

Obsah

Předmluva	3
Obsah	5
1 Jednoduché programy	15
1.1 První program	15
1.1.1 Zdrojový program	15
1.1.2 Překlad, sestavení a spuštění	16
1.1.3 Co jsme to naprogramovali	16
1.2 Druhý program	21
1.2.1 Proměnné	22
1.2.2 Vstupy a výstupy	22
1.2.3 Funkce faktorial()	22
1.2.4 Organizace programu	27
1.2.5 Reálná čísla	27
1.2.6 Externí proměnné	28
1.2.7 Překlad programu složeného z více souborů	29
1.3 Třetí program	31
1.3.1 Deklarace třídy	32
1.3.2 Organizace programu	35
1.3.3 Třída string	35
2 Komplexní čísla	38
2.1 Implicitní hodnoty parametrů	38
2.2 Inicializační část konstruktoru	40
2.3 Přetěžování operátorů	40
2.4 Volání konstruktoru	43
2.5 Vložené funkce	43
2.6 Vstup a výstup komplexních čísel	44
3 Ukazatele	46
3.1 Předávání parametrů při volání funkce	46
3.1.1 Předávání ukazatelů	47
3.1.2 Předávání parametrů odkazem (jen C++)	47
3.2 Seznam	48
3.2.1 Prvek seznamu	48
3.2.2 Vlastní seznam	49

4	Další možnosti	58
4.1	Parametrizace	58
4.1.1	Deklarace typedef	58
4.1.2	Další možnosti	60
4.1.3	Šablony	60
4.2	Ošetření chyb	66
4.2.1	Výjimky	67
5	Začínáme	70
5.1	Popis jazyka C++	70
5.2	Množina znaků	70
5.3	Identifikátor	72
5.4	Zápis programu	73
5.5	Lexikální konvence	74
5.6	Komentář	74
5.7	Klíčová slova	75
5.8	Deklarace a definice	76
5.9	Program	77
6	Příkazy	78
6.1	Výrazový příkaz	78
6.2	Deklarace	79
6.3	Blok neboli složený příkaz	79
6.3.1	Prázdný příkaz	80
6.4	Podmíněné příkazy (větvení algoritmu)	81
6.4.1	Příkaz if	81
6.4.2	Příkaz switch (přepínač)	83
6.5	Cykly	86
6.5.1	Příkaz while	86
6.5.2	Příkaz do-while	86
6.5.3	Příkaz for	87
6.6	Příkazy způsobující přenos řízení	89
6.6.1	Příkaz break	89
6.6.2	Příkaz return	91
6.6.3	Příkaz skoku, návěští	92
6.6.4	Skoky v C++	94
6.7	Příkaz (deklarace) asm	95
7	Základní datové typy	96
7.1	Celočíselné typy	96
7.1.1	Celá čísla	96
7.1.2	Znakové typy	100
7.1.3	Převod mezi typy char a wchar_t	102
7.1.4	Logické hodnoty	102
7.1.5	Alternativní přístup k celým číslům v C99	102
7.2	Reálná čísla	103
7.3	void	106

7.4	Komplexní čísla (jen C99)	106
7.4.1	Typy pro komplexní čísla	106
7.4.2	Makra pro komplexní čísla	107
7.4.3	Matematické funkce	107
7.4.4	Konverze	107
8	Pole, ukazatele a dynamické proměnné	108
8.1	Pole	108
8.1.1	Jednorozměrná pole	108
8.1.2	Znakové řetězce	111
8.1.3	Vícerozměrná pole	112
8.2	Ukazatele	114
8.2.1	Deklarace	114
8.2.2	Dereferencování ukazatelů	115
8.2.3	Ukazatel bez doménového typu	116
8.2.4	Ukazatel nikam	116
8.2.5	Ukazatel jako logická hodnota	116
8.2.6	Ukazatele na konstanty, konstantní ukazatele	117
8.2.7	Restringované ukazatele v C99	118
8.3	Dynamické proměnné	119
8.3.1	Operátor new	120
8.3.2	Operátor delete	122
8.3.3	Přístupový vektor	124
8.3.4	Dynamické proměnné v jazyce C	124
8.4	Adresová aritmetika	125
8.4.1	Porovnávání ukazatelů	125
8.4.2	Číslo + ukazatel	126
8.4.3	Rozdíl dvou ukazatelů	127
8.5	Vztah polí a ukazatelů	127
8.6	Ukazatele na funkce	128
8.7	Reference	130
8.7.1	Použití referencí	130
8.7.2	Konstantní reference	131
8.7.3	Funkce, které vracejí výsledek odkazem	131
8.8	Příklad: Eratosthenovo síto	132
9	Výčtové typy, struktury a unie	135
9.1	Výčtové typy	135
9.1.1	Deklarace výčtového typu	135
9.1.2	Hodnoty výčtových konstant	137
9.1.3	Typová kontrola	137
9.1.4	Rozsah výčtového typu	138
9.1.5	Bitové příznaky	138
9.1.6	Přetěžování operátorů pro výčtové typy	139
9.2	Struktury a unie	140
9.2.1	Struktury	140
9.2.2	Unie	143

