

Содержание

Предисловие	17
Благодарности	18
Глава 1. Об этой книге	21
1.1. Что необходимо знать, приступая к чтению этой книги	22
1.2. Структура книги в целом	22
1.3. Как читать эту книгу	23
1.4. Некоторые замечания о стиле программирования	24
1.5. Стандарт и практика	26
1.6. Примеры кода и дополнительная информация	26
1.7. Обратная связь с авторами	26
Часть I. Основы	29
Зачем нужны шаблоны	29
Глава 2. Шаблоны функций	31
2.1. Первое знакомство с шаблонами функций	31
2.1.1. Определение шаблона	31
2.1.2. Использование шаблонов	32
2.2. Вывод аргументов	34
2.3. Параметры шаблонов	35
2.4. Перегрузка шаблонов функций	37
2.5. Резюме	42
Глава 3. Шаблоны классов	43
3.1. Реализация шаблона класса Stack	43
3.1.1. Объявление шаблонов классов	44
3.1.2. Реализация функций-членов	45
3.2. Использование шаблона класса Stack	47
3.3. Специализации шаблонов класса	49
3.4. Частичная специализация	51
3.5. Аргументы шаблона, задаваемые по умолчанию	52
3.6. Резюме	54

Глава 4. Параметры шаблонов, не являющиеся типами	57
4.1. Параметры шаблонов классов, не являющиеся типами	57
4.2. Параметры шаблонов функций, не являющиеся типами	61
4.3. Ограничения на параметры шаблонов, не являющиеся типами	62
4.4. Резюме	63
Глава 5. Основы работы с шаблонами	65
5.1. Ключевое слово <code>typename</code>	65
5.2. Использование <code>this-></code>	67
5.3. Шаблоны-члены классов	68
5.4. Шаблонные параметры шаблонов	72
5.5. Инициализация нулем	78
5.6. Использование строковых литералов в качестве аргументов шаблонов функций	79
5.7. Резюме	82
Глава 6. Применение шаблонов на практике	83
6.1. Модель включения	83
6.1.1. Ошибки при компоновке	83
6.1.2. Шаблоны в заголовочных файлах	85
6.2. Явное инстанцирование	87
6.2.1. Пример явного инстанцирования шаблона	87
6.2.2. Сочетание модели включения и явного инстанцирования	88
6.3. Модель разделения	89
6.3.1. Ключевое слово <code>export</code>	90
6.3.2. Ограничения модели разделения	92
6.3.3. Составление программы для модели разделения	93
6.4. Шаблоны и <code>inline</code>	94
6.5. Предварительно откомпилированные заголовочные файлы	95
6.6. Отладка шаблонов	98
6.6.1. Дешифровка ошибок-романов	98
6.6.2. Мелкое инстанцирование	100
6.6.3. Длинные имена	103
6.6.4. Трассировщики	103
6.6.5. Интеллектуальные трассировщики	107
6.6.6. Прототипы	108
6.7. Заключение	108
6.8. Резюме	109
Глава 7. Основные термины в области шаблонов	111
7.1. “Шаблон класса” или “шаблонный класс”	111
7.2. Инстанцирование и специализация	112
7.3. Объявления и определения	113

Содержание	9
7.4. Правило одного определения	114
7.5. Аргументы и параметры шаблонов	114
Часть II. Углубленное изучение шаблонов	117
Глава 8. Вглубь шаблонов	119
8.1. Параметризованные объявления	119
8.1.1. Виртуальные функции-члены	122
8.1.2. Связывание шаблонов	122
8.1.3. Первичные шаблоны	124
8.2. Параметры шаблонов	124
8.2.1. Параметры типа	125
8.2.2. Параметры, не являющиеся типами	125
8.2.3. Шаблонные параметры шаблона	126
8.2.4. Аргументы шаблона, задаваемые по умолчанию	127
8.3. Аргументы шаблонов	128
8.3.1. Аргументы шаблонов функций	129
8.3.2. Аргументы типов	132
8.3.3. Аргументы, не являющиеся типами	133
8.3.4. Шаблонные аргументы шаблонов	135
8.3.5. Эквивалентность	137
8.4. Друзья	138
8.4.1. Дружественные функции	138
8.4.2. Дружественные шаблоны	141
8.5. Заключение	142
Глава 9. Имена в шаблонах	143
9.1. Систематизация имен	143
9.2. Поиск имен	145
9.2.1. Поиск, зависящий от аргументов	147
9.2.2. Внесение дружественных имен	149
9.2.3. Внесение имен классов	150
9.3. Синтаксический анализ шаблонов	151
9.3.1. Зависимость от контекста в нешаблонных конструкциях	152
9.3.2. Зависимые имена типов	154
9.3.3. Зависимые имена шаблонов	156
9.3.4. Зависимые имена в объявлениях using	158
9.3.5. ADL и явные аргументы шаблонов	159
9.4. Наследование и шаблоны классов	160
9.4.1. Независимые базовые классы	160
9.4.2. Зависимые базовые классы	161
9.5. Заключение	164

