

Obsah

Úvod	9
1 Zdrojový text programu	11
1.1 Funkce v jazyce C++, direktiva <i>#include</i>	12
1.2 Direktiva <i>#define</i>	13
1.3 Základní datové typy, konstanty	15
1.3.1 Celočíselné typy	15
1.3.2 Znaky	16
1.3.3 Logické hodnoty	17
1.3.4 Řetězce	17
1.3.5 Čísla v pohyblivé řádové čárce	19
1.4 Identifikátory a deklarace	20
1.4.1 Deklarace datového objektu s konstantní hodnotou – <i>const</i>	21
1.4.2 Deklarace sdílené proměnné – <i>volatile</i> ^{††}	21
1.5 Operátory	21
1.5.1 Operátory <i>() [] . -> .* ->* ::</i>	23
1.5.2 Unární operátory <i>! ~ ++ -- + - * & sizeof new delete</i>	24
1.5.3 Aritmetické binární operátory <i>+ - * / %</i>	26
1.5.4 Operátory bitového posuvu <i><< >></i>	26
1.5.5 Operátory srovnání <i>< <= > >= == !=</i>	27
1.5.6 Operátory <i>& ^ </i>	28
1.5.7 Logické operátory <i>&& </i>	28
1.5.8 Podmíněný výraz <i>?:</i>	29
1.5.9 Operátory přiřazení <i>= *= /= %= += -= <<= >>= &= ^=</i> <i> =, lvalue, rvalue</i>	29
1.5.10 Konverze hodnot	30
1.5.11 Operátor <i>,</i> (čárka)	32
1.5.12 Pořadí výpočtu operandů	33
2 Pointery (ukazatelé)	35
2.1 Operace s pointery, operátor dereference <i>*</i>	36
2.2 Konstantní pointery, pointery na konstantní hodnotu – <i>const</i>	38
3 Příkazy, složený příkaz	39
3.1 Příkaz – výraz	39
3.2 Složený příkaz (blok)	39
3.3 Podmíněný příkaz – <i>if</i>	40
3.4 Příkazy cyklu – <i>while, for</i>	40

3.5	Prázdný příkaz	42
3.6	Příkazy <i>continue</i> , <i>break</i>	42
3.7	Přepínač – <i>switch</i>	43
3.8	Návěští, příkaz skoku – <i>goto</i>	44
4	Deklarace pole a strukturovaného typu, výčtový typ	47
4.1	Deklarace odvozeného datového typu – <i>typedef</i>	47
4.2	Pole	48
4.2.1	Pole s jedním rozměrem	48
4.2.2	Pointery a jednorozměrná pole, operátor indexu <code>[]</code>	50
4.2.3	Vícerozměrná pole	51
4.2.4	Pointery a vícerozměrná pole	53
4.3	Strukturované datové typy	54
4.3.1	Datový typ <i>struct</i>	54
4.3.2	Datový typ <i>union</i>	57
4.4	Bitová pole ^{††}	58
4.5	Výčtový typ <i>enum</i>	59
5	Funkce a reference	61
5.1	Reference	61
5.1.1	Deklarace reference	62
5.2	Funkce	63
5.2.1	Funkční hodnota, příkaz <i>return</i>	64
5.2.2	Formální parametry funkce	64
5.2.3	Typ výsledku funkce	70
5.2.4	Implicitní hodnoty parametrů funkce	73
5.2.5	Polymorfní funkce	74
5.2.6	Prototyp funkce, deklaráce funkce	75
5.2.7	Funkce jako datový typ	77
5.2.8	Funkce jako formální parametr	78
5.2.9	Funkce s proměnným počtem parametrů [†]	79
5.2.10	Funkce <i>inline</i>	81
6	Vstupy a výstupy	83
6.1	Vstupy a výstupy pro standardní zařízení – <i>stdout</i> , <i>stdin</i> , <i>stderr</i>	83
6.1.1	Výstup na standardní zařízení – <i>printf</i>	83
6.1.2	Vstup ze standardního zařízení – <i>scanf</i>	86
6.1.3	Další funkce a makra výstupu na stand. zař. – <i>putchar</i> , <i>puts</i>	88
6.1.4	Další funkce a makra vstupu ze stand. zař. – <i>getchar</i> , <i>gets</i>	88
6.2	Vstupy a výstupy pro soubory	89
6.2.1	Otevření souboru – <i>fopen</i>	89
6.2.2	Formátový a neformátový vstup/výstup	90
6.2.3	Formátový výstup – <i>fprintf</i>	90
6.2.4	Neformátový výstup – <i>fwrite</i>	91
6.2.5	Další funkce a makra pro výstup – <i>fputc</i> , <i>putc</i> , <i>fputs</i> , <i>putw</i>	91
6.2.6	Formátový vstup – <i>fscanf</i>	92
6.2.7	Neformátový vstup – <i>fread</i>	92
6.2.8	Další funkce a makra pro vstup – <i>fgetc</i> , <i>getc</i> , <i>fgets</i> , <i>getw</i>	93