9.2.3	Anonymní unie	145
9.2.4	Přetěžování operátorů pro struktury a unie	146
10	Operátory, výrazy	147
10.1	l-hodnota a r-hodnota	147
10.2	Konverze	147
10.2.1	Celočíselná a reálná rozšíření	148
10.2.2	Obvyklé aritmetické konverze	149
10.2.3	Konverze ukazatelů	149
10.2.4	Standardní konverze	150
10.2.5	Konverze v C99	150
10.3	Přehled operátorů	151
10.3.1	Přístupové operátory	152
10.3.2	Přetypování	157
10.3.3	Správa paměti	159
10.3.4	Získání adresy: operátor &	160
10.3.5	Určení velikosti a typu	161
10.3.6	Aritmetické operátory	162
10.3.7	Relační operátory	164
10.3.8	Logické operátory	165
10.3.9	Bitové operace	167
10.3.10	Operátory inkrementování a dekrementování ++ a --	168
10.3.11	Operátor čárka ,	169
10.3.12	Přiřazovací operátory =, += atd.	170
10.4	Konstantní výrazy	171
11	Proměnné, deklarace	172
11.1	Paměťové třídy	172
11.1.1	Automatické proměnné (paměťová třída auto)	172
11.1.2	Registrové proměnné (paměťová třída register)	172
11.1.3	Externí proměnné (paměťová třída extern)	173
11.1.4	Statické proměnné (paměťová třída static)	173
11.1.5	Měnitelné složky konstantních objektů (paměťová třída mutable)	175
11.2	Cv-kvalifikátory	175
11.2.1	Konstanty (modifikátor const)	175
11.2.2	Nestálé proměnné (modifikátor volatile)	175
11.3	Další modifikátory	176
11.4	Deklarace	176
11.4.1	Oč jde	177
11.4.2	Syntax deklarace	177
11.4.3	Implicitní int	179
11.5	Význam základních tvarů deklarátoru	180
11.6	Inicializace	181
11.7	Deklarace typu	182
11.7.1	typedef	182
11.7.2	Označení typu	183
11.8	Pracovní definice a neúplné deklarace (C90)	183

11.8.1	Pracovní definice	183
11.8.2	Neúplné deklarace	184
11.9	Platnost a viditelnost deklarace	184
11.9.1	Oblast platnosti	185
11.9.2	Rozdělení identifikátorů	186
11.9.3	Oblast viditelnosti identifikátoru	187
12	Funkce	189
12.1	Deklarace funkce	189
12.1.1	Deklarace funkce podle Kernighana a Ritchieho	190
12.1.2	Tělo funkce	191
12.2	Parametry funkce	191
12.2.1	Předávání parametrů hodnotou	191
12.2.2	Předávání ukazatelů	192
12.2.3	Předávání parametrů odkazem	192
12.2.4	Ukazatelem, nebo odkazem?	193
12.2.5	Výpustka	193
12.2.6	Implicitní hodnoty parametrů	194
12.2.7	Nepoužité parametry	196
12.3	Vracená hodnota	196
12.3.1	Funkce, které vracejí výsledek odkazem	197
12.4	Různé modifikátory	198
12.4.1	Specifikace paměťové třídy	198
12.4.2	Vložené funkce (inline)	198
12.4.3	Volací konvence	199
12.5	Přetěžování funkcí	200
12.5.1	Proč vlastně přetěžovat funkce?	202
12.6	Spolupráce C s C++	202
12.6.1	Vnitřní jména	202
12.6.2	Specifikace sestavování	203
12.7	Funkce main()	204
12.7.1	Omezení kladená na funkci main()	204
12.7.2	Parametry funkce main()	204
12.7.3	Běh programu	206
12.8	Prototyp funkce	207
12.8.1	Informativní deklarace funkce v jazyce C	207
12.9	Identifikátor <code>_func_</code> (jen C99)	208
13	Preprocesor	209
13.1	Prázdná direktiva <code>#</code>	209
13.2	Vložení jednoho souboru do jiného: direktiva <code>#include</code>	210
13.3	Definice maker: direktiva <code>#define</code>	211
13.3.1	Platnost direktivy <code>#define</code>	211
13.3.2	Makra bez parametrů	211
13.3.3	Makra s parametry	211
13.3.4	Předdefinovaná makra	212
13.3.5	Makro <code>assert()</code>	212

