

# Összefoglaló tartalomjegyzék

---

Bevezetés.....	1
<b>1. rész: ISMERKEDJÜNK MEG A C++ PROGRAMOZÁSSAL .....</b>	<b>7</b>
1. fejezet: Írjuk meg az első C++ programodat.....	9
2. fejezet: Deklaráljunk változókat konstans értékkel.....	35
3. fejezet: Végezzünk matematikai műveleteket.....	51
4. fejezet: Végezzünk logikai műveleteket.....	59
5. fejezet: Vezéreljük a programfolyamatot .....	73
<b>2. rész: VÁLJUNK FUNKCIONÁLIS C++ PROGRAMOZÓVÁ.....</b>	<b>91</b>
6. fejezet: Hozzunk létre függvényeket.....	93
7. fejezet: Tároljunk sorozatokat tömbökben.....	111
8. fejezet: Vessünk egy első pillantást a C++ mutatóira.....	129
9. fejezet: Vessünk egy második pillantást a C++ mutatóira.....	145
10. fejezet: A C++ előfeldolgozója .....	163
<b>3. rész: ISMERKEDJÜNK MEG AZ OSZTÁLYOKKAL .....</b>	<b>177</b>
11. fejezet: Ismerkedjünk meg az objektumorientált programozással.....	179
12. fejezet: Léptessük magasabb osztályba a C++-t.....	185
13. fejezet: Mutassunk rá objektumokra, és bámuljuk őket.....	201
14. fejezet: Védjük meg a tagokat: Ne zavarjanak! .....	219
15. fejezet: „Ó, miért építesz fel, csak hogy rögtön le is bonts?” .....	227
16. fejezet: Használjunk konstruktorokat argumentumokkal.....	239

17. fejezet: A másoló/áthelyező konstruktor .....	263
18. fejezet: Statikus tagok: Talán egy kis olajozás segít?.....	277
<b>4. RÉSZ: ÖRÖKLŐDÉS .....</b>	<b>287</b>
19. fejezet: Örököljünk egy osztályt .....	289
20. fejezet: Ismerkedjünk meg a virtuális tagfüggvényekkel: ezek tényleg valódiak? .....	299
21. fejezet: Bontsunk újra osztályokat.....	309
<b>5. rész: BIZTONSÁG.....</b>	<b>319</b>
22. fejezet: Egy új értékadó operátor, ha elfogadod .....	321
23. fejezet: Használjunk adatfolyam-bemenetet és -kimenetet.....	335
24. fejezet: Kezeljünk hibákat és kivételeket.....	357
25. fejezet: Örököljünk többszörös öröklődéssel .....	367
26. fejezet: Próbáljuk ki a C++ sablonokat .....	379
27. fejezet: Szabványosítsunk a szabványos sablonkönyvtárral .....	389
28. fejezet: Írjunk olyan kódot, amit nem lehet meghekkelni .....	401
<b>6. rész: TOP 10.....</b>	<b>427</b>
29. fejezet: Tíz módszer, amellyel elkerülheted a hibákat a programodban.....	429
30. fejezet: Tíz módszer, amellyel megvédheted a programjaidat a hekkerektől .....	437
Tárgymutató .....	453

