

Úvod

Obsah

Úvod	I
Obsah	I
Seznam tabulek	XIII
Seznam obrázků	XIV
Seznam ukázkových příkladů	XV
Předmluva	XVIII
Typografické konvence a značky	XX

Část I - Ovládání Borland C++

1. Historie Borland C++	1
1.1. Historie jazyků C a C++	1
1.2. Přehled verzí Borland C++ a Turbo C/C++	2
2. Integrované vývojové prostředí	5
2.1. Integrované prostředí pro DOS	5
2.1.1. Spouštění integrovaného prostředí a obecné informace	5
2.1.2. Horké klávesy	7
2.1.3. Integrovaný editor	8
2.1.4. Některé volby pro prostředí a integrovaný editor	10
2.1.5. Volby pro spuštění integrovaného prostředí	11
2.2. Integrované prostředí pro Windows	11
2.2.1. Spouštění integrovaného prostředí a obecné pokyny	12
2.2.2. Horké klávesy	13
2.2.3. Integrovaný editor	14
2.2.4. Některé volby pro prostředí a integrovaný editor	15
2.2.5. AppExpert a ClassExpert (Borland C++ 4.xx)	16
2.3. Integrovaný překladač	18

2.3.1. Používání integrovaného překladače	19
2.3.2. Některé volby integrovaného překladače	20
2.3.2.1. Nastavování cílové platformy	20
2.3.2.2. Nastavování adresářů	22
2.3.2.3. Volby překladače	22
2.3.2.4. Volby linkeru	23
2.4. Správce projektů	24
2.5. Integrovaný debugger	25
3. Nástroje pro DOS	27
3.1. Samostatný překladač	27
3.1.1. 32bitový překladač	30
3.2. Samostatný linker	30
3.3. Preprocessor	33
3.4. Knihovnick (Turbo Librarian)	34
3.5. MAKE	35
3.6. PRJ2MAK	37
3.7. GREP	37
3.8. Turbo Help (BC++ 3.1)	38
3.9. TEMC a KeyMapper	38
4. Nástroje pro Windows	40
4.1. Resource Workshop	40
4.1.1. Prostředky	41
4.1.2. Soubory s prostředky a projekty v Resource WorkShopu	43
4.2. Překladače a linkery souborů s prostředky	44
4.3. Nástroje pro práci s knihovnamí importovaných funkcí	45
4.3.1. IMPDEF	45
4.3.2. IMPLIB a IMPLIBW	45
4.4. WinSight	45
4.5. WinSpector	46
4.6. Help Compiler	47

5. Velké společné nástroje (pro DOS i Windows)	48
5.1. Turbo Debugger	48
5.1.1. Obecné informace o ladění programů	49
5.1.2. Možnosti, jak procházet program	49
5.1.3. Další možnosti Turbo Debuggeru	50
5.1.4. Ladění programů pro Windows	51
5.1.5. Další nástroje k Turbo Debuggeru	52
5.2. Turbo Profiler	52
5.2.1. Možnosti Turbo Profileru	53
5.2.2. Profilování programů pro Windows	54
5.3. Turbo Assembler	54

Část II - Základy programování v jazyce C a C++

6. Struktura programů	56
6.1. Vnější struktura programu	56
6.1.1. Moduly	56
6.1.2. Jak probíhá překlad	57
6.2. Vnitřní struktura programu	57
6.2.1. Strukturované programování	58
6.2.2. Jednotlivé složky programu	58
7. Jednoduché programy	61
7.1. První program - výpis na obrazovku	61
7.1.1. Psaní a překlad prvního programu	61
7.1.2. Chyby	63
7.2. Druhý program - formátování výstupů	64
7.3. Volba C a C++ překladače	66

Část III - Základní programové konstrukce

8. Proměnné, základní typy, konstanty a vstupy	67
8.1. Základní typy	67
8.2. Proměnné	68
8.2.1. Definování proměnných	68
8.2.2. Základní operace s proměnnými	69
8.2.3. Globální a lokální proměnné	70
8.3. Konstanty	70
8.3.1. Číselné konstanty (ordinální)	71
8.3.2. Číselné konstanty (reálné)	71

8.3.3. Znakové konstanty	72
8.3.4. Řetězcové a enumerické konstanty	73
8.4. Formátování proměnných pro výstup (a vstup)	73
8.5. Vstupy	76