13.3.6	Příklady	214
13.4	Zrušení makra: direktiva <code>#undef</code>	218
13.5	Podmíněný překlad: direktivy <code>#if</code> apod.	218
13.5.1	Operátor <code>defined</code>	219
13.5.2	Direktivy <code>#ifdef</code> a <code>#ifndef</code>	220
13.6	Změna číslování řádků souboru: direktiva <code>#line</code>	221
13.7	Vyvolání chyby při překladu: direktiva <code>#error</code>	222
13.7.1	Varování: direktiva <code>#warn</code> (C99)	222
13.8	Direktiva závislá na implementaci: <code>#pragma</code>	222
14	Objektové typy	224
14.1	Třída	224
14.1.1	Deklarace objektového typu	224
14.1.2	Datové složky	226
14.1.3	Metody	226
14.1.4	Přístupová práva	229
14.1.5	Statické složky tříd	231
14.1.6	Deklarace typu v těle třídy	233
14.2	Inicializace, konstruktory a destruktory	234
14.2.1	Konstruktory a destruktory	235
14.2.2	Konstruktory	236
14.2.3	Destruktory	244
14.3	Odvozené třídy: předkové, potomci, dědění	247
14.3.1	Nevirtuální dědění	248
14.3.2	Třída je oblast	251
14.3.3	Virtuální dědění	251
14.3.4	Potomek může zastoupit předka	253
14.3.5	Konstruktory, destruktory a dědění	255
14.4	Polymorfismus (virtuální metody)	258
14.4.1	Časná a pozdní vazba	259
14.4.2	Abstraktní třídy a čistě virtuální metody	261
14.4.3	Implementace polymorfismu	262
14.5	Třídní ukazatele	263
14.6	Třídy, struktury a unie	264
14.6.1	Struktury a třídy	264
14.6.2	Unie	264
15	Přetěžování operátorů	265
15.1	Základní pravidla	265
15.1.1	Operátorové funkce	266
15.2	Skupina 4 (ostatní operátory)	266
15.2.1	Unární operátory	266
15.2.2	Binární operátory	269
15.3	Skupina 2 (operátory přetěžovatelné jen jako metody)	272
15.3.1	Operátor indexování <code>[]</code>	272
15.3.2	Operátor volání funkce <code>()</code>	275
15.3.3	Přiřazovací operátor <code>=</code>	276

15.3.4	Operátor nepřímého přístupu ->	278
15.3.5	Operátor přetypování	279
15.4	Skupina 3 (operátory new a delete)	281
15.4.1	Globální operátory new a delete	281
15.4.2	Operátory new a delete jako metody objektových typů	284
15.4.3	Delete s dodatečnými parametry	287
16	Šablony	288
16.1	Deklarace šablony	288
16.1.1	Parametry šablony	288
16.2	Objektové typy	290
16.2.1	Metody	291
16.2.2	Statické atributy	292
16.2.3	Specializace	292
16.2.4	Vnořené šablony	293
16.2.5	Typ jako složka typu	294
16.3	Funkce	295
16.3.1	Specializace	296
16.3.2	Přetěžování	296
16.3.3	Spřátelené funkce	298
16.4	Explicitní generování specializací	299
16.5	Vazba jmen	300
16.6	Parciální specializace šablon objektových typů	301
16.7	Organizace programu	303
16.7.1	Současné překladače	304
16.8	Různá omezení	305
17	Výjimky	306
17.1	Výjimky v C++	307
17.1.1	Základní pojmy	307
17.1.2	Syntax výjimek	307
17.1.3	Když vznikne výjimka	308
17.1.4	Obsluha	313
17.1.5	Výjimky a funkce	316
17.1.6	Výjimky a alokace paměti	320
17.1.7	Výjimky v obsluze	322
17.1.8	Výjimkový objekt	322
17.1.9	Neošetřené a neočekávané výjimky	323
17.1.10	Standardní výjimky	323
17.2	Strukturované výjimky v jazyce C	325
17.2.1	První přiblížení	326
17.2.2	Přenos informací o výjimce	326
17.2.3	Syntax strukturovaných výjimek	327
17.2.4	Vznik strukturovaných výjimek	329
17.2.5	Filtr	330
17.2.6	Koncovka bloku	332
17.2.7	Neošetřené výjimky	336

