

СОДЕРЖАНИЕ

Предисловие Скотта Мейерса	11
Предисловие Джона Влиссидеса	15
Предисловие	17
Аудитория	18
Библиотека Loki	19
Структура книги	20
Благодарности	21
Часть I. Методы	23
Глава 1. Разработка классов на основе стратегий	25
1.1. Разнообразие методов разработки программного обеспечения	25
1.2. Недостатки универсального интерфейса	26
1.3. Опасно ли множественное наследование?	28
1.4. Преимущества шаблонов	29
1.5. Стратегии и классы стратегий	30
1.5.1. Реализация классов стратегий с помощью шаблонных параметров	32
1.5.2. Реализация классов стратегий с помощью шаблонных функций-членов	34
1.6. Расширенные стратегии	34
1.7. Деструкторы классов стратегий	35
1.8. Факультативные возможности, предоставляемые неполной конкретизацией	36
1.9. Комбинирование классов стратегий	37
1.10. Настройка структур с помощью классов стратегий	39
1.11. Совместимые и несовместимые стратегии	39
1.12. Разложение классов на стратегии	41
1.13. Резюме	43
Глава 2. Приемы программирования	45
2.1. Статическая проверка диагностических утверждений	46
2.2. Частичная специализация шаблонов	48
2.3. Локальные классы	50
2.4. Отображение целочисленных констант в типы	51
2.5. Отображение одного типа в другой	53
2.6. Выбор типа	54
2.7. Распознавание конвертируемости и наследования на этапе компиляции	56
2.8. Оболочка вокруг класса type_info	59
2.9. Классы NullType и EmptyType	61
2.10. Характеристики типов	61
2.10.1. Реализация характеристик указателей	62
2.10.2. Распознавание основных типов	63
2.10.3. Оптимальные типы параметров	64
2.10.4. Удаление квалификаторов	65
2.10.5. Применение класса TypeTraits	66
2.10.6. Заключение	67
2.11. Резюме	68

Глава 3. Списки типов	71
3.1. Зачем нужны списки типов	71
3.2. Определение списков типов	73
3.3. Линеаризация создания списков типов	74
3.4. Вычисление длины списка	75
3.5. Интермеццо	76
3.6. Индексированный доступ	77
3.7. Поиск элемента	78
3.8. Добавление элемента	79
3.9. Удаление элемента	80
3.10. Удаление дубликатов	81
3.11. Замена элемента	82
3.12. Частично упорядоченные списки типов	83
3.13. Генерация класса на основе списка типов	86
3.13.1. Генерация распределенных иерархий	86
3.13.2. Генерация кортежей	91
3.13.3. Генерация линейных иерархий	92
3.14. Резюме	95
3.15. Краткое описание класса Typelist	96
Глава 4. Размещение в памяти небольших объектов	99
4.1. Стандартный механизм распределения динамической памяти	100
4.2. Как работает стандартный механизм распределения динамической памяти	100
4.3. Распределитель памяти для небольших объектов	102
4.4. Класс Chunk	103
4.5. Класс FixedAllocator	106
4.6. Класс SmallObjAllocator	110
4.7. Трюк	112
4.8. Просто, сложно и снова просто	114
4.9. Применение	115
4.10. Резюме	116
4.11. Краткое описание механизма распределения памяти для небольших объектов	117
Часть II. Компоненты	119
Глава 5. Обобщенные функторы	121
5.1. Шаблон Command	122
5.2. Шаблон Command в реальном мире	124
5.3. Вызываемые сущности в языке C++	125
5.4. Скелет шаблонного класса Functor	126
5.5. Реализация оператора пересылки Functor::operator()	131
5.6. Работа с функторами	132
5.7. Один пишем, два в уме	134
5.8. Преобразование типов аргументов и возвращаемого значения	136
5.9. Указатели на функции-члены	137
5.10. Связывание	141
5.11. Сцепление	143
5.12. Первая практическая проблема: стоимость функций пересылки	144

