

TARTALOM

ELŐSZÓ	xv
BEVEZETÉS	xvii
1. FEJEZET	
A C# ÁTTEKINTÉSE	1
1.1. A C# és a C++ közötti különbségek	1
1.1.1. C#	1
1.1.2. C++	2
1.1.3. CLR-szemétyűjtés	3
1.2. C# példaprogram	4
1.3. A C# 2.0 új eszközei – áttekintés	6
1.4. A C# 3.0 új eszközei – áttekintés	7
1.5. Összefoglalás	8
2. FEJEZET	
A C# ÉS A CLR	10
2.1. A JIT-fordító	10
2.2. Az assemblyk és az assemblytöltő	12
2.2.1. Az alkalmazás munkahalmazának minimalizálása	13
2.2.2. Az assemblyk elnevezése	14
2.2.3. Az assemblyk betöltése	14
2.3. Metaadatok	15
2.4. Nyelvfüggetlenség	17
2.5. Összefoglalás	17
3. FEJEZET	
A C# SZINTAXISA – ÁTTEKINTÉS	18
3.1. A C# erősen típusos nyelv	18
3.2. Kifejezések	19

3.3.	Utasítások és kifejezések.....	21
3.4.	Típusok és változók.....	21
3.4.1.	Értéktípusok.....	23
3.4.2.	Referenciatípusok.....	26
3.4.3.	Alapértelmezett változóinicializálás.....	27
3.4.4.	Implicit típusú lokális változók.....	28
3.4.5.	Típuskonverzió.....	31
3.4.6.	Az as és is operátorok.....	33
3.4.7.	Generikus típusok.....	35
3.5.	Névterek.....	37
3.5.1.	Névterek definiálása.....	37
3.5.2.	Névterek használata.....	38
3.6.	Vezérlőszerkezetek.....	40
3.6.1.	if-else, while, do-while és for.....	40
3.6.2.	switch.....	40
3.6.3.	foreach.....	41
3.6.4.	break, continue, goto return és throw.....	42
3.7.	Összefoglalás.....	42

4. FEJEZET

OSZTÁLYOK, STRUKTÚRÁK ÉS OBJEKTUMOK.....	43	
4.1.	Osztálydefiniációk.....	45
4.1.1.	Mezők.....	46
4.1.2.	Konstruktorok.....	50
4.1.3.	Metódusok.....	51
4.1.4.	Tulajdonságok.....	52
4.1.5.	Bezárás.....	58
4.1.6.	Hozzáférhetőség.....	62
4.1.7.	Interfészek.....	64
4.1.8.	Öröklődés.....	66
4.1.9.	Zárt osztályok.....	74
4.1.10.	Absztrakt osztályok.....	74
4.1.11.	Beágyazott osztályok.....	76
4.1.12.	Indexelők.....	79
4.1.13.	Részleges osztályok.....	81
4.1.14.	Részleges metódusok.....	82
4.1.15.	Statikus osztályok.....	84
4.1.16.	Foglalt tagnevek.....	86
4.2.	Értéktípus-definiciók.....	87
4.2.1.	Konstruktorok.....	88
4.2.2.	A this kulcsszó jelentése.....	90
4.2.3.	Véglegesítők.....	93
4.2.4.	Interfészek.....	94
4.3.	Anonim típusok.....	94

4.4.	Objektuminitializálók	98
4.5.	Becsomagolás és kicsomagolás.....	101
4.5.1.	Mikor kerül sor csomagolásra?	105
4.5.2.	Hatékonyág és zűrzavar	107
4.6.	System.Object	108
4.6.1.	Egyenlőség és annak értelmezése.....	110
4.6.2.	Az IComparable interfész.....	110
4.7.	Objektumok létrehozása	111
4.7.1.	A new kulcsszó	111
4.7.2.	Mezőinicializálás.....	112
4.7.3.	Statikus (osztály) konstruktorok.....	114
4.7.4.	Példánykonstruktorok és meghívásuk sorrendje	117
4.8.	Objektumok megsemmisítése	121
4.8.1.	Véglegesítők	122
4.8.2.	Determinisztikus megsemmisítés	123
4.8.3.	Kivételkezelés	124
4.9.	Felszabadítható objektumok.....	125
4.9.1.	Az IDisposable interfész	125
4.9.2.	A using kulcsszó	127
4.10.	A metódusparaméterek típusai	129
4.10.1.	Értékargumentumok	130
4.10.2.	ref argumentumok	130
4.10.3.	out paraméterek	132
4.10.4.	Paramétertömbök.....	132
4.11.	Metódusok túlterhelése	133
4.12.	Öröklődés és virtuális metódusok	134
4.12.1.	Virtuális és absztrakt metódusok.....	134
4.12.2.	Az override és a new használata metódusoknál	135
4.12.3.	Zárt metódusok.....	137
4.12.4.	Még néhány szó a C# virtuális metódusairól	138
4.13.	Öröklődés, tartalmazás és delegálás.....	138
4.13.1.	Interfészek vagy osztályöröklődés?.....	138
4.13.2.	Delegálás és kompozíció versus öröklődés	140
4.14.	Összefoglalás.....	142
5. FEJEZET		
INTERFÉSZEK ÉS SZERZŐDÉSEK		143
5.1.	Az interfészek definiálják a típusokat	144
5.2.	Interfészek definiálása	145
5.2.1.	Mit tartalmazhat egy interfész?	146
5.2.2.	Interfészek származtatása és tagok elrejtése	147
5.3.	Interfészek implementálása	149
5.3.1.	Implicit interfészimplementáció.....	149
5.3.2.	Explicit interfészimplementáció.....	150

