

Tartalomjegyzék

Előszó	xiii
1. Bevezetés	1
2. A C++ nem objektumorientált újdonságai	3
2.1. A C és a C++ nyelv	3
2.1.1. Függvényparaméterek és visszatérési érték	3
2.1.2. A main függvény	4
2.1.3. A bool típus	4
2.1.4. C stílusú több-bájtos sztringek	5
2.1.5. Változódeklaráció mint utasítás	5
2.2. Függvénynevek túlterhelése	6
2.3. Alapértelmezett függvényargumentumok	8
2.4. Paraméterátadás referenciatípussal	10
3. Objektumok és osztályok	17
3.1. Az objektumorientáltság alapelvei	17
3.2. Egységbe záras a C++-ban	20
3.2.1. Tagváltozók	22
3.2.2. Tagfüggvények	22
3.3. Adatrejtés	25
3.4. Konstruktorok és destruktorok	28
3.5. Dinamikus adattagot tartalmazó osztályok	31
3.5.1. Dinamikus memóriakezelés	32
3.5.2. Dinamikus adattagok támogatása	34
3.5.3. A másolókonstruktor	39
3.5.4. Összefoglalás	43
3.6. Friend függvények és osztályok	44
3.6.1. Friend függvények	44
3.6.2. Friend osztályok	45
3.6.3. A friend viszony tulajdonságai	46
3.7. Tagváltozók inicializálása	47
3.8. Statikus tagok	50
3.9. Beágyazott definíciók*	54

4.	Konstansok és inline függvények	59
4.1.	A const használata	59
4.1.1.	Konstans „változók”	59
4.1.2.	Konstans pointerek	60
4.1.3.	Konstans függvényparaméterek	61
4.1.4.	Konstans tagváltozók	62
4.1.5.	Konstans tagfüggvények	63
4.1.6.	Mutable tagváltozók	66
4.1.7.	Const – nem const konverziók	66
4.2.	Inline függvények	68
5.	A C++ I/O alapjai	73
5.1.	A szabványos adatfolyamok	73
5.2.	Manipulátorok és formázás	80
5.3.	Állománykezelés	86
5.4.	Egyéb műveletek	90
6.	Operátorok és túlterhelésük	93
6.1.	Az operátorokról általában	93
6.2.	Függvényszintaxis és túlterhelés	94
6.3.	Példa: egy komplex számokat megvalósító osztály	96
6.4.	Speciális operátorok túlterhelése	102
6.5.	Általános szabályok	106
6.6.	<i>A pointer-tag operátorok*</i>	<i>107</i>
7.	Általánosítás és specializáció	111
7.1.	Tagváltozók és tagfüggvények öröklése	112
7.2.	Behelyettesíthetőség	118
7.3.	A virtuális függvények megvalósítása	132
7.4.	Többszörös öröklés	137
7.4.1.	A többszörös öröklés gyakorlati alkalmazása	140
7.4.2.	<i>A többszörös öröklés és a void*</i>	<i>152</i>
7.5.	Virtuális öröklés	153
7.6.	A konstruktorok és destruktorok automatikus feladatai	159
8.	Típuskonverziók	165
8.1.	Beépített típusok közti típuskonverziók	165
8.2.	A felhasználói típusok konverziói	167
8.2.1.	Konverzió független típusok között	168
8.2.2.	Konverzió az öröklési hierarchia mentén	173
8.3.	A C++ típuskonverziós operátorai	175

