódus megvalósítása	695
Generikus típusok reflexiója.....	697
Metódusparaméterek és visszatérési értékek reflexiója	698
Dinamikusan betöltődő szerelvények	700
Megosztott szerelvények	702
A késői kötés	704
A System.Activator osztály	705
Paraméter nélküli metódusok hívása.....	706
Paraméterekkel rendelkező metódusok hívása	707

Programozás attribútumokkal.....	708
Attribútumok feldolgozása	709
Attribútumok alkalmazása a C# nyelvben.....	710
Attribútumok konstruktorparamétereinek meghatározása.....	712
Az Obsolete attribútum működés közben.....	713
C#-attribútum rövidített formája.....	713
Egyedi attribútum készítése.....	714
Egyedi attribútumok alkalmazása.....	715
Nevesített tulajdonság szintaxisa	716
Az attribútumok használatának korlátozása	716
Alkalmazásszintű (és modulszintű) attribútumok.....	718
A Visual Studio 2008 AssemblyInfo.cs fájlja	718
Attribútumok korai kötés használatával.....	720
Attribútumok késői kötés használatával.....	721
Reflexió, késői kötés és egyedi attribútumok kifejtése.....	723
Kiterjeszhető alkalmazás készítése	724
A CommonSnappableTypes.dll létrehozása.....	725
C# beépülő modul létrehozása	726
A Visual Basic beépülő modul készítése	726
Kiterjeszhető Windows Forms-alkalmazás készítése	727
Összefoglalás.....	731

17. Folyamatok, alkalmazástartományok és objektumkörnyezetek..... 733

A tradicionális Win32-es folyamat áttekintése	733
A szálak áttekintése	734
Folyamatok kezelése a .NET platformban	736
A futó folyamatok listázása.....	739
Egy adott folyamat vizsgálata.....	740
Egy folyamat szálkészletének vizsgálata.....	741
Egy folyamat modulkészletének vizsgálata.....	743
Folyamatok programozott indítása és leállítása	745
A .NET-alkalmazástartomány	746
Folyamat alkalmazástartományainak felsorolása	749
Új alkalmazástartományok programozott létrehozása.....	750
Alkalmazástartományok programozott eltávolítása	752
Az objektumkontextus határai.....	754
Kontextusváltó és kötött típusok.....	755

Kötött objektum meghatározása	756
Egy objektum kontextusának vizsgálata	757
A folyamat, alkalmazástartomány és a kontextus összefoglalása	759
Összefoglalás.....	759

18. Többszálú alkalmazások létrehozása 761

A folyamat, az alkalmazástartomány, a kontextus és a szál kapcsolata	761
A konkurencia problémája	763
A szálszinkronizáció szerepe	764
A .NET-metódusreferenciák rövid áttekintése	764
A metódusreferenciák aszinkron viselkedése	767
A BeginInvoke() és az EndInvoke() metódusok	767
A System.IAsyncResult interfész.....	768
Metódushívás aszinkron módon.....	769
A hívó szál szinkronizálása	770
Az AsyncCallback metódusreferencia szerepe.....	771
Az AsyncResult osztály szerepe	773
Egyedi állapotadatok átadása és fogadása	774
A System.Threading névtér.....	775
A System.Threading.Thread osztály.....	777
Hozzáférés az aktuális szál statisztikáihoz	778
A Name tulajdonság.....	779
A Priority tulajdonság	780
Második szál létrehozása kódból	781
A ThreadStart metódusreferencia használata	782
A ParameterizedThreadStart metódusreferencia használata	784
Előtérben és háttérben futó szálak.....	786
A konkurencia kérdése	787
Szinkronizáció a C# lock kulcsszavával	790
Szinkronizáció a System.Threading.Monitor típusal	792
Szinkronizáció a System.Threading.Interlocked típusal.....	793
Szinkronizáció a [Synchronization] attribútummal	795
Programozás időzített visszahívással	796
A CLR ThreadPool	798
A BackgroundWorker komponens	800
A BackgroundWorker típus használata.....	801
Adatfeldolgozás a BackgroundWorker típusal.....	803
Összefoglalás.....	805

19. A CIL és a dinamikus szerelvények szerepe 807

A CIL-programozás természete.....	807
CIL-direktívák, -attribútumok és -vezérlőkódok.....	808
A CIL-direktívák szerepe.....	809
A CIL-attribútumok szerepe	810
A CIL-vezérlőkódok szerepe.....	810
A CIL-vezérlőkódok/CIL-mnemoni megkülönböztetése.....	810
Verem írása/olvasása: A CIL veremalapú természete.....	811
A round-trip engineering	813
A CIL-kódcímkék szerepe	817
Fejlesztés a CIL-lel: egy *.il fájl módosítása.....	818
A CIL-kód fordítása ilasm.exe használatával	820
CIL-kód fordítása SharpDevelop használatával.....	821
A peverify.exe szerepe	823
CIL-direktívák és -attribútumok	823
Külsőleg hivatkozott szerelvények meghatározása CIL-ben.....	823
Az aktuális szerelvény definiálása CIL-ben.....	824
Névtér definiálása CIL-ben.....	825
Osztálytípusok definiálása CIL-ben.....	826
Interfészek definiálása és implementálása CIL-ben.....	827
Struktúrák definiálása CIL-ben.....	828
Felsorolt típusok definiálása CIL-ben	829
Generikus típusok definiálása a CIL-ben	829
A CILTypes.il fájl fordítása.....	830
.NET-alaposztálykönyvtár, C#- és CIL-adattípus-leképezések	832
Típustagok definiálása CIL-ben.....	833
Adatmezők definiálása CIL-ben	833
Típuskonstruktorok definiálása CIL-ben.....	834
Tulajdonságok definiálása CIL-ben.....	835
Tagparaméterek definiálása	835
A CIL-vezérlőkódok.....	836
A .maxstack direktíva.....	839
Lokális változók deklarálása CIL-ben.....	840
Paraméterek leképzése lokális változókba CIL-ben.....	841
A rejtett „this” referencia	842
Ciklusok reprezentációja a CIL-ben	843

.NET-szerelvény készítése köztes nyelven.....	844
A CILCars.dll létrehozása.....	844
A CILCarClient.exe létrehozása.....	847
Dinamikus szerelvények	850
A System.Reflection.Emit névtér	851
A System.Reflection.Emit.ILGenerator szerepe.....	852
Dinamikus szerelvény generálása	853
A szerelvény- és a modulkészlet generálása	856
A ModuleBuilder típus szerepe	857
A HelloClass típus és a sztringtípusú tagváltozó generálása	859
A konstruktorok generálása	859
A SayHello() metódus generálása	861
Dinamikusan létrehozott szerelvény használata.....	861
Összefoglalás.....	863
Tárgymutató	865
A szakmai lektorról	879
A szerzőről.....	881