17.3	Strukturované výjimky a C++	337
18	Dynamická identifikace typů, přetypování	338
18.1	Dynamická identifikace typů	338
18.1.1	Operátor typeid	338
18.2	Přetypování	341
18.2.1	Nové operátory	341
19	Prostory jmen	350
19.1	O co vlastně jde	350
19.2	Deklarace prostoru jmen	350
19.2.1	Deklarace po částech	353
19.2.2	Anonymní prostor jmen	353
19.3	Zpřístupnění prostoru jmen	354
19.3.1	Alias (přezdívka) prostoru jmen	354
19.3.2	using	355
19.4	Prostory jmen a třídy	356
19.5	Vyhledávání funkcí a operátorů v závislosti na parametrech	357
20	Vstupy a výstupy	359
20.1	Soubory a datové proudy	359
20.1.1	Binární a textové soubory	359
20.1.2	Standardní datové proudy	360
20.1.3	Zpracování souboru	360
20.2	Objektové datové proudy v C++	361
20.2.1	Hierarchie tříd a šablon	361
20.2.2	Formátované vstupy a výstupy	369
20.2.3	Výjimky	373
20.2.4	Znakově orientované vstupy a výstupy	374
20.2.5	Rozšiřování možností vstupů a výstupů	376
20.3	Standardní vstupy a výstupy jazyka C (stdio.h)	380
20.3.1	Otevření a zavření souboru	380
20.3.2	Formátovaný zápis do souboru	382
20.3.3	Makra pro formátování celočíselných typů (jen C99)	389
20.3.4	Funkce pro znakový výstup	390
20.3.5	Formátovaný vstup ze souboru	390
20.3.6	Makra pro čtení rozšířených celočíselných typů (jen C99)	395
20.3.7	Funkce pro znakový vstup	395
20.3.8	Neformátované čtení a zápis	397
20.3.9	Standardní vstup a výstup	398
20.3.10	Paměťové proudy	399
20.3.11	Vstup a výstup s výpustkou	400
20.3.12	Některé další funkce	401
20.3.13	Formátované operace se širokými znaky	401
20.4	Práce se soubory pomocí identifikačních čísel (<io.h>)	402
20.4.1	Otevření a zavření souboru	402
20.4.2	Čtení a zápis	404

20.5 Konzolové vstupy a výstupy (<conio.h>)	405
20.5.1 Základní vstupy a výstupy	405
21 Knihovny	408
21.1 Hlavičkové soubory	408
21.2 Lokální nastavení	408
21.3 Práce se znaky a znakovými řetězci	410
21.3.1 Druh znaku	411
21.3.2 Pole znaků	411
21.4 Práce s pamětí	412
21.5 Zpracování chyb a ladění	413
21.5.1 Globální proměnná errno	413
21.5.2 Makro assert	413
21.5.3 Dlouhý skok	414
21.5.4 Zpracování signálů	414
21.6 Čísla a počítání	414
21.6.1 Matematické funkce	414
21.6.2 Informace o číselných typech	414
21.6.3 Konverze čísel na řetězce a naopak	415
21.6.4 Náhodná čísla	415
21.6.5 Komplexní čísla	415
21.7 Běh programu	415
21.8 Datum a čas	416
21.9 Třídění a vyhledávání	416
21.10 Knihovna kontejnerů a algoritmů	417
21.10.1 Kontejnery a iterátory	417
21.10.2 Druhy iterátorů	418
21.10.3 Platnost iterátorů	419
21.10.4 Kontejnery	419
21.10.5 Generické algoritmy	423
21.10.6 Algoritmy, které mění prvky posloupností	423
21.10.7 Algoritmy pro setříděné kontejnery	424
21.10.8 Některé další algoritmy	425
21.11 Příklad: filtr SORT	425
21.11.1 Přepínač -n	427
21.11.2 Řazení podle české abecedy	429
22 C++0x: Očekávaná rozšíření	432
22.1 Knihovna kontejnerů	432
22.1.1 Netříděné asociativní kontejnery	432
22.1.2 Šablona array<>	433
22.1.3 Šablona tuple<>	434
22.2 Automatické ukazatele	434
22.3 Regulární výrazy	435
22.3.1 Podporované gramatiky regulárních výrazů	435
22.4 Matematické výpočty	436
22.4.1 Speciální funkce	436

22.4.2	Funkce komplexní proměnné	436
22.4.3	Řízení výpočetního prostředí	436
22.5	Pseudonáhodná čísla	437
22.5.1	Generátory rovnoměrně rozdělených pseudonáhodných čísel	437
22.5.2	Generátory zadaných rozdělení	437
22.5.3	„Stroje“	438
22.5.4	Třída variate_generator<>	438
22.6	Kompatibilita s C99	438
22.6.1	Celočíselné datové typy	439
22.6.2	Logické hodnoty	439
22.6.3	Nové specifikátory formátu	439
Literatura		440
Rejstřík		442