9. Névterek	179
9.1. Bevezetés a névterek használatába	179
9.2. A globális hatókör elérése	182
9.3. Tagfüggvények definiálása	182
9.4. Using deklarációk	183
9.5. Argumentumfüggő névfeloldás	184
9.6. Névtér alias	185
10. Kivételkezelés	187
10.1. A kivételek használatának alapjai	187
10.1.1. Hagyományos hibakezelés	187
10.1.2. A kivételkezelés alapjai	189
10.1.3. Egymásba ágyazott try-catch blokkok	195
10.1.4. Az elkapott kivétel újradobása	196
10.1.5. A verem visszacsévézése	197
10.2. Kivételkezelés a gyakorlatban	198
10.2.1. A kivételek típusossága	198
10.2.2. Kivételhierarchiák	200
10.2.3. Saját kivételosztályok	205
10.2.4. Kezeletlen kivételek	208
10.2.5. Függvények kivételspecifikációja	208
10.3. Kivételkezelési technikák	209
10.3.1. Erőforrás-kezelés	210
10.3.2. Bevezetés az auto_ptr használatába	212
10.3.3. A konstruktor és a destruktor működése	214
11. C++ sablonok	217
11.1. Függvénysablonok	218
11.1.1. Függvénysablon-specializáció	221
11.1.2. Példák függvénysablonokra	223
11.1.3. A hívott függvény kiválasztása	224
11.2. Osztálysablonok	225
11.2.1. Osztálysablonok írása	228
11.2.2. Bővebben a sablonparaméterekről	235
11.2.3. Alapértelmezett sablonparaméterek	241
11.2.4. Pointerek, referenciák és konstansok, mint sablonparaméterek	243
11.2.5. Tagfüggvénysablonok	244
11.2.6. Az osztálysablonok és az öröklés	245
11.2.7. <i>Osztálysablon-specializáció*</i>	<i>247</i>
11.2.8. A sablonok fordítása	250
11.2.9. <i>További osztálysablon technikák*</i>	<i>256</i>

12. Bevezetés a C++ szabványos sablonkönyvtárába	263
12.1. Alapkonceptiók	264
12.1.1. Tárolók	265
12.1.2. Iterátorok	276
12.1.3. Algoritmusok	286
12.2. Függvényobjektumok*	296
12.2.1. Asszociatív tárolók rendezése	296
12.2.2. Előregyártott függvényobjektumok	299
12.3. Szabványos algoritmusok*	304
12.3.1. Az algoritmusok áttekintése	305
12.3.2. Nem módosító algoritmusok	311
12.3.4. Sorrendváltoztató algoritmusok	323
12.3.5. Rendezett tartományt feltételező algoritmusok	334
12.3.6. Numerikus algoritmusok	339
12.4. STL tárolók*	342
12.4.1. Szekvenciális tárolók	344
12.4.2. Asszociatív tárolók	351
12.4.3. Állandó és változó méretű bittömbök	356
12.4.4. A tárolók műveleteinek összehasonlítása	358
12.4.5. Nem szabványos tárolók	364
12.5. Szövegkezelés	364
12.5.1. Inicializálás	365
12.5.2. Méret és memóriakezelés	366
12.5.3. Alapműveletek	367
12.5.4. I/O függvények	376
12.5.5. Szöveges konverziók*	380
12.6. Magyar nyelvű fejlesztés	384
12.6.1. A szabványos nemzetközi karakterkészlet	385
12.6.2. A nyelvi környezetek programozása*	394
12.7. További lehetőségek	426
13. A modellezés és a C++	427
13.1. Osztályok	428
13.2. Kapcsolatok	430
13.2.1. Asszociációk	431
13.2.2. Sablonok modellezése	442
13.3. Kódgenerálás és kódvisszafejtés	443
14. Esettanulmány	445
14.1. Alapfunkciók	445
14.1.1. A feladat áttekintése	445
14.1.2. A termékek reprezentálása	446
14.1.3. Összetett termékek	453

14.1.4. Termékek nyilvántartása	456
14.1.5. Termékobjektumok létrehozása	459
14.1.6. Tesztelés	464
14.2. További funkciók az STL használatával	466
15. Utószó	471
„A” függelék: A C++ operátorok precedenci táblázata	473
„B” függelék: Fejlesztőkörnyezetek	475
B.1. Microsoft Visual C++	475
B.1.1. Új solution létrehozása	475
B.1.2. A fejlesztés környezete	476
B.1.3. Fordítás, futtatás és nyomkövetés	478
B.1.4. Solutionkezelés	480
B.1.5. Néhány hasznos funkció	481
B.1.6. Dokumentáció	483
B.2. A GNU C++ fordító	484
„C” függelék: Áttérés C nyelvről C++ nyelvre	487
Kislexikon	491
Tárgymutató	501
Irodalomjegyzék	505