# Tartalomjegyzék

A szerzőről.....	v
Ajánlás.....	v
A szerző köszönetnyilvánítása.....	v
<b>Összefoglaló tartalomjegyzék.....</b>	<b>vii</b>
<b>Karakterkódolási bonyodalmak .....</b>	<b>xvii</b>
<b>Bevezetés.....</b>	<b>1</b>
A könyvről .....	1
A könyvben használt ikonok .....	4
A könyvön túl.....	4
Hol is kezdjük? .....	5
<b>1. rész: ISMERKEDJÜNK MEG A C++ PROGRAMOZÁSSAL .....</b>	<b>7</b>
<b>1. fejezet: Írjuk meg az első C++ programodat.....</b>	<b>9</b>
Értsük meg a C++ fogalmait.....	10
Telepítsük a Code::Blocks-ot .....	11
Windows.....	11
Ubuntu Linux .....	14
Macintosh .....	16
Írjuk meg az első C++ programodat.....	20
Hozzunk létre egy projektet.....	20
Írjuk be a C++ kódot.....	22
Csaljunk.....	24
Buildeljük a programodat .....	25
Futtassuk a programodat.....	27
Tekintsük át az annotált programot .....	28
Vizsgáljuk meg az összes C++ program szerkezetét.....	28
Adjunk magyarázatot a forráskódhoz megjegyzésekkel.....	29
Alapozzuk a programokat C++ utasításokra.....	30
Írjunk deklarációkat.....	30
Generáljunk kimenetet .....	31
Számítsunk ki kifejezéseket .....	32
Tároljuk a kifejezések eredményeit .....	32
Vizsgáljuk meg a Konverzió fennmaradó részét .....	32
<b>2. fejezet: Deklaráljunk változókat konstans értékkel.....</b>	<b>35</b>
Deklaráljunk változókat .....	36
Deklaráljunk különböző típusú változókat .....	36
Tekintsük át az egész számok korlátozásait a C++-ban.....	37
Oldjuk meg a csonkolás problémáját .....	38

Tekintsük át a lebegőpontos számok korlátait .....	39
Deklaráljunk változótípusokat .....	41
A konstansok típusai .....	43
A numerikus típusok értéktartományai .....	44
Speciális karakterek.....	45
Széles rakományok a karakterek országútján.....	46
Tényleg logikusak ezek a számítások? .....	47
Kevert típusú kifejezések.....	47
Automatikus deklarációk.....	48
<b>3. fejezet: Végezzünk matematikai műveleteket.....</b>	<b>51</b>
Végezzünk el egyszerű kétoperandusú aritmetikai műveleteket.....	52
Bontsunk fel kifejezéseket .....	53
Határozzuk meg a műveletek sorrendjét .....	53
Végezzünk egyoperandusú műveleteket.....	54
Használjunk értékadó operátorokat.....	56
<b>4. fejezet: Végezzünk logikai műveleteket.....</b>	<b>59</b>
Miért érdemes a logikai műveletekkel vesződni? .....	60
Használjuk az egyszerű logikai operátorokat.....	60
Tároljunk logikai értékeket .....	61
Használjunk logikai int változókat .....	63
Óvatosan a logikai műveletekkel lebegőpontos változókon! .....	63
Fejezzünk ki bináris számokat .....	65
A tízes számrendszer.....	65
Más számrendszerek .....	66
A kettes számrendszer .....	66
Végezzünk bitenkénti logikai műveleteket .....	68
Az egybites operátorok .....	69
Használjuk a bitenkénti operátorokat .....	70
Egy egyszerű teszt .....	70
<b>5. fejezet: Vezéreljük a programfolyamatot .....</b>	<b>73</b>
Vezéreljük a programfolyamatot elágaztató utasításokkal.....	74
Futtassunk ciklusokat egy programban.....	76
Futtassunk egy ciklust, amíg egy feltétel igaz .....	76
Használjuk az automatikus növelési/csökkentési funkciót .....	78
Használjuk a for ciklust .....	80
Kerüljük el a rettegett végtelen ciklust .....	83
A for each ciklus.....	83
Alkalmazzunk speciális ciklusvezérlőket.....	85
Ágyazzunk egymásba vezérlési utasításokat .....	87
Kapcsoljunk át másik csatornára?.....	89
<b>2. rész: VÁLJUNK FUNKCIONÁLIS C++ PROGRAMOZÓVÁ.....</b>	<b>91</b>
<b>6. fejezet: Hozzunk létre függvényeket .....</b>	<b>93</b>
Írjunk meg és használjunk egy függvényt .....	94
Definiáljuk az első függvényünket.....	96

