

Obsah

Úvod	15
Co je nového ve druhém vydání	15
Co je obsahem 2. dílu této knihy	15
Jak získat 2. díl	16
Co je potřeba vědět před čtením této knihy	16
Jak se učit jazyk C++	16
Cíle	17
Kapitoly	18
Cvičení	21
Řešení cvičení	22
Zdrojový kód	22
Jazykové standardy	23
Jazyková podpora	23
Elektronická verze semináře	24
CD ROMy, semináře a konzultace	24
Chyby	24
Použité konvence	25
České verze programů	25
1. Úvod do objektů	26
1.1 Postup abstrakce v programovacích jazycích	26
1.2 Rozhraní (interface) objektu	28
1.3 Skrytá implementace	29
1.4 Opětné použití implementace	30
1.5 Dědičnost – opětné použití rozhraní	31
Relace „je“ a „je-podobný“	34
1.6 Zaměnitelné objekty a polymorfismus	35
1.7 Vytvoření a zrušení objektu	37
1.8 Ošetření výjimek, zpracování chyb	38
1.9 Analýza a návrh	39
Fáze 0: Příprava plánu	41
Fáze 1: Co budeme vytvářet?	42
Fáze 2: Jak budeme systém budovat?	43
Fáze 3: Vytvoření jádra	46
Fáze 4: Iterace případů užití	47
Fáze 5: Evoluce	47
Plánování se vyplatí	48
1.10 Extrémní programování	49
Nejprve pište testy	49
Programování v páru	50
1.11 Proč má C++ úspěch	51
Zlepšení jazyka C	51
Efektivnost	52
Systémy lze snáze popsat a pochopit	52
Velký význam knihoven	52

Opětné použití zdrojového kódu pomocí šablon	52
Ošetření chyb	52
Rozsáhlé programové celky	53
1.12 Přechod od procedurálního programování k OOP	53
Návod	53
Organizace týmu	54
1.13 Shrnutí.....	56
2. Vytváření a používání objektů	57
2.1 Proces překladu jazyka	57
Interprety	57
Kompilátory.....	58
Proces překladu.....	59
2.2 Nástroje pro oddělený překlad	60
Deklarace versus definice	60
Spojování.....	65
Používání knihoven	65
2.3 Váš první program v C++.....	67
Používání třídy iostreams	67
Jmenné prostory (Namespaces)	67
Základy struktury programu.....	69
„Ahoj lidi!“ („Hello, world!“)	69
Spouštění kompilátoru.....	70
2.4 Více o vstupně-výstupních proudech	71
Spojování znakových polí.....	71
Čtení vstupu	72
Volání jiných programů.....	72
2.5 Představujeme řetězce	73
2.6 Čtení a zápis do souborů	74
2.7 Představujeme vektor.....	75
2.8 Shrnutí.....	79
2.9 Cvičení	80
3. Jazyk C v C++.....	81
3.1 Vytváříme funkce	81
Návratové hodnoty funkcí.....	83
Použití knihoven funkcí v C	84
Vytváření vlastních knihoven.....	84
3.2 Řídící příkazy	85
Pravda a nepravda	85
if – else	85
while	86
do – while	87
for	88
Klíčová slova break a continue.....	88
switch.....	90
Vhodné a nevhodné použití příkazu goto	91
Rekurze	92

3.3 Úvod do práce s operátory	93
Pořadí vyhodnocování operátorů	93
Inkrementační a dekrementační operátory	93
3.4 Úvod do datových typů	94
Základní vestavěné typy	94
bool, true a false	95
Specifikátory	96
Úvod do použití ukazatelů	97
Přístup k vnějším objektům	100
Úvod do použití odkazů v C++	102
Ukazatelé a odkazy jako modifikátory	103
3.5 Viditelnost proměnných	105
Definice proměnných za běhu	106
3.6 Alokace paměti	107
Globální proměnné	107
Lokální proměnné	109
Statické proměnné	109
Specifikátor extern	110
Konstanty	112
Specifikátor volatile	113
3.7 Operátory a jejich použití	114
Přiřazení	114
Matematické operátory	114
Relační operátory	116
Logické operátory	116
Bitové operátory	117
Operátory bitového posunu	117
Unární operátory	120
Ternární operátor	120
Čárkový operátor	121
Na co si při použití operátorů dávat pozor	121
Operátory přetypování	122
Explicitní přetypování v C++	123
sizeof – operátor sám o sobě	126
Klíčové slovo asm	127
Explicitní operátory	127
3.8 Tvorba vlastních datových typů	128
Přejmenování typů příkazem typedef	128
Sdružování proměnných do struktur	129
Zpřehlednění programu pomocí typu enum	131
Úspory paměti použitím typu union	133
Pole	134
3.9 Typy pro ladění	143
Příznaky pro řízení ladění	143
Převod proměnných a výrazů na řetězce	145
Makro assert() v jazyce C	146
3.10 Adresy funkcí	146
Definice ukazatele na funkci	147
Složitější deklarace a definice	147