5.3.3.	Interfészimplementáció felülírása származtatott osztályokban	152
5.3.4.	Óvatosan járjunk el, ha értéktípusokban implementálunk interfészeket.....	156
5.4.	Interfésztagok kiválasztási szabályai	157
5.5.	Explicit interfészimplementáció értéktípusokban	161
5.6.	Metódusaink ellátása verziószámmal	164
5.7.	Szerződések	165
5.7.1.	Szerződések implementációja osztályokban	165
5.7.2.	Interfészszerződések.....	167
5.8.	Interfészek vagy osztályok	168
5.9.	Összefoglalás.....	172

6. FEJEZET

OPERÁTOROK TÚLTERHELÉSE..... 173

6.1.	Attól még, hogy lehet, nem biztos, hogy célszerű is.....	173
6.2.	A túlterhelt operátorok típusa és formátuma	174
6.3.	Ne módosítsuk az operátor operandusait!	175
6.4.	Számít-e a paraméterek sorrendje?	176
6.5.	Az összeadás operátorának túlterhelése	176
6.6.	Túlterhelhető operátorok	178
6.6.1.	Összehasonlító operátorok	179
6.6.2.	Konverziós operátorok	182
6.6.3.	Logikai operátorok	184
6.7.	Összefoglalás.....	187

7. FEJEZET

KIVÉTELKEZELÉS ÉS KIVÉTELBIZTOSSÁG 188

7.1.	Kivételkezelés a CLR-ben.....	188
7.2.	Kivételkezelő módszerek a C#-ban.....	189
7.2.1.	Kivételdobás.....	189
7.2.2.	A kezeletlen kivételekkel kapcsolatos változások a .NET keretrendszerben.....	189
7.2.3.	A try utasítás szintaxisának áttekintése	191
7.2.4.	Kivételek újradobása és átadása	193
7.2.5.	A finally blokkban dobott kivételek	196
7.2.6.	A véglegesítőben dobott kivételek	196
7.2.7.	Statikus konstruktorokban dobott kivételek.....	198
7.3.	Hol célszerű kezelnünk a kivételeket?	199
7.4.	Ne használjunk kivételeket a vezérlés irányításához!	199
7.5.	A kivételsemlegesség megvalósítása.....	200
7.5.1.	A kivételsemleges kód alapvető szerkezete.....	201
7.5.2.	Korlátozott végrehajtási tartományok	207
7.5.3.	Kritikus véglegesítők és a SafeHandle	209

7.6.	Egyéni kivételt megvalósító osztályok létrehozása.....	214
7.7.	Lefoglalt erőforrások és kivételek használata	216
7.8.	Visszavonási lehetőség biztosítása	220
7.9.	Összefoglalás.....	223

8. FEJEZET

SZTRINGEK KEZELÉSE	224	
8.1.	A sztringek áttekintése	224
8.2.	Sztringliterálok	225
8.3.	Formátumspecifikáló szimbólumok és a globalizáció	227
8.3.1.	Az Object.ToString, az IFormattable és a CultureInfo.....	227
8.3.2.	Egyéni CultureInfo típusok létrehozása és bejegyzése	229
8.3.3.	Formátumsztringek.....	231
8.3.4.	A Console.WriteLine és a String.Format	232
8.3.5.	Sztringformázás egyéni típusokban – példák	233
8.3.6.	Az ICustomFormatter	235
8.3.7.	Sztringek összehasonlítása	237
8.4.	Külső forrásokból származó sztringek kezelése.....	239
8.5.	A StringBuilder	241
8.6.	Keresés sztringekben reguláris kifejezésekkel.....	243
8.6.1.	Keresés reguláris kifejezésekkel	243
8.6.2.	Keresés és csoportosítás	245
8.6.3.	Részszttring helyettesítése a Regex segítségével	249
8.6.4.	Reguláris kifejezések létrehozása.....	251
8.7.	Összefoglalás.....	253