Definiáljuk a sorozatOsszegese() függvényt.....	96
Hívjuk meg a sorozatOsszegese() függvényt.....	97
Oszd meg és uralkodj .....	97
Ismerkedjünk meg a függvények részleteivel.....	98
Ismerkedjünk meg az egyszerű függvényekkel .....	98
Ismerkedjünk meg az argumentummal rendelkező függvényekkel .....	99
Terheljünk túl függvényneveket.....	102
Definiáljunk függvényprototípusokat.....	104
Állítsunk argumentumokat alapértelmezett értékre.....	105
Használjunk érték szerinti átadást hivatkozás szerinti helyett .....	107
Változótárolási típusok .....	108

## **7. fejezet: Tároljunk sorozatokat tömbökben..... 111**

Vegyük számba a tömbök argumentumait .....	111
Használjunk tömböket.....	113
Inicializáljunk egy tömböt.....	117
Nyúljunk túl messzire egy tömbben .....	117
Használjunk tartományalapú for ciklusokat tömbökhöz.....	118
Definiáljunk és használjunk tömbökből álló tömböket .....	119
Használjunk karaktertömböket.....	119
Hozzunk létre egy karaktertömböt .....	119
Hozzunk létre egy karaktersorozatot, azaz sztringet .....	121
Kezeljük sztringeket karakteresen .....	122
Adjunk hozzá néhány könyvtári függvényt .....	125
Csináljunk helyet több-bájtos sztringeknek.....	126

## **8. fejezet: Vessünk egy első pillantást a C++ mutatóira..... 129**

Változóméret .....	129
Mi van megadva egy címben? .....	131
Címoperátorok .....	131
Használjunk mutatóváltozókat .....	133
Használjunk különböző típusú mutatókat .....	134
Adjunk át mutatókat függvényeknek.....	135
Használjunk érték szerinti átadást .....	135
Adjunk át mutatóértékeket.....	136
Használjunk hivatkozás szerinti átadást.....	136
Konstans bosszúság a const kulcsszóval .....	137
Használjuk a halomnak nevezett memóriablokkot.....	139
Korlátozott hatáskör .....	139
Vizsgáljuk meg a hatáskör problémáját .....	141
Adjunk meg egy megoldást a halom használatával .....	141

## **9. fejezet: Vessünk egy második pillantást a C++ mutatóira..... 145**

Definiáljuk a mutatóváltozókön végzett műveleteket.....	145
Vizsgáljuk meg újra a tömböket a mutatóváltozók ismeretében	146
Használjunk operátorokat egy tömb címén.....	148
Terjesszük ki a mutatóműveleteket a sztringekre .....	149
Adjuk meg a sztringek mutatóalapú kezelésének indokait.....	151
Használjunk operátorokat a chartól eltérő mutatótípusokhoz...	152