6.2.9	Test konce souboru – <i>feof</i>	93
6.2.10	Uzavření souboru – <i>fclose</i>	94
6.2.11	Ošetření chyb – <i>ferror</i>	94
6.2.12	Výpisy zpráv o chybě – <i>perror</i>	95
6.2.13	Přímý přístup k souboru – <i>ftell, fseek, fgetpos, fsetpos</i>	95
6.2.14	Přejmenování (přesun) a zrušení souboru – <i>rename, remove</i>	96
6.3	Funkce vstupu/výstupu v paměti – <i>scanf, printf</i>	97
7	Základní struktura programu, změna typu (cast)	99
7.1	Specifikace uložení v paměti – <i>auto, extern, static, register</i>	101
7.2	Blízké a vzdálené pointery, paměťové modely – <i>near, far, huge</i>	103
7.3	Změna typu – (cast)	106
8	Standardní funkce, pole pointerů	111
8.1	Alokace a uvolnění paměti – <i>new, delete, <alloc.h>, <malloc.h></i>	111
8.1.1	Operátory <i>new</i> a <i>delete</i>	111
8.1.2	Alokace a uvolnění paměti – <i>malloc, calloc, free</i>	113
8.1.3	Realokace paměti – <i>realloc</i>	114
8.1.4	Zjištění velikosti volné paměti – <i>coreleft</i>	115
8.1.5	Test paměti – <i>heapcheck</i>	115
8.2	Funkce pro práci s řetězci – <i><string.h></i>	116
8.2.1	Kopírování řetězce – <i>strcpy, strncpy</i>	116
8.2.2	Spojení dvou řetězců – <i>strcat, fstrcat, strncat, fstrncat</i>	116
8.2.3	Délka řetězce – <i>strlen</i>	117
8.2.4	Kopírování řetězce – <i>stpcpy</i>	117
8.2.5	Alokace paměti a uložení řetězce – <i>strdup</i>	118
8.2.6	Srovnání dvou řetězců – <i>strcmp, strncmp, stricmp, strnicmp</i>	118
8.2.7	Vyhledání znaku v řetězci – <i>strchr, strrchr, strpbrk</i>	119
8.2.8	Vyhledání podřetězce v řetězci – <i>strstr</i>	120
8.2.9	Změna velikosti písmen v řetězci – <i>strupr, strlwr</i>	120
8.2.10	Nastavení obsahu řetězce – <i>strset, strnset</i>	121
8.3	Funkce pro práci s bloky paměti – <i><mem.h>, <memory.h></i>	121
8.3.1	Přenos (kopie) bloku paměti – <i>memcpy, memmove</i>	121
8.3.2	Srovnání bloků paměti – <i>memcmp, memicmp</i>	122
8.3.3	Vyhledání znaku v paměti – <i>memchr</i>	122
8.3.4	Nastavení obsahu bloku paměti – <i>memset</i>	123
8.4	Funkce a makra pro práci se znaky – <i><ctype.h></i>	123
8.4.1	Makra pro test znaků – <i>is...</i>	123
8.4.2	Úpravy znaku – <i>tolower, toupper, toascii</i>	124
8.5	Pole pointerů	124
9	Třídy, objekty	127
9.1	Deklarace třídy – <i>struct, class</i>	128
9.2	Přístup k datovým členům a funkcím třídy	129
9.3	Konstruktor a destruktory	130
9.4	Definice členských funkcí, funkce <i>inline</i>	133
9.5	Statické datové členy – <i>static</i>	135
9.6	Statické funkce – <i>static</i>	136