9. FEJEZET

TÖMBÖK, KOLLEKCIÓTÍPUSOK ÉS BEJÁRÓK	254	
9.1.	Tömbök – bevezetés.....	254
9.1.1.	Implicit típusú tömbök	255
9.1.2.	Típuskonvertálhatóság és kovariancia.....	258
9.1.3.	Rendezhetőség és kereshetőség.....	259
9.1.4.	Szinkronizálás	259
9.1.5.	Vektorok és tömbök.....	260
9.2.	Téglalap alakú többdimenziós tömbök	262
9.3.	Egyenetlen többdimenziós tömbök	264
9.4.	Kollekciótípusok	266
9.4.1.	Az ICollection<T> és az ICollection összehasonlítása	266
9.4.2.	Kollekciók szinkronizálása.....	268
9.4.3.	Listák	269
9.4.4.	Szótárak	270
9.4.5.	Halmazok.....	271

9.4.6.	A System.Collections.ObjectModel	271
9.4.7.	Hatékonyág	274
9.5.	Az IEnumerable<T>, az IEnumerator<T>, az IEnumerable és az IEnumerator	275
9.5.1.	Kollekciókat előállító típusok	279
9.6.	Bejárók	280
9.6.1.	Előrehaladó, hátrahaladó és kétirányú bejárók	285
9.7.	Kollekcióinicializálók	290
9.8.	Összefoglalás	291

10. FEJEZET

DELEGÁLTAK, ANONIM FÜGGVÉNYEK ÉS ESEMÉNYEK 292

10.1.	A delegáltak áttekintése	292
10.2.	Delegáltak létrehozása és használata	293
10.2.1.	Egyszerű delegált	294
10.2.2.	Delegátláncok	295
10.2.3.	Delegátláncok elemeinek feldolgozása	298
10.2.4.	Kötetlen (nyílt példányú) delegáltak	299
10.3.	Események	302
10.4.	Anonim metódusok	307
10.4.1.	Az elfogott változókkal kapcsolatos problémák	312
10.4.2.	Anonim metódusok és a paraméterkötés delegáltakban	314
10.5.	A Stratégia minta	318
10.6.	Összefoglalás	320

11. FEJEZET

GENERIKUS TÍPUSOK 321

11.1.	A generikus típusok és a C++ sablonjai közötti különbségek	322
11.2.	Generikus típusok – hatékonyság és típusbiztonság	323
11.3.	Generikus típusok definíciói és az összetett típusok	326
11.3.1.	Generikus osztályok és struktúrák	326
11.3.2.	Generikus interfészek	329
11.3.3.	Generikus metódusok	330
11.3.4.	Generikus delegáltak	332
11.3.5.	Generikus típuskonverzió	336
11.3.6.	Alapértelmezett érték kifejezések	336
11.3.7.	Nullkezelő típusok	338
11.3.8.	Az összetett típusok és a hozzáférhetőség	340
11.3.9.	A generikus típusok és az öröklődés	341
11.4.	Korlátozások	342
11.4.1.	Korlátozások osztálytípustól eltérő típusokban	348
11.5.	A rendszer generikus kollekciói	349
11.6.	A rendszer generikus interfészei	350

11.7. Válogatott problémák és megoldásaik	352
11.7.1. Konverzió és operátorok a generikus típusokban.....	352
11.7.2. Összetett típusok dinamikus létrehozása	363
11.8. Összefoglalás.....	364
12. FEJEZET	
SZÁLKEZELÉS A C#-BAN	366
12.1. Szálkezelés a C#-ban és a .NET-ben	366
12.1.1. Szálak elindítása	367
12.1.2. Az IOU-minta és az aszinkron metódushívások	370
12.1.3. Szálállapotok	370
12.1.4. Szálak befejezése.....	373
12.1.5. Szálak megállítása és szüneteltetett szálak újraindítása	375
12.1.6. Szálvég megvárása	376
12.1.7. Előtér- és háttérszálak	376
12.1.8. Szálak lokális tárolása	377
12.1.9. A kezeletlen szálak és a COM-egységek szerepe.....	381
12.2. Szálak szinkronizálása.....	383
12.2.1. Egyszerű szinkronizálás az Interlocked osztállyal	384
12.2.2. A Monitor osztály	390
12.2.3. Zárolóobjektumok	399
12.2.4. A semafor.....	405
12.2.5. Események.....	405
12.2.6. A Win32 szinkronizáló objektumai és a WaitHandle	406
12.3. A ThreadPool használata	409
12.3.1. Aszinkron metódushívások	410
12.3.2. Időzítők.....	418
12.4. Összefoglalás.....	420
13. FEJEZET	
A C# KANONIKUS ALAKJAINAK VIZSGÁLATA.....	421
13.1. Referenciatípusok kanonikus alakja.....	422
13.1.1. Használjunk zárt osztályokat	422
13.1.2. A nem virtuális interfész (NVI) minta használata.....	424
13.1.3. Klónozható-e az objektum?	427
13.1.4. Felszabadítható-e az objektum?	433
13.1.5. Szükség van-e az objektumban véglegesítőre?.....	436
13.1.6. Mit jelent az objektum szempontjából az egyenlőség?.....	444
13.1.7. Az Equals felülírásakor írjuk felül a GetHashCode-ot is.....	452
13.1.8. Rendezhető-e az objektum?	456
13.1.9. Formázható-e az objektum?	458
13.1.10. Konvertálható-e az objektum?	462
13.1.11. Mindig törekedjünk a típusbiztonságra	465
13.1.12. Megváltoztathatatlan referenciatípusok használata.....	469