Vessünk össze egy mutatót és egy tömböt.....	152
Mikor hamis értékű egy mutató? .....	154
Deklaráljunk és használjunk mutatókból álló tömböket .....	155
Használjunk karaktersztringekből álló tömböket .....	156
Férjük hozzá a main() argumentumaihoz .....	158
<b>10. fejezet: A C++ előfeldolgozója .....</b>	<b>163</b>
Mi az az előfeldolgozó? .....	163
Ágyazzunk be fájlokat .....	164
#Definiáljunk dolgokat .....	168
Oké, mi lenne, ha nem #definiálnánk dolgokat?.....	170
Vegyük számba az egyéb lehetőségeket .....	171
Ágyazzunk be dolgokat, de csak #ha azt mondjuk.....	173
Belsőleg definiált objektumok.....	174
Typedef .....	176
<b>3. rész: ISMERKEDJÜNK MEG AZ OSZTÁLYOKKAL.....</b>	<b>177</b>
<b>11. fejezet: Ismerkedjünk meg az objektumorientált programozással .....</b>	<b>179</b>
Absztraháljuk a mikrohullámú sütőket.....	179
Készítsünk funkcionális nachost.....	180
Készítsünk objektumorientált nachost .....	181
Osztályozzuk a mikrohullámú sütőket.....	181
Miért érdemes osztályozni? .....	182
<b>12. fejezet: Léptessük magasabb osztályba a C++-t.....</b>	<b>185</b>
Ismerkedjünk meg az osztállyal .....	185
Az osztályok formátuma .....	186
Férjük hozzá egy osztály tagjaihoz.....	187
Aktiváljuk az objektumainkat .....	187
Modellezzünk valós objektumokat .....	188
Miért érdemes a tagfüggvényekkel vesződni?.....	189
Adjunk hozzá egy tagfüggvényt .....	189
Hívjunk meg egy tagfüggvényt .....	190
Férjük hozzá más tagokhoz egy tagfüggvényből .....	192
Hatókörfeloldás (persze nem vízben oldjuk fel).....	194
Definiáljunk tagfüggvényt az osztályban .....	195
Őrizzünk meg egy tagfüggvényt az osztályon kívül is.....	197
Terheljünk túl tagfüggvényeket.....	199
<b>13. fejezet: Mutassunk rá objektumokra, és bámuljuk őket.....</b>	<b>201</b>
Deklaráljuk objektumok tömbjeit .....	201
Deklaráljuk objektumok mutatóit .....	202
Oldjuk fel egy objektummutató hivatkozását.....	203
Mutassunk rá mutatónyilakkal .....	204
Adjunk át objektumokat függvényeknek .....	205
Hívjunk meg egy függvényt egy objektumértékkel .....	205
Hívjunk meg egy függvényt egy objektummutatóval .....	206
Hívjunk meg egy függvényt a hivatkozási operátorral.....	208
Miért érdemes mutatókkal vagy hivatkozásokkal vesződni?.....	209

Térjünk vissza a halomhoz .....	210
Foglaljuk le objektumok halmait .....	210
Amikor lefoglalják a memóriát helyetted .....	211
Kapcsoljunk össze objektumokat láncolt listákkal .....	212
Végezzünk egyéb műveleteket láncolt listákon .....	213
Kössünk össze objektumokat a LáncoltListaAdatok programmal .....	214
Egy reménysugár: a C++ könyvtárában megtalálható konténerek listája .....	217
<b>14. fejezet: Védjük meg a tagokat: Ne zavarjanak! .....</b>	<b>219</b>
Védjük meg a tagokat .....	219
Miért van szükség védett tagokra? .....	219
Fedezzük fel, hogy hogyan működnek a védett tagok .....	220
Lássuk az érveket a védett tagok használata mellett .....	222
Védjük meg az osztály belső állapotát .....	222
Használjunk egy korlátozott felületű osztályt .....	223
Adjunk hozzáférést védett tagokhoz nem tag függvényeknek .....	223
<b>15. fejezet: „Ó, miért építesz fel, csak hogy rögtön le is bonts?” .....</b>	<b>227</b>
Hozzunk létre objektumokat .....	227
Használjunk konstruktorokat .....	228
Építsünk fel egyetlen objektumot .....	229
Építsünk fel több objektumot .....	230
Építsünk fel egy emeletes objektumot .....	231
Szedjünk ízekre egy destruktort .....	233
Miért van szükség a destruktorra? .....	233
Használjunk destruktorkat .....	234
<b>16. fejezet: Használjunk konstruktorokat argumentumokkal .....</b>	<b>239</b>
Lássunk el konstruktorokat argumentumokkal .....	240
Használjunk egy konstruktort .....	240
Amikor túl sokat kérünk az építőmunkásoktól: Terheljük túl a konstruktort .....	242
Állítsunk be alapértelmezett értékeket alapértelmezett konstruktorokban .....	246
Építsünk fel osztálytagokat .....	248
Építsünk fel egy összetett adattagot .....	248
Építsünk fel egy konstans adattagot .....	254
Szervezzük át a felépítés sorrendjét .....	254
A helyi objektumok sorban épülnek fel .....	255
A statikus objektumok csak egyszer épülnek fel .....	255
Minden globális objektum a main() függvény előtt épül fel .....	257
A globális objektumok felépítésének nincs meghatározott sorrendje .....	257
A tagok abban a sorrendben épülnek fel, amelyben deklarálnak vannak .....	259
A destruktorkat a konstruktorok fordított sorrendjében végzik el a megsemmisítést .....	259
Építsünk fel tömböket .....	259
Konstruktorok mint konverziós eszközök .....	260

