

Obsah

O pastech a propastech	11
1. Zahřívací kolo.....	15
1.1 Příkazy	15
Příkaz switch.....	15
Zatoulané else.....	16
1.2 Operátory	17
Rovnost není přiřazení.....	17
&& není &, není 	18
Ne každý operátor má takovou prioritu, jakou bychom právě potřebovali	19
Matematická zkratka v Céčku?	20
Není dělení jako dělení	20
Asociativita operátorů.....	21
Pořadí, ve kterém se vyhodnocují operandy	21
Co se změní dříve?.....	22
1.3 Důsledky lexikální konvence	23
1.4 Syntax	24
Jen středník.....	24
1.5 Funkce	26
Volání funkce	26
Prototyp? K čemu to?	27
1.6 Sestavování a proměnné	27
Různé tváře jedné proměnné.....	28
Dvě proměnné se stejným jménem	29
1.7 Registrové pseudoproměnné.....	29
2. Makra	31
2.1 Jedna závorka, žádná závorka	31
2.2 Jednou stačí.....	32
2.3 Makra mohou být zatraceně dlouhá.....	32
2.4 Může makro nahradit deklaraci typu?	33
2.5 Středník v makru	33
2.6 Makro jako příkaz: <code>assert</code>	34
2.7 Aha, mezera.....	38
2.8 Preprocesor pracuje s položkami.....	39
2.9 Nedělejme z Céčka Pascal	39
3. Pole a ukazatele	41
3.1 Pole nejsou ukazatele a ukazatele nejsou pole	41
Ukazatele	41
Pole.....	41
Pole podrobněji.....	42
Jednorozměrná pole.....	42

Vícerozměrná pole.....	43
Ukazatel na ukazatel.....	44
Co je špatně v úvodním příkladu.....	45
Jak to mělo být správně.....	46
Jednou ukazatel, podruhé pole.....	46
U funkcí je to jedno.....	47
3.2 Pole.....	47
Počítáme prvky pole.....	47
Ukazatel na prvek za posledním.....	48
3.3 Ukazatele.....	48
Toulavé ukazatele.....	48
Ukazatel na objekt, který již neexistuje.....	49
4. Funkce v C++	53
4.1 Předávání parametrů odkazem.....	53
Předávání ukazatelů.....	53
Pozor na pole.....	53
Předávání odkazem v C++.....	54
Kdy je předávání odkazem nezbytné.....	56
Konstantní reference.....	57
4.2 Výsledek.....	58
Ukazatel na neexistující objekt.....	58
Vracíme výsledek odkazem.....	58
Odkaz na neexistující objekt.....	60
Dereferencovaný ukazatel.....	60
4.3 Standardní konverze.....	61
4.4 Výpustka.....	62
4.5 Funkce podle Kerninghana a Ritchieho.....	63
4.6 Pryč s konstantou NULL.....	63
4.7 Kterou funkci vlastně voláme?.....	64
Rozlišování přetížených funkcí.....	65
Standardní konverze nemusejí být přenositelné.....	69
5. Vstupy a výstupy	70
5.1 Nejběžnější chyby na úvod.....	70
5.2 Čtení souboru.....	72
Binární soubor.....	72
Textový soubor.....	72
Čtení řetězců.....	74
Znaky a jiná data.....	75
getchar() vrací int.....	76
5.3 Aktualizace souboru.....	78
5.4 Vlastní výstupní operátor.....	80
Šířka pole.....	80
Definice vstupních a výstupních operátorů.....	81
5.5 Paměťové proudy.....	82
5.6 Proč nepíše?.....	83
5.7 Textové a binární soubory.....	84

Binární soubor	84
Textový soubor	84
Otevření souboru	85
Otevření souboru v nesprávném režimu	86
Formátovaný výstup v binárním režimu	87
Neformátovaný vstup a výstup v textovém režimu	87
5.8 Konzola	88
Je konzola textový soubor?	88
6. Trocha počítání	90
6.1 Problémy s celými čísly	90
Celočíselné dělení	90
Celočíselné přetečení	91
Celočíselné konverze	95
Platí asociativní zákon?	96
6.2 Problémy s reálnými čísly	97
Porovnávání reálných čísel	98
Sčítání reálných čísel	98
Knihovni funkce	100
Není arkustangens jako arkustangens	101
7. Zapouzdření, metody	104
7.1 Zapouzdření má být vodotěsné	104
Dočasné proměnné	106
7.2 Volání metod	107
Třída je oblast viditelnosti	107
Dominance	108
7.3 Metody, které volá překladač automaticky	109
Další automaticky volané metody	111
7.4 Konstantní objekty	112
Metody konstantních objektů	112
Dvě verze jedné metody	113
Když konstanty nejsou konstantní	114
Měnitelné složky konstant	114
8. Dědičnost	116
8.1 Přetypování mezi předkem a potomkem	116
8.2 Adresa objektu	117
8.3 Přetypování v novém stylu	120
Operátor <code>dynamic_cast</code>	121
9. Konstruktory a destruktory	128
9.1 Inicializace předků a atributů	128
Konstruktor by měl inicializovat všechny složky instance	128
Inicializační část konstruktoru	129
Konstruktory virtuálních předků	130
Inicializace nepřímého virtuálního předka	131