Použití ukazatelů na funkce.....	148
Pole ukazatelů na funkce	149
3.11 Make: řízení odděleného překladu	150
Funkce sestavovacího programu	151
Sestavovací soubory v této knize	153
Vzorový sestavovací soubor.....	154
3.12 Shrnutí.....	155
3.13 Cvičení	156
4. Datové abstrakce	160
4.1 Jednoduchá knihovna podobná knihovnám v C	160
Dynamická alokace paměti.....	164
Špatný odhad	167
4.2 V čem je chyba?	168
4.3 Jednoduchý objekt	168
4.4 Co je to objekt?	174
4.5 Abstraktní datové typy	174
4.6 Podrobnosti o objektech	175
4.7 Konvence pro psaní hlavičkových souborů.....	176
Význam hlavičkových souborů	177
Potíže s vícenásobnou deklarací.....	178
Direktivy preprocesoru #define, #ifdef a #endif	179
Standardní tvar hlavičkových souborů.....	179
Jmenné prostory v hlavičkových souborech	180
Použití hlavičkových souborů v projektech	180
4.8 Vnořené struktury	181
Globální příslušnost.....	184
4.9 Shrnutí.....	185
4.10 Cvičení.....	185
5. Ukrývání implementace.....	188
5.1 Nastavení hranic	188
5.2 Správa přístupu v C++	188
Klíčové slovo protected	190
5.3 Klíčové slovo friend.....	190
5.4 Hníždění deklarací friend.....	192
Je to čisté?	194
5.5 Uspořádání objektu	195
5.6 Třída	195
Úprava příkladu Stash s využitím řízeného přístupu	198
Úprava příkladu Stack s využitím řízeného přístupu	198
5.7 Třídy typu Handle.....	199
Skrývání implementace	199
Zamezení opakovanému překladu	200
5.8 Shrnutí.....	202
5.9 Cvičení	202

6. Inicializace a čištění paměti	204
6.1 Zajištění inicializace pomocí konstrukturu	204
6.2 Čištění paměti pomocí destrukturu	206
6.3 Jak se zbavit bloků definic	208
Cykly for	209
Alokace paměti	210
6.4 Třída Stash s konstruktory a destruktory	211
6.5 Třída Stack s konstruktory a destruktory	214
6.6 Inicializace agregovaných typů	216
6.7 Implicitní konstruktory	218
6.8 Shrnutí	219
6.9 Cvičení	220
7. Přetěžování funkcí a implicitní parametry	221
7.1 Další úprava názvů	222
Přetěžování u návratových hodnot	222
Typově bezpečné spojování	223
7.2 Příklad na přetížení	224
7.3 Typ union	226
7.4 Implicitní parametry	229
Parametry pro alokaci místa (placeholder)	230
7.5 Přetěžování nebo implicitní parametry?	231
7.6 Shrnutí	235
7.7 Cvičení	235
8. Konstanty	237
8.1 Nahrazení hodnoty	237
Klíčové slovo const v hlavičkových souborech	238
Konstanty a bezpečnost	238
Konstanty a strukturované typy	239
Odlišnosti od jazyka C	240
8.2 Ukazatele	241
Ukazatel na konstantu	241
Konstantní ukazatel	242
Formátování	243
8.3 Přiřazení a typová kontrola	243
Pole znaků s literály	244
8.4 Funkční parametry a návratové hodnoty	244
Předávání konstant hodnotou	244
Vrácení konstant hodnotou	245
Dočasné objekty	246
8.5 Předávání a vrácení adres	247
Standardní předávání parametrů	249
8.6 Třídy	250
Konstanta ve třídě	250
Inicializační seznam konstrukturu	251
Konstruktory pro vestavěné typy	251