<b>17. fejezet: A másoló/áthelyező konstruktor .....</b>	<b>263</b>
Másoljunk egy objektumot .....	263
Miért van szükség a másoló konstruktorra? .....	263
Használjuk a másoló konstruktort.....	264
Az automatikus másoló konstruktor .....	266
Vessük össze a részleges másolatok és a teljes másolatok létrehozását .....	267
Hosszú az út az ideiglenes objektumokkal .....	272
Kerüljük el az ideiglenes objektumok használatát, végleg .....	273
Az áthelyező konstruktor.....	274
<b>18. fejezet: Statikus tagok: Talán egy kis olajozás segít?.....</b>	<b>277</b>
Definiáljunk statikus tagokat .....	277
Miért van szükség statikus tagokra? .....	277
Használjunk statikus tagokat.....	278
Hivatkozzunk statikus adattagokra .....	279
Hogyan használhatunk statikus adattagokat?.....	280
Deklaráljunk statikus tagfüggvényeket .....	281
Miről is van szó tulajdonképpen?.....	284
<b>4. RÉSZ: ÖRÖKLŐDÉS .....</b>	<b>287</b>
<b>19. fejezet: Örököljünk egy osztályt .....</b>	<b>289</b>
Szükség van az örökségre? .....	290
Hogyan örököl egy osztály?.....	290
Használjunk egy alosztályt .....	293
Építsünk fel egy alosztályt .....	294
Semmisítsünk meg egy alosztályt .....	295
Örököljünk konstruktorokat .....	295
Legyünk RENDELKEZIK kapcsolatban .....	296
<b>20. fejezet: Ismerkedjünk meg a virtuális tagfüggvényekkel: ezek tényleg valódiak? .....</b>	<b>299</b>
Miért van szükség a polimorfizmusra? .....	302
Hogyan működik a polimorfizmus? .....	303
Mikor fordulhat elő, hogy egy virtuális függvény nem is az?.....	304
Vegyük át a virtuális szempontokat .....	306
<b>21. fejezet: Bontsunk újra osztályokat.....</b>	<b>309</b>
Végezzünk újrabontást.....	309
Valósítsunk meg absztrakt osztályokat.....	313
Írjuk le az absztrakt osztályok fogalmát.....	314
Csináljunk egy absztrakt osztályból tisztességes osztályt .....	316
Adjunk át absztrakt osztályokat .....	316
<b>5. rész: BIZTONSÁG.....</b>	<b>319</b>