Část III - Základní programové konstrukce

9. Definice proměnných, typy	80
9.1. Definice proměnných	80
9.1.1. Základní typy	81
9.1.2. Modifikátory	81
9.1.3. Paměťové třídy	82
9.2. Typy	85
9.3. Definování nových typů (typedef)	86
10. Výrazy a operátory	88
10.1. Konstrukce výrazu	88
10.2. Operátory	88
10.2.1. Aritmetické operátory	91
10.2.2. Bitové operátory	93
10.2.3. Relační a logické operátory	94
10.2.4. Přiřazovací operátory	95
10.2.5. Operátory pro práci s ukazateli	96
10.2.6. Operátory přístupu ke složkám	96
10.2.7. Operátor "sizeof"	97
10.2.8. Převody mezi typy (operátor přetypování)	97
10.2.9. Ostatní operátory	98
10.3. Priorita operátorů	99
10.4. Volání funkcí	100
10.5. Přetěžování operátorů	100
11. Řízení běhu programu	101
11.1. Podmínky	101
11.1.1. Podmínky pomocí operátoru ? :	103
11.2. Cykly	104
11.2.1. Cyklus While	104
11.2.2. Cyklus Do-While	105
11.2.3. Cyklus For	107
11.3. Příkaz break	110

11.4. Příkaz switch	111
11.5. Příkaz goto a návěští	113
12. Funkce	115
12.1. Na co se používají funkce	115
12.2. Definování funkcí	115
12.3. Volání funkcí	116
12.4. Zvláštní případy	117
12.5. Deklarace prototypu	120
12.6. Parametry volané odkazem	121
12.7. Rekurze a rekurzivní funkce	122
12.8. "Inline" funkce	123
12.9. Nepoužité parametry	124
12.10. Nepovinné (implicitní) parametry	124
12.11. Přetěžování funkcí	126
12.12. Starší způsob definice funkcí	128
12.13. Standardní knihovna funkcí	128
13. Preprocessor	129
13.1. O preprocessoru	129
13.2. Jednoduchá makra	129
13.3. Falešná makra	131
13.4. Podmíněný překlad, testování existence a obsahu makra	131
13.5. Makra s parametry	133
13.6. Direktiva #include	134
13.7. Direktiva "#line"	135
13.8. Direktiva "#error"	135
13.9. Direktiva "#pragma"	136
13.10. Předdefinovaná makra	136
13.11. Preprocessor pro příkazovou řádku	138

Část IV - Složitější datové typy

14. Výčtový typ	139
14.1. Jednoduchý výčtový typ	139
14.2. Hodnotami doplněný	141
14.3. Typ boolean	143

15. Ukazatele	144
15.1. Ukazatel a instance	144
15.2. Ukazatele na objekty	145
15.2.1. Definice ukazatele na objekt	145
15.2.2. Používání ukazatelů na objekt	146
15.3. Ukazatele na funkce	148
15.3.1. Definice ukazatele na funkci	148
15.3.2. Používání ukazatele na funkci	149
15.4. Pravidla bezpečnosti práce s ukazateli	150
15.5. Parametry volané odkazem	152
15.6. Ukazatel na ukazatele	153
15.7. Načítání a tištění ukazatelů	154
15.8. Konstanta "NULL"	154
15.9. Neurčitý ukazatel, ukazatel na cokoliv	155
15.10. Ukazatele a konstanty	155
15.11. Aritmetika s ukazateli	156
15.11.1. Porovnávání ukazatelů	156
15.11.2. Sčítání (odečítání) ukazatele a čísla	157
15.11.3. Odečítání dvou ukazatelů	157
16. Pole	159
16.1. Definice pole	159
16.2. Používání pole	159
16.3. Co mají pole společného s ukazateli	160
16.4. Inicializace pole	161
16.5. Přístup k polím pomocí aritmetiky ukazatelů	162
16.6. Vícerozměrná pole	162
16.7. Pole jako parametr funkcí	165
16.8. Funkce pro práci s poli	167
16.9. Řetězce	167
17. Řetězce	168
17.1. Řetězce jako pole	168
17.2. Definice řetězce	168
17.3. Načítání a vypisování řetězců	169
17.3.1. Problémy s načítáním řetězců	170
17.4. Řetězce jako parametry funkcí	172
17.5. Konstantní řetězec (literál)	174
17.6. Funkce pro práci s řetězcí	175