13.2. Értéktípusok kanonikus alakja	473
13.2.1. A teljesítmény javítása érdekében írjuk felül az Equals-t	474
13.2.2. Támogat-e az értéktípus interfészeket?	478
13.2.3. Implementáljuk az interfésztagok és származtatott metódusok típusbiztos változatait	479
13.3. Összefoglalás.....	482
13.3.1. Ellenőrző lista referenciatípusokhoz	482
13.3.2. Ellenőrző lista értéktípusokhoz	484
14. FEJEZET	
BŐVÍTŐ METÓDUSOK	486
14.1. Bővítő metódusok – bevezetés	486
14.1.1. Hogyan azonosítja a fordító a bővítő metódusokat?	487
14.1.2. Mi történik a háttérben?	490
14.1.3. A kód olvashatósága és érthetősége	491
14.2. Alkalmazási javaslatok.....	492
14.2.1. Használjunk bővítő metódusokat származtatás helyett	493
14.2.2. Különítsük el a bővítő metódusokat egy önálló névtérbe	494
14.2.3. A típus szerződésének módosítása elronthatja a bővítő metódusok működését.....	495
14.3. Átalakítók	496
14.4. Műveletláncok.....	500
14.5. Egyéni bejárók.....	502
14.5.1. A funkcionális programozás elemei	503
14.6. A Látogató minta.....	510
14.7. Összefoglalás.....	513
15. FEJEZET	
LAMBDA-KIFEJEZÉSEK	514
15.1. Lambda-kifejezések – bevezetés.....	514
15.1.1. Lambda-kifejezések.....	515
15.1.2. Lambda-utasítások.....	521
15.2. Kifejezésfák.....	521
15.2.1. Műveletek kifejezésekkel	524
15.2.2. A függvény mint adat	525
15.3. Lambda-kifejezések használata alkalmazásokban	526
15.3.1. Bejárók és generátorok – ismétlés.....	526
15.3.2. Lezárások (változóelfogás) és a memoizáció.....	530
15.3.3. A Currying-módszer.....	535
15.3.4. Anonim rekurzió.....	537
15.4. Összefoglalás.....	539

16. FEJEZET

A LINQ	540
16.1. Az adatkezelés megvalósítása	541
16.1.1. Lekérdező kifejezések	541
16.1.2. Bővítő metódusok és lambda-kifejezések – ismétlés	543
16.2. Szabványos lekérdező operátorok	544
16.3. A C# lekérdező kulcsszavai.....	546
16.3.1. A from záradék és a tartományváltozók.....	546
16.3.2. A join záradék.....	548
16.3.3. A where záradék és a szűrők	550
16.3.4. Az orderby záradék	550
16.3.5. A select záradék és a projekció.....	552
16.3.6. A let záradék.....	553
16.3.7. A group záradék.....	555
16.3.8. Az into záradék és a folytatások.....	558
16.4. A lustaság előnyei.....	559
16.4.1. A C# bejárói és a lustaság.....	560
16.4.2. A lusta kiértékelés meghiúsulása.....	561
16.4.3. Lekérdezések azonnali végrehajtása	563
16.4.4. Kifejezésfák – ismétlés.....	564
16.5. A funkcionális programozás technikái	564
16.5.1. Egyéni szabványos lekérdező operátorok és a lusta kiértékelés	565
16.5.2. A foreach utasítás helyettesítése.....	573
16.6. Összefoglalás.....	575
IRODALOMJEGYZÉK	577
TÁRGYMUTATÓ	579