

# Obsah

Předmluva	3
Obsah	6
<b>1 Seznam</b>	<b>11</b>
1.1 Seznam – první verze . . . . .	11
1.1.1 Návrh třídy seznam . . . . .	11
1.1.2 Implementace . . . . .	12
1.1.3 Zkoušíme třídu seznam . . . . .	17
1.2 Kopírování instancí . . . . .	18
1.2.1 Zákaz kopírování . . . . .	19
1.2.2 Implementace kopírovacího konstrukturu . . . . .	20
1.3 Přiřazování instancí . . . . .	21
1.3.1 Zákaz přiřazování . . . . .	22
1.3.2 Implementace přiřazovacího operátoru . . . . .	22
1.4 Jednotná inicializace a konstantní seznamy . . . . .	23
1.4.1 Jednotná inicializace . . . . .	24
1.4.2 Konstantní seznam . . . . .	25
1.5 Spojování seznamů . . . . .	27
1.5.1 Operátor += . . . . .	27
1.5.2 Operátor + . . . . .	29
1.6 Stěhování neboli tunelování (C++11) . . . . .	29
1.6.1 Tunelovací konstruktor třídy seznam . . . . .	30
1.6.2 Tunelovací přiřazovací operátor třídy seznam . . . . .	31
1.7 Šablonová implementace . . . . .	31
1.7.1 Úprava stávající implementace . . . . .	32
1.7.2 Funkce vytvoř_seznam() . . . . .	35
1.8 Typ hodnot ukládaných do seznamu . . . . .	38
1.8.1 Požadavky kladené na typ hodnot . . . . .	38
1.8.2 Pojmy . . . . .	39
<b>2 Objektově orientované programování</b>	<b>41</b>
2.1 Objektový typ . . . . .	41
2.1.1 Deklarace objektového typu . . . . .	41
2.1.2 Datové složky . . . . .	43
2.1.3 Metody . . . . .	45

2.1.4	Obvyklá struktura programu s objekty v C++ . . . . .	51
2.1.5	Specifikace přístupu . . . . .	52
2.1.6	Přátelé . . . . .	54
2.1.7	Statické složky tříd . . . . .	55
2.1.8	Deklarace typu v těle třídy . . . . .	58
2.2	Inicializace, konstruktory a destruktory . . . . .	60
2.2.1	Agregát a jeho inicializace . . . . .	60
2.2.2	Konstruktory a destruktory . . . . .	61
2.2.3	Konstruktory . . . . .	62
2.2.4	Použití konstruktorů . . . . .	66
2.2.5	Kopírování a stěhování obsahu instancí . . . . .	72
2.2.6	Destruktory . . . . .	75
2.3	Třídy, struktury a unie . . . . .	77
2.3.1	Struktury a třídy . . . . .	77
2.3.2	Unie . . . . .	78
<b>3</b>	<b>Dědění a polymorfismus</b>	<b>79</b>
3.1	Syntaktická pravidla . . . . .	79
3.2	Nevirtuální dědění . . . . .	80
3.2.1	Dědění a přístupová práva . . . . .	81
3.2.2	Konflikt jmen . . . . .	83
3.2.3	Třída je oblast viditelnosti deklaráce . . . . .	84
3.3	Virtuální dědění . . . . .	85
3.4	Potomek může zastoupit předka . . . . .	87
3.4.1	Přetypování instance potomka na předka . . . . .	87
3.4.2	Přetypování ukazatele na potomka na ukazatel na předka . . . . .	88
3.4.3	Přetypování předka na potomka . . . . .	89
3.4.4	Přiřazení, parametry, reference . . . . .	89
3.5	Konstruktory, destruktory a dědění . . . . .	90
3.5.1	Volání konstruktoru virtuálního předka . . . . .	93
3.5.2	Dědění konstruktorů (C++11) . . . . .	94
3.6	Polymorfismus (virtuální metody) . . . . .	96
3.6.1	Časná a pozdní vazba . . . . .	98
3.6.2	Abstraktní třídy a čistě virtuální metody . . . . .	101
3.6.3	Finální metody a finální třídy (C++11) . . . . .	102
3.6.4	Implementace polymorfismu . . . . .	103
3.6.5	Konstruktory, destruktory a virtuální metody . . . . .	104
3.7	Třídní ukazatele . . . . .	104
<b>4</b>	<b>Přetěžování operátorů</b>	<b>107</b>
4.1	Základní pravidla . . . . .	107
4.1.1	Operátorové funkce . . . . .	108
4.2	Běžné operátory . . . . .	108
4.2.1	Unární operátory . . . . .	108
4.2.2	Binární operátory . . . . .	113
4.3	Operátory přetěžovatelné jen jako metody . . . . .	116
4.3.1	Operátor indexování [ ] . . . . .	116

