

# Obsah



<b>Předmluva Scotta Meyerse</b>	<b>11</b>
<b>Předmluva Johna Vlissidese</b>	<b>14</b>
<b>Předmluva k českému vydání</b>	<b>15</b>
<b>Úvod</b>	<b>17</b>
Publikum	18
Knihovna Loki	19
Uspořádání	20
<b>Poděkování</b>	<b>21</b>
 <b>část 1</b>	 <b>23</b>
<b>kapitola 1 Návrh tříd založený na zásadách</b>	<b>25</b>
1.1 Rozmanitost návrhu softwaru	25
1.2 Selhání rozhraní „vše-v-jednom“	26
1.3 Je vícenásobná dědičnost řešením?	27
1.4 Přínos šablon	28
1.5 Zásady a třídy zásad	29
1.6 Obohacené zásady	34
1.7 Destruktory tříd zásad	34
1.8 Volitelná funkcionalita skrze vytváření neúplných instancí	35
1.9 Kombinování tříd zásad	36
1.10 Jak přizpůsobovat struktury pomocí tříd zásad	38
1.11 Kompatibilní a nekompatibilní zásady	39
1.12 Dekompozice třídy na zásady	41
1.13 Shrnutí	42
 <b>kapitola 2 Techniky</b>	 <b>45</b>
2.1 Aserce při kompilaci	45
2.2 Částečná specializace šablon	48
2.3 Lokální třídy	50
2.4 Mapování celočíselných konstant na typy	51
2.5 Mapování typu na typ	53
2.6 Výběr typů	54
2.7 Detekce konvertibility a dědičnosti při kompilaci	56
2.8 Obal okolo type_info	59
2.9 NullTyp a PrazdnyTyp	61
2.10 Rysy typů	61
2.11 Shrnutí	69



<b>kapitola 3</b>	<b>Seznamy typů</b>	<b>71</b>
3.1	Potřeba seznamů typů	71
3.2	Definování seznamu typů	73
3.3	Linearizace tvorby seznamů typů	74
3.4	Výpočet délky	75
3.5	Vsuvka	76
3.6	Indexovaný přístup	76
3.7	Prohledávání seznamu typů	78
3.8	Připojování k seznamu typů	79
3.9	Smazání typu ze seznamu typů	80
3.10	Mazání duplikátů	81
3.11	Nahrazení prvku v seznamu typů	82
3.12	Částečné setřídění seznamu typů	83
3.13	Generování tříd se seznamy typů	86
3.14	Shrnutí	96
3.15	SeznamTypu ve zkratce	97
<b>kapitola 4</b>	<b>Alokace paměti pro malé objekty</b>	<b>99</b>
4.1	Implicitní alokátor volné paměti	100
4.2	Jak pracuje alokátor paměti	100
4.3	Alokátor paměti pro malé objekty	102
4.4	Segmenty	103
4.5	Alokátor bloku paměti pevné velikosti	106
4.6	Třída AlokatorMalychObjektu	109
4.7	Malý trik	111
4.8	Jednoduché, komplikované a nakonec opět jednoduché	114
4.9	Základy používání	115
4.10	Shrnutí	116
4.11	Alokátor malých objektů ve zkratce	116
<b>Část 2</b>	<b>Komponenty</b>	<b>119</b>
<b>kapitola 5</b>	<b>Zobecněné funktory</b>	<b>121</b>
5.1	Návrhový vzor Příkaz	122
5.2	Příkaz ve skutečném světě	124
5.3	Volatelné entity C++	125
5.4	Kostra šablony třídy Funktor	126
5.5	Implementace předávacího Funktor::operator()	130
5.6	Manipulace s funktory	132
5.7	Postavíte jeden a druhý dostanete zdarma	134
5.8	Konverze typů argumentů a návratových typů	135
5.9	Manipulace s ukazateli na členské funkce	137
5.10	Vázání	141