18. Objekty, objektově orientované programování	176
18.1. Teorie objektově orientovaného programování	176
18.1.1. Terminologie kolem objektů	178
18.2. Základní deklarace objektu	179
18.2.1. Typy objektů, (struct a class)	181
18.3. Přístup k obsahu objektu	182
18.3.1. Příkaz friend	183
18.4. Položky objektů	184
18.4.1. Deklarování položek objektů	184
18.4.1.1. Velikost objektů	186
18.4.1.2. Operátor :: (příslušnost k objektu)	187
18.4.2. Používání položek objektů	187
18.4.3. Bitové položky objektů	189
18.4.3.1. Nevyužité bitové položky	193
18.4.4. Statické položky objektů	193
18.4.5. Konstantní položky objektů, reference v objektech a objektové položky s konstruktorem	196
18.4.6. Objekt odkazující se sám na sebe	197
18.4.7. Ukazatele do objektů	198
18.5. Metody objektů	200
18.5.1. Funkce	201
18.5.1.1. Deklarace, definice a používání funkcí jako metod	201
18.5.1.2. Přetěžování funkcí	205
18.5.2. Ukazatel "na sebe" (this)	205
18.5.3. Konstruktory	206
18.5.3.1. Standardní konstruktory	209
18.5.3.2. Konstruktory položek a předků	210
18.5.4. Destruktory	211
18.5.5. Desatero konstruktorů a destruktorů	213
18.5.6. Inline metody	214
18.5.7. Statické metody	217
18.5.8. Konstantní metody	217
18.5.8.1. Přetěžování konstantních metod nekonstantními	220
18.5.9. Operátory jako metody	220
18.6. Dědičnost objektů	222
18.6.1. Principy a způsoby dědění	222

18.6.1.1. Dědění z více předků současně	225
18.6.2. Přiřazování předků a potomků	226
18.6.3. Dědění složek	227
18.6.3.1. Dědění položek a funkcí	227
18.6.3.2. Změna přístupových práv ke složkám ...	230
18.6.3.3. Dědění konstruktorů a destruktorů	231
18.6.4. Překrývání složek	234
18.6.5. Virtuální metody	237
18.6.5.1. Čisté virtuální funkce a abstraktní objekty	239
18.6.5.2. Virtuální destruktory	240
18.6.5.3. Virtuální funkce a konstruktory	241
18.6.6. Násobná dědičnost	241
18.7. Vnořené typy	243
18.8. Objekty a přetěžování operátorů	245
<u>19. Unie</u>	<u>246</u>
19.1. Anonymní unie	247
<u>20. Dynamická alokace paměti</u>	<u>250</u>
20.1. Na co se používá a jak funguje dynamická	250
20.2. Alokování a dealokování operátory new a delete	251
20.2.1. Operátory new a delete s konstruktory a destruktory	253
20.2.2. Operátor new a výjimky (pouze BC++ 4.xx)	254
20.3. Alokování a dealokování funkcemi jazyka C	255
20.3.1. Alokování a uvolňování velkých bloků	258
<u>21. Ovládání paměti</u>	<u>259</u>
21.1. 16bitové platformy	259
21.1.1. Paměťové modely	259
21.1.2. Blízké (near), vzdálené (far) a normalizované (huge) ukazatele	261
21.1.3. Používání překryvných segmentů	262
21.2. 32bitové platformy	262
21.2.1. Obecné principy 32bitových programů	263
21.2.2. 32bitové programy pod DOSem	263