Explicitní volání jiného konstruktoru	133
9.2 Konstruktory volá konstruktory předků a destruktory volá destruktory předků	135
9.3 Kopírovací konstruktory	136
Parametry kopírovacího konstruktoru.....	136
Když nestačí implicitní kopírovací konstruktor.....	137
Přiřazovací operátor nenahradí kopírovací konstruktor	139
9.4 Implicitní volání konstruktoru.....	139
Explicitní konstruktory	141
Starší překladače.....	142
10. Virtuální metody.....	144
10.1 Kdy použít virtuální metody.....	144
Rozhraní musí být stejné.....	146
10.2 Implicitní hodnoty parametrů.....	146
10.3 Virtuální metody a vícenásobná dědičnost	147
10.4 Virtuální destruktory	149
10.5 Konstruktory, destruktory a virtuální metody.....	151
10.6 Zbytečné virtuální metody.....	153
10.7 Deklarace patří do hlavičkových souborů	153
10.8 Čistě virtuální destruktory	155
10.9 Jak se vyhnout pozdní vazbě.....	155
10.10 Shrnutí.....	156
11. Správa paměti	157
11.1 Operátory new a delete	157
Operátor new vrací ukazatel správného typu	157
Operátor new nemusí uspět.....	157
Napišeme-li new, musíme napsat i delete.....	158
delete musí mít své new.....	158
Zapomeňme na malloc () a free ()	159
Nekonečný cyklus v new.....	159
Operátor delete []	160
11.2 Vývoj operátoru new.....	161
12. Přetěžování operátorů.....	163
12.1 Pravidla slušného chování	163
Aritmetické operátory +, -, *, / a %.....	163
Přiřazovací operátor =	163
Globální operátor new.....	164
Operátory &&, a ,	165
Jen konvence, nic víc.....	165
12.2 Zřetěžené operátory.....	165
12.3 Kdy nevracet výsledek odkazem.....	167
12.4 Přiřazovací operátor =	170
Když nestačí implicitní přiřazovací operátor	170
Přiřazovací operátor se nedědí.....	171
Přiřazovací operátor nemůže zastoupit kopírovací konstruktor	171

Shodné objekty a přiřazování	172
Paměť jen tak okopírovaná.....	174
Když nedodržíme pravidla slušného chování	176
12.5 Složené přiřazovací operátory	176
12.6 Operátory new a delete	178
Operátory new a delete lze předefinovat	178
new má mít své delete	180
Operátor new a pole	181
Co vrací přetížený operátor new [] ?	182
12.7 Operátor ->.....	183
12.8 Operátory , && a ,	183
12.9 Operátory ++ a --	184
13. Výjimky	188
13.1 Výjimku je třeba zachytit	188
Výjimky a dědičnost	188
Žádné jiné konverze.....	189
Černá díra na výjimky.....	190
Výjimky chytáme odkazem.....	191
Jednosměrná ulice.....	192
13.2 Zdánlivé bezpečí.....	195
Únik prostředků při výjimce	195
Výjimky v destrukturu	199
Výjimky v konstrukturu.....	201
13.3 Závazky je třeba dodržovat.....	205
Specifikace výjimek a šablony.....	206
Funkce volaná z unexpected ()	207
13.4 Borland C++: podrobnější informace o výjimce	207
13.5 Výjimky by měly zůstat výjimečné.....	208
14. Závěrečné kolo.....	211
14.1 errno	211
14.2 Proměnné operačního systému	212
14.3 Šablony a spřátelené funkce.....	213
14.4 Optimalizace	215
Optimalizace vracené hodnoty.....	215
Přetěžováním k efektivitě	216
14.5 Inicializace globálních proměnných	217
14.6 Operátory const_cast a reinterpret_cast.....	219
15. Složité deklarace.....	221
15.1 Začneme od začátku.....	221
15.2 Jak je tedy čist	222
15.3 Několik příkladů.....	222
15.4 Označení typu	225
15.5 Skok na pevně danou adresu (DOS)	225

16. Kompatibilita a přenositelnost.....	227
16.1 Co v C++ chybí	227
Deklarace funkce	227
Deklarace proměnných	228
Datové typy.....	230
16.2 Co je v C++ jinak.....	234
Deklarace funkcí.....	234
Konstanty.....	234
Znaky.....	235
Jména struktur.....	235
Makra.....	235
Knihovní funkce	236
Uložení instance v paměti.....	236
16.3 Různé verze jazyka.....	237
Virtuální metody	237
Destruktory	238
Přístupová práva	238
Operátory	239
Výjimky, dynamická identifikace typů	239
Šablony	240
Konverze.....	240
Strukturované výjimky.....	241
16.4 ANSI?	242

Literatura	243
-------------------------	------------

Rejstřík.....	244
----------------------	------------