9.7	Spřátelené třídy – <i>friend</i>	137
9.8	Spřátelené funkce – <i>friend</i>	138
9.9	Konstantní funkce – <i>const</i>	139
9.10	Konstantní objekt – <i>const</i> †	139
9.11	Dočasné (temporary) objekty	140
9.12	Pole objektů	140
9.13	Objekty jako datové členy tříd	141
9.14	Pointery na členy – operátory <i>.* ->*</i> ††	142
9.15	Deklarace lokálních tříd †	143
9.16	Skrytí datových členů (encapsulation)	145
10	Dědění tříd	147
10.1	Dědění jedné třídy	147
10.1.1	Přístupové vlastnosti členů při dědění třídy	147
10.1.2	Konstruktory a destruktory	149
10.1.3	Operátor rozlišení a jeho použití – <i>::</i>	150
10.1.4	Virtuální funkce – <i>virtual</i>	152
10.1.5	Pointery na třídu, virtuální destruktory	156
10.2	Dědění více tříd	158
10.2.1	Konstruktory a destruktory	158
10.2.2	Vícenásobný přístup, virtuální dědění třídy – <i>virtual</i>	160
11	Streamy	165
11.1	Streamy pro standardní zařízení – <i>iostream.h</i>	165
11.1.1	Výstup na standardní zařízení – <i>cout</i>	166
11.1.2	Formátování výstupu – manipulátory	167
11.1.3	Vstup se standardního zařízení – <i>cin</i>	170
11.1.4	Výstup na zařízení <i>stderr</i> – <i>cerr, clog</i>	172
11.2	Streamy pro datové soubory – <i>fstream.h</i>	173
11.2.1	Otevření a uzavření souboru	173
11.2.2	Formátový vstup/výstup – <i><< >></i>	175
11.2.3	Neformátový vstup/výstup – <i>write, put, read, get, getline</i>	175
11.2.4	Test konce souboru – <i>eof, gcount</i>	177
11.2.5	Příznaky chyb při zpracování – <i>good, bad, fail, !</i>	178
11.2.6	Přímý přístup k souboru – <i>tellg, seekg</i>	179
11.2.7	Streamy pro vstup/výstup v paměti – <i>strstream.h</i>	180
11.2.8	Streamy pro vstup z paměti – <i>istrstream</i>	180
11.2.9	Streamy pro výstup do paměti – <i>ostrstream</i>	181
11.2.10	Streamy pro vstup/výstup v paměti – <i>strstream</i>	182
12	Překrytí (overloading) operátorů	183
12.1	Překrytí operátoru indexu <i>[]</i>	185
12.2	Překrytí operátoru volání funkce <i>()</i>	186
12.3	Překrytí operátoru výběru členu <i>-></i> ††	186
12.4	Překrytí operátoru inkrementace <i>++</i> a dekrementace <i>--</i>	187
12.5	Překrytí operátorů srovnání <i>== != < > <= >=</i>	188
12.6	Pointer na objekt – <i>this</i>	188
12.7	Překrytí operátoru přiřazení <i>=</i> , copy-konstruktor	189

12.8	Konverze typu <code>()</code>	193
12.9	Překrytí operátoru <code><<</code>	195
12.10	Překrytí operátoru <code>>></code>	195
12.11	Překrytí operátoru funkcemi, které nejsou členské	196
12.12	Překrytí operátorů <code>new</code> a <code>delete</code> ^{††}	199
13	Vzory (templates)	203
13.1	Vzory – funkce	210
14	Vyjímky (exceptions)	213
14.1	Vyjímky při alokaci paměti operátorem <code>new</code>	220
15	Návrh a koncepce programu	223
15.1	Direktivy podmíněného překladu – <code>#if</code> , <code>#ifdef</code> , <code>#ifndef</code>	223
15.2	Rozdělení programu na více částí – <code>project</code>	225
15.3	Formální parametry hlavní funkce – <code>main</code>	231
15.4	Návratová hodnota hlavní funkce – <code>main</code>	231
15.5	Funkce ukončení programu – <code>exit</code> , <code>abort</code>	232
15.6	Deklarace <code>asm</code>	232
15.7	Makro <code>assert</code>	233
15.8	Spojení modulů přeložených z jazyků C a C++ – <code>extern "C"</code>	233
15.9	Prázdňá direktiva <code>#</code> , operátor <code>##</code> [†]	234
15.10	Standardně definovaná makra	235
15.11	Direktivy <code>#error</code> , <code>#pragma</code> [†]	235
16	Nové prvky jazyka *	237
16.1	Třída pro uložení řetězců – <code>string</code>	237
16.2	Novější způsoby změny typu	241
16.2.1	Změna typu <code>const/volatile</code> – <code>const_cast</code>	241
16.2.2	Změna typu pointeru a reference na třídu – <code>dynamic_cast</code>	242
16.2.3	Změna typu – <code>static_cast</code> , <code>reinterpret_cast</code>	244
16.3	Identifikace datového typu – <code>typeid</code>	244
16.4	Vymezení rozsahu platnosti – <code>namespace</code>	246