22. Reference	264
22.1. Používání typu reference	264
22.2. Parametry volané odkazem pomocí referencí	265
23. Přetěžování operátorů	267
23.1. Na co je přetěžování operátorů	267
23.2. Rozdělení operátorů pro přetěžování	268
23.3. Přetěžování operátorů jako funkce	268
23.3.1. Přetěžování operátorů ++ a --	270
23.3.2. Přetěžování operátorů přiřazení	273
23.4. Přetěžování operátorů jako metody	274
23.4.1. Přetížení operátoru přiřazení	277
23.4.2. Přetížení operátoru přetypování	278
23.4.3. Přetížení operátorů new a delete	280
23.4.4. Ostatní netypické operátory	281
24. Šablony	282
24.1. Principy šablon	282
24.2. Deklarace šablon a jejich parametry	282
24.2.1. Typové parametry	283
24.2.2. Hodnotové parametry	283
24.3. Šablony funkcí	284
24.4. Šablony objektů	287
24.4.1. Šablony a dědičnost	289
25. Ošetřování výjimek v C++	292
25.1. Účel a způsoby ošetřování výjimek	292
25.2. Ošetřování výjimek	292
25.2.1. Co s neošetřenými a neočekávanými výjimkami ...	296
25.2.2. Standardní výjimky Borland C++	296
26. Informace o typech za běhu programu a dynamické přetypování	298
26.1. Identifikace typů	298
26.2. Dynamické převody typů	300
26.2.1. Operátor const_cast	300
26.2.2. Operátor dynamic_cast	300
26.2.3. Operátor reinterpret_cast	301
26.2.4. Operátor static_cast	301

Část V - Knihovna Borland C++

27. Základní funkce	302
27.1. Funkce na ukončení programu	302
27.2. Náhodná čísla	303
27.3. Zvuky	305
27.4. Další důležité funkce	306
28. Funkce pro vstupy a výstupy	310
28.1. Pojmy okolo vstupů a výstupů	310
28.2. Funkce pro práci se standardním vstupem a výstupem	311
28.3. Funkce pro práci s konzolou	314
28.3.1. Funkce pro práci s kurzorem	314
28.3.2. Funkce pro práci s klávesnicí	316
28.3.3. Funkce pro práci s barvami	318
28.3.4. Funkce pro vstup a výstup textu na konzolu	322
28.3.5. Funkce pro práci s obrazovkou jako celkem	324
28.4. Knihovna EasyWin	330
28.5. Funkce pro souborové vstupy a výstupy	330
28.5.1. Obecné principy souborových vstupů a výstupů	331
28.5.2. Funkce pro práci s proudy	331
28.5.3. Funkce pro práci s handlery	342
28.5.4. Jiné periferie než disk	347
28.5.4.1. Obrazovka	347
28.5.4.2. Tiskárna	348
28.5.4.3. Porty	348
28.5.4.4. NUL	348
28.5.5. Ošetření chyb vstupů a výstupů	349
28.5.6. Standardní vstupy a výstupy	351
29. Funkce pro práci s pamětí	353
29.1. Funkce pro manipulaci s pamětí	353
29.2. Funkce pro dynamickou alokaci paměti	356
30. Funkce pro práci s řetězcí	357

31. Práce se soubory a funkce DOSu	366
31.1. Funkce pro práci se soubory	366
31.2. Funkce pro práci s adresáři a disky	368
31.3. Některé další funkce DOSu	375
32. Matematická knihovna	378
32.1. Reálná čísla, matematický koprocessor	378
32.1.1. Reálná čísla	378
32.1.2. Matematický koprocessor a jeho emulace	379
32.2. Funkce matematické knihovny	379
32.3. Matematické konstanty	383
33. Grafická knihovna BGI	384
33.1. Základní principy	384
33.2. Otevírání a uzavírání grafického zařízení	385
33.3. Grafický kurzor a funkce pro práci s ním	388
33.4. Barvy	389
33.5. Výplně	391
33.6. Kreslení základních útvarů	393
33.7. Text a fonty	396
33.8. Další funkce	398
34. Knihovna objektově orientovaných proudů	402
34.1. Teorie objektově orientovaných proudů	402
34.2. Komponenty knihovny iostream	402
34.3. Základní manipulace s proudy	406
34.3.1. Předdefinované proudy	406
34.3.2. Operátory << a >>	406
34.3.2.1. Přetěžování operátorů << a >> pro nестandardní typy	408
34.3.3. Formátování pomocí manipulátorů	409
34.3.3.1. Definování vlastních manipulátorů	412
34.3.4. Důležité metody proudů	413
34.4. Souborové proudy	416
34.4.1. Metody souborových proudů	418
34.5. Paměťové (interní proudy)	420
34.5.1. Metody paměťových proudů	422
34.6. Konzolové proudy	423
34.6.1. Metody konzolových proudů	424

Dodatky

A. Klíčová slova	427
B. Použitá a doporučená literatura	428
Rejstřík	429