5.13. Вторая практическая проблема: распределение динамической памяти	146
5.14. Реализация операций Undo и Redo с помощью класса Functor	147
5.15. Резюме	148
5.16. Краткое описание класса Functor	148
Глава 6. Реализация шаблона Singleton	151
6.1. Статические данные + статические функции != синглтон	152
6.2. Основные идиомы языка C++ для поддержки синглтонов	153
6.3. Обеспечение уникальности синглтонов	154
6.4. Разрушение объектов класса Singleton	155
6.5. Проблема висячей ссылки	157
6.6. Проблема адресации висячей ссылки (I): феникс	159
6.6.1. Проблемы, связанные с функцией atexit	161
6.7. Проблема висячей ссылки (II): синглтон с заданной продолжительностью жизни	162
6.8. Реализация синглтонов, имеющих заданную продолжительность жизни	164
6.9. Продолжительность жизни объектов в многопоточной среде	167
6.9.1. Шаблон блокировки с двойной проверкой	168
6.10. Сборка	170
6.10.1. Разложение класса SingletonHolder на стратегии	171
6.10.2. Требования, предъявляемые к стратегиям класса SingletonHolder	171
6.10.3. Сборка класса SingletonHolder	172
6.10.4. Реализации стратегий	174
6.11. Работа с классом SingletonHolder	175
6.12. Резюме	176
6.13. Краткое описание шаблонного класса SingletonHolder	177
Глава 7. Интеллектуальные указатели	179
7.1. Сто первое описание интеллектуальных указателей	179
7.2. Особенности интеллектуальных указателей	180
7.3. Хранение интеллектуальных указателей	182
7.4. Функции-члены интеллектуальных указателей	183
7.5. Стратегии владения	185
7.5.1. Глубокое копирование	185
7.5.2. Копирование при записи	186
7.5.3. Подсчет ссылок	187
7.5.4. Связывание ссылок	189
7.5.5. Разрушающее копирование	190
7.6. Оператор взятия адреса	192
7.7. Неявное приведение к типам обычных указателей	193
7.8. Равенство и неравенство	195
7.9. Отношения порядка	200
7.10. Обнаружение и регистрация ошибок	202
7.10.1. Проверка во время инициализации	202
7.10.2. Проверка перед разыменованием	203
7.10.3. Сообщения об ошибках	203
7.11. Интеллектуальные указатели на константные объекты и константные интеллектуальные указатели	204
7.12. Массивы	205

7.13. Интеллектуальные указатели и многопоточность	205
7.13.1. Многопоточность на уровне объектов	205
7.13.2. Многопоточность на уровне регистрации данных	207
7.14. Сборка	209
7.14.1. Многопоточность на уровне объектов	210
7.14.2. Стратегия Ownership	212
7.14.3. Стратегия Conversion	214
7.14.4. Стратегия Checking	214
7.15. Резюме	215
7.16. Краткий обзор класса SmartPtr	216
Глава 8. Фабрики объектов	217
8.1. Для чего нужны фабрики объектов	218
8.2. Фабрики объектов в языке C++: классы и объекты	220
8.3. Реализация фабрики объектов	221
8.4. Идентификаторы типов	225
8.5. Обобщение	227
8.6. Мелкие детали	230
8.7. Фабрика клонирования	231
8.8. Использование фабрики объектов в сочетании с другими обобщенными компонентами	234
8.9. Резюме	235
8.10. Краткий обзор шаблонного класса Factory	235
8.11. Краткий обзор шаблонного класса CloneFactory	236
Глава 9. Шаблон Abstract Factory	239
9.1. Архитектурная роль шаблона Abstract Factory	239
9.2. Обобщенный интерфейс шаблона Abstract Factory	242
9.3. Реализация класса AbstractFactory	245
9.4. Реализация шаблона Abstract Factory на основе прототипов	249
9.5. Резюме	252
9.6. Краткий обзор классов AbstractFactory и ConcreteFactory	253
Глава 10. Шаблон Visitor	255
10.1. Основы шаблона Visitor	255
10.2. Перегрузка и функция-ловушка	261
10.3. Уточнение реализации: шаблон Acyclic Visitor	262
10.4. Обобщенная реализация шаблона Visitor	268
10.5. Назад — к “простому” шаблону Visitor	274
10.6. Отладка вариантов	277
10.6.1. Функция-ловушка	277
10.6.2. Нестрогое инспектирование	279
10.7. Резюме	279
10.8. Краткий обзор обобщенных компонентов шаблона Visitor	280
Глава 11. Мультиметоды	281
11.1. Что такое мультиметоды?	282
11.2. Когда нужны мультиметоды	282
11.3. Двойное переключение по типу: грубый подход	284
11.4. Автоматизированный грубый подход	286

11.5. Симметричность грубого подхода	290
11.6. Логарифмический двойной диспетчер	294
11.6.1. Логарифмический диспетчер и наследование	296
11.6.2. Логарифмический диспетчер и приведение типов	297
11.7. Класс FnDispatcher и симметрия	299
11.8. Двойная диспетчеризация функторов	300
11.9. Преобразование аргументов: static_cast или dynamic_cast?	302
11.10. Мультиметоды с постоянным временем выполнения	307
11.11. Классы BasicDispatcher и BasicFastDispatcher как стратегии	310
11.12. Перспективы	311
11.13. Резюме	312
11.14. Краткий обзор двойных диспетчеров	314
Приложение. Многопоточная библиотека в стиле минимализма	319
П.1. Критика многопоточности	320
П.2. Подход, реализованный в библиотеке Loki	321
П.3. Атомарные операции с целочисленными типами	321
П.4. Мьютексы	323
П.5. Семантика блокировки в объектно-ориентированном программировании	325
П.6. Модификатор volatile	327
П.7. Семафоры, события и другие полезные вещи	327
П.8. Резюме	327
Библиография	329
Предметный указатель	331