Konstanty třídy použité během překladu	253
Řešení pomocí výčtového typu ve starších kódech.....	254
Konstantní objekty a funkční členy	255
Záměna: bitové vs. logické konstanty.....	257
Práce s ROM.....	259
8.7 Klíčové slovo volatile	259
8.8 Shrnutí.....	261
8.9 Cvičení	261
9. Inline funkce.....	264
9.1 Úskalí preprocesoru	264
Makra a přístup	267
9.2 Inline funkce	267
Inline funkce uvnitř tříd	268
Přístupové funkce.....	269
9.3 Třídy Stash a Stack a inline funkce	273
9.4 Inline funkce a překladač	277
Omezení	277
Dopředné odkazy	278
Skruté akce uvnitř konstruktorů a destruktorů	279
9.5 Trocha pořádku	280
9.6 Další vlastnosti preprocesoru.....	281
Vkládání značek	281
9.7 Vylepšená kontrola chyb.....	282
9.8 Shrnutí.....	284
9.9 Cvičení	285
10. Správa jmen	287
10.1 Statické prvky z jazyka C.....	287
Statické proměnné uvnitř funkcí	287
Řízení vazby.....	291
Další paměťové třídy	293
10.2 Prostory jmen	293
Vytvoření prostoru jmen	293
Používání jmenného prostoru.....	295
Použití prostorů jmen.....	299
10.3 Statické prvky tříd v C++	300
Definice statických datových členů	300
Vnošené a lokální třídy.....	303
Statické členské funkce	304
10.4 Závislost statických inicializací.....	307
Co si počít?.....	308
10.5 Alternativní určení vazby.....	314
10.6 Shrnutí.....	314
10.7 Cvičení	315

11. Reference a kopírovací konstruktor	319
11.1 Ukazatele v C++.....	319
11.2 Reference v C++.....	319
Reference ve funkcích.....	320
Pravidla předávání parametrů.....	322
11.3 Kopírovací konstruktor.....	323
Předávání a návrat parametru hodnotou.....	323
Vytvoření objektu kopírovacím konstruktorem.....	328
Implicitní kopírovací konstruktor.....	332
Alternativy ke kopírovacím konstruktorům.....	334
11.4 Ukazatele na členské prvky.....	335
Funkce.....	337
11.5 Shrnutí.....	339
11.6 Cvičení.....	340
12. Přetěžování operátorů	343
12.1 Proč operátory přetěžovat.....	343
12.2 Syntax.....	344
12.3 Přetížitelné operátory.....	345
Unární operátory.....	345
Inkrementace a dekrementace.....	348
Binární operátory.....	349
Parametry a návratové hodnoty.....	358
Neobvyklé operátory.....	360
Operátory, které není možné přetížit.....	367
12.4 Nečlenské operátory.....	368
Základní pravidla.....	370
12.5 Přetížení operátoru přiřazení.....	370
Chování operátoru =.....	371
12.6 Automatický převod typu.....	380
Převod konstruktoru.....	380
Převod operátorů.....	381
Příklad na převod typu.....	383
Pasti související s automatickým převodem typů.....	384
12.7 Shrnutí.....	387
12.8 Cvičení.....	387
13. Dynamické vytváření objektů	390
13.1 Vytváření objektů.....	390
Přístup jazyka C k paměti typu hromada.....	391
Operátor new.....	392
Operátor delete.....	393
Jednoduchý příklad.....	393
Režie správce paměti.....	394
13.2 Úprava předchozích příkladů.....	395
Delete void* je pravděpodobně chyba.....	395
Jak je to s odpovědností za smazání u ukazatelů.....	396