4.3.2	Operátor volání funkce ()	119
4.3.3	Přiřazovací operátor =	120
4.3.4	Operátor nepřímého přístupu ->	123
4.3.5	Operátor přetypování	125
4.4	Operátory new a delete	128
4.4.1	Globální operátory new a delete	128
4.4.2	Operátory new a delete jako metody objektových typů	133
4.4.3	Delete s dodatečnými parametry	135
4.5	Uživatелеm definované literály (C++11)	135
4.5.1	Pravidla definice literálu	136
4.5.2	Funkce pro uživatelské literály	137
4.5.3	Volání odpovídající funkce	138
<b>5</b>	<b>Šablony</b>	<b>143</b>
5.1	Deklarace šablony	143
5.1.1	Parametry šablony	144
5.2	Šablony objektových typů	146
5.2.1	Šablony metod	148
5.2.2	Šablony statických datových složek	149
5.2.3	Instance šablony	150
5.2.4	Vnořené šablony	151
5.2.5	Typ jako složka typu	152
5.3	Šablony funkcí	153
5.3.1	Instance šablony volné funkce	154
5.3.2	Přetěžování šablon funkcí	156
5.3.3	Spřátelené funkce a šablony	160
5.4	Šablony aliasu a šablony proměnných	164
5.4.1	Šablony aliasu (C++11)	164
5.4.2	Šablony proměnných (C++14)	165
5.5	Explicitní generování instancí	166
5.6	Vazba jmen	167
5.6.1	Specifikace template	167
5.7	Částečná specializace šablon objektových typů	168
5.7.1	Deklarace částečné specializace	168
5.8	Šablonové metaprogramování	173
5.8.1	Základní nástroje	173
5.9	Šablony s proměnným počtem parametrů (C++11)	179
5.9.1	Balík parametrů	179
5.9.2	Použití balíku parametrů	179
5.9.3	Redukce balíku (C++17)	185
5.9.4	Použití variadických šablon	187
5.10	Automatické odvození typu parametru (C++17)	193
5.10.1	Základní použití	194
5.10.2	Použití v balíku parametrů šablon	195
5.11	Generické lambda-výrazy (C++14)	196
5.11.1	Syntax generického lambda-výrazu	196
5.11.2	Jak je generický lambda-výraz implementován	196

5.12	Organizace programu . . . . .	198
5.12.1	Současné překladače . . . . .	199
<b>6</b>	<b>Výjimky</b>	<b>201</b>
6.1	O co jde . . . . .	201
6.2	Výjimky v C++ . . . . .	202
6.2.1	Základní pojmy . . . . .	202
6.2.2	Syntax nástrojů pro práci s výjimkami . . . . .	203
6.2.3	Když vznikne výjimka . . . . .	204
6.2.4	Obsluha . . . . .	207
6.2.5	Výjimky a funkce . . . . .	211
6.2.6	Výjimky a alokace paměti . . . . .	218
6.2.7	Výjimky v obsluze . . . . .	219
6.2.8	Výjimkový objekt . . . . .	220
6.2.9	Neošetřené a neočekávané výjimky . . . . .	220
6.2.10	Standardní výjimky . . . . .	221
6.2.11	Vnořené výjimky (C++11) . . . . .	223
6.2.12	Třída <code>system_error</code> . . . . .	225
6.3	Strukturované výjimky v jazyce C . . . . .	225
6.3.1	První přiblížení . . . . .	225
6.3.2	Přenos informací o výjimce . . . . .	226
6.3.3	Syntax strukturovaných výjimek . . . . .	227
6.3.4	Vznik strukturovaných výjimek . . . . .	229
6.3.5	Filtr . . . . .	230
6.3.6	Koncovka bloku . . . . .	232
6.3.7	Neošetřené výjimky . . . . .	236
6.3.8	Strukturované výjimky a C++ . . . . .	237
<b>7</b>	<b>Datové typy za běhu programu</b>	<b>238</b>
7.1	Dynamická identifikace typů . . . . .	238
7.1.1	Operátor <code>typeid</code> . . . . .	238
7.2	Přetypování . . . . .	241
7.2.1	Přetypovací operátory . . . . .	241
<b>8</b>	<b>Jmenné prostory</b>	<b>250</b>
8.1	O co vlastně jde . . . . .	250
8.2	Deklarace jmenného prostoru . . . . .	250
8.2.1	Deklarace po částech . . . . .	254
8.2.2	Anonymní jmenný prostor . . . . .	255
8.2.3	Vložený jmenný prostor (C++11) . . . . .	255
8.3	Zpřístupnění jmenného prostoru . . . . .	257
8.3.1	Přezdívka (alias) jmenného prostoru . . . . .	257
8.3.2	Deklarace a direktiva <code>using</code> . . . . .	258
8.4	Jmenné prostory a třídy . . . . .	260
8.5	Vyhledávání funkcí a operátorů . . . . .	260

<b>9</b>	<b>Pojmy a koncepty</b>	<b>263</b>
9.1	Knihovní pojmy . . . . .	263
9.2	Koncepty (C++17) . . . . .	268
9.2.1	Oč jde . . . . .	268
9.2.2	Zástupné symboly . . . . .	270
9.2.3	Zkrácená deklarace šablony . . . . .	271
9.2.4	Koncepty . . . . .	272
9.2.5	Omezení . . . . .	273
9.2.6	Požadavky (requires) . . . . .	276
9.2.7	Další pravidla . . . . .	280
<b>10</b>	<b>Dodatek</b>	<b>281</b>
10.1	Příkazy . . . . .	281
10.1.1	Příkaz switch . . . . .	281
10.1.2	Příkaz if . . . . .	283
10.2	Deklarace, proměnné . . . . .	285
10.2.1	Vložené proměnné . . . . .	285
10.2.2	Specifikátor auto a seznam inicializátorů . . . . .	286
10.2.3	Vazba struktur . . . . .	286
10.2.4	Znakový literál v UTF-8 . . . . .	289
10.2.5	Deklarace výčtového typu . . . . .	289
10.3	Moduly . . . . .	289
10.3.1	Význam . . . . .	289
10.3.2	Deklarace modulu . . . . .	289
10.3.3	Import modulu . . . . .	291
10.3.4	Závěrečné poznámky . . . . .	292
	<b>Literatura</b>	<b>293</b>
	<b>Rejstřík</b>	<b>296</b>