5.11	Řetězení požadavků	143
5.12	Problémy skutečného světa poprvé: Cena předávacích funkcí	144
5.13	Problémy skutečného světa podruhé: Alokace paměti v haldě	145
5.14	Implementace funkcí „zpět“ a „opakovat“ pomocí třídy Funktor	147
5.15	Shrnutí	147
5.16	Funktor ve zkratce	148
<b>kapitola 6</b>	<b>Implementace Singletonu</b>	<b>151</b>
6.1	Statická data + statické funkce != Singleton	152
6.2	Základní idiomy jazyka C++ podporující Singletony	153
6.3	Zajištění jedinečnosti Singletonu	154
6.4	Odstranění Singletonu	155
6.5	Problém mrtvého odkazu	157
6.6	Řešení problému mrtvého odkazu (I): Singleton Fénix	159
6.7	Řešení problému mrtvého odkazu (II): Singletony s dlouhověkostí	161
6.8	Implementace Singletonů s dlouhověkostí	164
6.9	Co si počít při větvení do vláken	167
6.10	Jak to všechno sestavit dohromady	170
6.11	Práce s DrzitelSingletonu	175
6.12	Shrnutí	177
6.13	Šablona třídy DrzitelSingletonu ve zkratce	177
<b>kapitola 7</b>	<b>Inteligentní ukazatele</b>	<b>179</b>
7.1	Inteligentní ukazatele 101	179
7.2	Kompromis	180
7.3	Ukládání inteligentních ukazatelů	182
7.4	Členské funkce inteligentního ukazatele	183
7.5	Strategie práce s vlastnictvím	185
7.6	Operátor získání adresy	192
7.7	Implicitní konverze na typy obyčejných ukazatelů	193
7.8	Rovnost a nerovnost	195
7.9	Operátory uspořádání	200
7.10	Kontrola a oznamování chyb	202
7.11	Inteligentní ukazatele na konstanty a konstantní inteligentní ukazatele	204
7.12	Pole	205
7.13	Inteligentní ukazatele a větvení do vláken	205
7.14	Sestavení všeho dohromady	209
7.15	Shrnutí	216
7.16	IntUkazatel ve zkratce	216
<b>kapitola 8</b>	<b>Továrny na objekty</b>	<b>219</b>
8.1	Proč potřebujeme továrny na objekty	220
8.2	Objektové továrny v C++: Třídy a objekty	222



8.3	Implementování továrny na objekty	223
8.4	Identifikátory typů	227
8.5	Zobecnění	229
8.6	Drobnosti	232
8.7	Klonovací továrny	233
8.8	Používání továren na objekty s jinými generickými komponentami	236
8.9	Shrnutí	237
8.10	Šablona třídy Tovarna ve zkratce	237
8.11	Šablona třídy KlonovacíTovarna ve zkratce	238
<b>kapitola 9</b>	<b>Abstraktní továrna</b>	<b>241</b>
9.1	Architektonická role Abstraktní továrny	241
9.2	Rozhraní generické Abstraktní továrny	245
9.3	Implementace třídy AbstraktniTovarna	248
9.4	Implementace Abstraktní továrny založená na prototypch	251
9.5	Shrnutí	255
9.6	AbstraktniTovarna a KonkretniTovarna ve zkratce	256
<b>kapitola 10</b>	<b>Návštěvník</b>	<b>259</b>
10.1	Základy Návštěvníka	259
10.2	Přetížení a všezachytávající funkce	266
10.3	Zdokonalení implementace: Acyklický návštěvník	267
10.4	Generická implementace Návštěvníka	272
10.5	Zpět k „Cyklickému“ návštěvníkovi	279
10.6	Zapojení variací	282
10.7	Shrnutí	284
10.8	Generické komponenty Návštěvníka ve zkratce	285
<b>kapitola 11</b>	<b>Multimetody</b>	<b>287</b>
11.1	Co jsou multimetody?	288
11.2	Kdy jsou multimetody potřeba?	288
11.3	Dvojitě přepnutí podle typu: Hrubá síla	289
11.4	Automatizovaný přístup hrubé síly	292
11.5	Symetrie s dispečerem hrubé síly	296
11.6	Logaritmický dvojitý dispečer	300
11.7	FnDispecer a symetrie	305
11.8	Dvojitý dispečer pro funktory	306
11.9	Konverze argumentů: static_cast nebo dynamic_cast?	308
11.10	Multimetody v konstantním čase: Čistá rychlost	313
11.11	ZakladniDispecer, ZakladniRychlyDispecer a zásady	316
11.12	Výhled	318
11.13	Shrnutí	319
11.14	Dvojitý dispečer ve zkratce	321

<b>dodatek</b>	<b>Minimalistická knihovna větvení do vláken</b>	<b>325</b>
A.1	Kritika větvení do vláken	325
A.2	Přístup knihovny Loki	327
A.3	Atomické operace na celočíselných typech	327
A.4	Mutexy	329
A.5	Sémantika zamykání v objektově orientovaném programování	330
A.6	Volitelný modifikátor volatile	332
A.7	Semaforey, události a další užitečné věci	333
A.8	Shrnutí	333
	<b>Rejstřík</b>	<b>335</b>