Třída Stash pro uložení ukazatelů	397
Test	399
13.3 new a delete pro pole	401
Jak vytvořit ukazatel tak, aby vypadal spíš jako pole	402
13.4 Pokud dojde paměť'	402
13.5 Přetížení new a delete	403
Přetížení globálního new a delete	404
Přetížení new a delete pro třídu	405
Přetížení new a delete pro pole	408
Volání konstruktoru	410
new a delete s umístěním	411
13.6 Shrnutí	413
13.7 Cvičení	413
14. Dědičnost a kompozice	415
14.1 Kompozice	415
14.2 Dědičnost	416
14.3 Inicializační seznam konstruktoru	418
Inicializace vnořených objektů	418
Vestavěné typy v inicializačním seznamu	419
14.4 Kombinace dědičnosti a kompozice	420
Automatické volání destrukturu	421
Pořadí volání konstruktorů a destrukturu	421
14.5 Skrývání názvů	423
14.6 Funkce, které se automaticky nedědí	426
Dědičnost a statické členské funkce	429
14.7 Volba mezi kompozicí a dědičností	429
Podtypy	431
Dědění private	433
Zveřejnění soukromých zděděných prvků	433
14.8 Specifikátor protected	434
Dědičnost se specifikátorem protected	435
14.9 Dědičnost a přetěžování operátorů	435
14.10 Vícenásobná dědičnost	437
14.11 Inkrementální vývoj	437
14.12 Přetypování směrem vzhůru	438
Proč „Přetypování směrem vzhůru?“	439
Přetypování směrem vzhůru a kopírovací konstruktor	439
14.13 Volba mezi kompozicí a dědičností – ještě jednou	442
Přetypování ukazatelů a odkazů	443
Kříže	443
14.14 Shrnutí	443
14.15 Cvičení	444
15. Polymorfizmus a virtuální metody	447
15.1 Vývoj programování v C++	447
15.2 Typování zdola nahoru	448

15.3 Problém	449
Vazba funkčního volání	449
15.4 Virtuální funkce	450
Rozšiřitelnost	451
15.5 Jak C++ implementuje dynamickou vazbu	453
Uložení informace o typu	454
Znázornění virtuálních funkcí	455
Co zůstává skryté	456
Umístění ukazatele vpointer	457
Objekty se liší	458
15.6 Proč virtuální funkce?	458
15.7 Abstraktní bazové třídy a čisté virtuální funkce	459
Definice čistých virtuálních metod	463
15.8 Tabulka virtuálních funkcí a dědičnost	464
Redukce objektů	465
15.9 Přetěžování a potlačování metod	467
Odlišný návratový typ	469
15.10 Virtuální funkce a konstruktory	471
Pořadí volání konstruktorů	471
Chování virtuálních funkcí uvnitř konstruktorů	472
15.11 Destruktory a virtuální destruktory	473
Čisté virtuální destruktory	474
Virtuální funkce uvnitř destruktorů	476
Vytváření objektové hierarchie	477
15.12 Přetěžování operátorů	479
15.13 Typování shora dolů	482
15.14 Shrnutí	484
15.15 Cvičení	485
16. Úvod do šablon	488
16.1 Kontejnery	488
Potřeba kontejnerů	490
16.2 Základní informace o šablonách	490
Řešení pomocí šablon	492
16.3 Syntax šablony	493
Definice ne-inline funkcí	494
Hlavičkové soubory	495
Zásobník IntStack jako šablona	495
Konstanty v šablonách	497
16.4 Třídy Stack a Stash jako šablony	498
Ukazatel Stash a využití šablony	500
16.5 Zapínání a vypínání vlastnictví	505
16.6 Vlastnictví objektů „hodnotou“	507
16.7 Zavedení iterátorů	509
Zásobník s iterátory	516
PStash s iterátory	519
16.8 Proč iterátory?	524

Šablony funkcí	526
16.9 Shrnutí.....	527
16.10 Cvičení	528
Štábní kultura	531
Obecná pravidla	531
Názvy souborů	532
Počáteční a koncové komentářové značky	532
Kulaté závorky, složené závorky a odsazení.....	533
Názvy identifikátorů	535
Pořadí připojování hlavičkových souborů	536
Ochrana připojení v hlavičkových souborech	536
Použití jmenného prostoru	537
Použití require() a assure().....	537
Rady programátorům	538
Doporučená literatura	547
Jazyk C.....	547
Jazyk C++ všeobecně.....	547
Můj vlastní seznam knih	548
Hluboká a temná zákoutí.....	549
Analýza a návrh.....	549
Rejstřík	552