<b>22. fejezet: Egy új értékadó operátor, ha elfogadod .....</b>	<b>321</b>
Vessük össze az operátorokat és a függvényeket .....	321
Szűrjünk be egy új operátort.....	322
A sekély másolatok mélyreható problémákkal járnak .....	323
Definiáljuk felül az értékadó operátort .....	324
Definiáljuk felül az index operátort .....	329
Az áthelyező konstruktor és az áthelyező operátor .....	330
<b>23. fejezet: Használjunk adatfolyam-bemenetet és -kimenetet.....</b>	<b>335</b>
Hogyan működik az adatfolyam-bemenet és -kimenet?.....	335
Az alapértelmezett adatfolyam-objektumok.....	336
Adatfolyam-bemenet és -kimenet .....	337
Megnyitási módok.....	339
Hé, fájl, milyen az állapotod?.....	340
Elő tudunk erre állni egy példával?.....	340
Az adatfolyamosztályok egyéb metódusai .....	344
Olvassunk és írjunk adatfolyamokat közvetlenül .....	346
Szabályozzuk a formátumot .....	348
Mit csinál az endl?.....	349
Pozicionáljuk a mutatót egy fájlban .....	350
Használjuk a stringstream alosztályokat .....	351
Manipuláljunk manipulátorokat.....	354
<b>24. fejezet: Kezeljünk hibákat és kivételeket.....</b>	<b>357</b>
Indokoljuk meg, hogy miért kezeljük új módon a hibákat.....	359
Ismerkedjünk meg a kivételkezelés mechanizmusával.....	360
Milyen dolgokat dobhatunk el?.....	363
Már adjuk is tovább.....	366
<b>25. fejezet: Örökljünk többszörös öröklődéssel .....</b>	<b>367</b>
Írjuk le a többszörös öröklődés mechanizmusát.....	367
Tisztázzuk az öröklődés kétértelműségeit .....	369
Lépjünk tovább a virtuális öröklődéssel .....	370
Építsünk fel többszörös öröklődésű objektumokat .....	377
Hallgassunk meg egy ellentétes véleményt.....	377
<b>26. fejezet: Próbáljuk ki a C++ sablonokat .....</b>	<b>379</b>
Általánosítsunk egy függvényt sablonná .....	380
Osztálysablonok .....	382
Tipppek a sablonok használatához .....	385
Külső sablonpéldányosítások .....	386
Valósítsunk meg egy inicializáló listát .....	386
<b>27. fejezet: Szabványosítsunk a szabványos sablonkönyvtárral .....</b>	<b>389</b>
A string konténer .....	390
Járjunk be listákat.....	395
Járjunk be egy listát .....	397
Végezzünk műveleteket egy teljes listán.....	398
Elő tudunk erre állni egy példával?.....	399

**28. fejezet: Írjunk olyan kódot, amit nem lehet meghekkelni ..... 401**

Ismerjük meg a hekker indítékait .....	401
Ismerjük meg, hogy mi az a kódinjektálás .....	404
Vizsgáljunk meg egy példát az SQL-injektálásra .....	404
Akadályozzuk meg a kódinjektálást.....	406
Idézzük elő pufferek túlcsordulását, a móka és a haszon kedvéért ....	407
Megnézhetünk erre egy példát? .....	408
Hogyan épül fel a hívási verem?.....	411
Hekkeljük meg a PufferTúlcsordulás programot .....	414
Kerüljük el a puffertúlcsordulást – az első kísérlet .....	418
Kerüljük el a puffertúlcsordulást – a második kísérlet .....	420
A string osztály egy újabb argumentuma .....	423
Miért nem használunk mindig sztringfüggvényeket? .....	424

**6. rész: TOP 10..... 427****29. fejezet: Tíz módszer, amellyel elkerülheted a hibákat**

<b>a programodban.....</b>	<b>429</b>
Engedélyezzük az összes figyelmeztető és hibaüzenetet .....	429
Használjunk világos és következetes kódolási stílust.....	430
Korlátozzuk a láthatóságot.....	431
Helyezzünk el megjegyzéseket a kódban írás közben.....	432
Lépésenként járjunk be minden útvonalat legalább egyszer .....	433
Kerüljük el a felüldefiniált operátorok használatát .....	433
Kezeljük módszeresen a halommemóriát.....	434
Kivételekkel kezeljük a hibákat .....	434
Deklaráljuk a destruktorkat virtuálisként .....	434
Kerüljük a többszörös öröklődés használatát .....	436

**30. fejezet: Tíz módszer, amellyel megvédheted a programjaidat**

<b>a hekkerektől .....</b>	<b>437</b>
Ne feltételezzünk semmit a felhasználói bevitelről.....	437
Elegánsan kezeljük a hibákat .....	438
Tartsunk fenn programnaplót .....	439
Kövessünk egy jó fejlesztési folyamatot.....	441
Alkalmazzunk jó verziókezelést .....	442
Biztonságosan hitelesítsük a felhasználókat.....	443
Kezeljünk távoli munkameneteket .....	446
Rejtsük el a kódot .....	447
Írjuk alá a kódot egy digitális tanúsítvánnyal .....	450
Használjunk biztonságos titkosítást, ahol csak szükséges .....	451

**Tárgymutató ..... 453**