Глава 10. Инстанцирование	165
10.1. Инстанцирование по требованию	165
10.2. Отложенное инстанцирование	167
10.3. Модель инстанцирования C++	170
10.3.1. Двухфазный поиск	170
10.3.2. Точки инстанцирования	171
10.3.3. Модели включения и разделения	174
10.3.4. Поиск в единицах трансляции	175
10.3.5. Примеры	176
10.4. Схемы реализации	178
10.4.1. “Жадное” инстанцирование	180
10.4.2. Инстанцирование по запросу	181
10.4.3. Итеративное инстанцирование	183
10.5. Явное инстанцирование	186
10.6. Заключение	190
Глава 11. Вывод аргументов шаблонов	193
11.1. Процесс вывода	193
11.2. Выводимый контекст	196
11.3. Особые ситуации вывода	198
11.4. Допустимые преобразования аргументов	199
11.5. Параметры шаблона класса	199
11.6. Аргументы функции по умолчанию	200
11.7. Метод Бартона–Нэкмана	201
11.8. Заключение	203
Глава 12. Специализация и перегрузка	205
12.1. Когда обобщенный код не совсем хорош	205
12.1.1. Прозрачная настройка	206
12.1.2. Семантическая прозрачность	207
12.2. Перегрузка шаблонов функций	208
12.2.1. Сигнатуры	210
12.2.2. Частичное упорядочение перегруженных шаблонов функций	212
12.2.3. Правила формального упорядочения	213
12.2.4. Шаблоны и нешаблоны	215
12.3. Явная специализация	215
12.3.1. Полная специализация шаблона класса	216
12.3.2. Полная специализация шаблона функции	220
12.3.3. Полная специализация члена	222
12.4. Частичная специализация шаблона класса	225
12.5. Заключение	229

Глава 13. Направления дальнейшего развития	231
13.1. Коррекция угловых скобок	231
13.2. Менее строгие правила использования ключевого слова <code>typename</code>	232
13.3. Аргументы шаблонов функций по умолчанию	233
13.4. Строковые литералы и выражения с плавающей точкой в качестве аргументов шаблонов	235
13.5. Менее строгие правила соответствия для шаблонных параметров шаблона	237
13.6. <code>typedef</code> -шаблоны	238
13.7. Частичная специализация шаблонов функций	239
13.8. Оператор <code>typeof</code>	241
13.9. Именованные аргументы шаблонов	242
13.10. Статические свойства	243
13.11. Пользовательская диагностика инстанцирования	244
13.12. Перегруженные шаблоны классов	247
13.13. Параметры-списки	248
13.14. Управление размещением данных	250
13.15. Вывод на основе инициализатора	251
13.16. Функциональные выражения	252
13.17. Заключение	254
Часть III. Шаблоны и конструирование	255
Глава 14. Полиморфные возможности шаблонов	257
14.1. Динамический полиморфизм	257
14.2. Статический полиморфизм	260
14.3. Сравнение динамического и статического полиморфизма	263
14.4. Новые виды шаблонов проектирования	265
14.5. Обобщенное программирование	266
14.6. Заключение	269
Глава 15. Классы свойств и стратегий	273
15.1. Пример: суммирование последовательности	273
15.1.1. Фиксированные классы свойств	274
15.1.2. Свойства-значения	277
15.1.3. Параметризованные свойства	281
15.1.4. Стратегии и классы стратегий	283
15.1.5. Различие между свойствами и стратегиями	285
15.1.6. Шаблоны членов и шаблонные параметры шаблонов	287
15.1.7. Комбинирование нескольких стратегий и/или свойств	289
15.1.8. Накопление с обобщенными итераторами	289
15.2. Функции типа	290

15.2.1. Определение типа элемента	291
15.2.2. Определение типов классов	293
15.2.3. Ссылки и квалификаторы	295
15.2.4. Свойства продвижения	298
15.3. Свойства стратегий	301
15.3.1. Типы параметров только для чтения	302
15.3.2. Копирование, обмен и перемещение	305
15.4. Заключение	310
Глава 16. Шаблоны и наследование	311
16.1. Именованные аргументы шаблона	311
16.2. Оптимизация пустого базового класса	315
16.2.1. Принципы размещения	315
16.2.2. Члены как базовые классы	318
16.3. Модель необычного рекуррентного шаблона	320
16.4. Параметризованная виртуальность	323
16.5. Заключение	324
Глава 17. Метапрограммы	325
17.1. Первый пример метапрограммы	325
17.2. Значения перечислимого типа и статические константы	327
17.3. Второй пример: вычисление квадратного корня	329
17.4. Применение переменных индукции	333
17.5. Полнота вычислений	336
17.6. Рекурсивное инстанцирование и рекурсивные аргументы шаблона	337
17.7. Метапрограммы для развертывания циклов	338
17.8. Заключение	342
Глава 18. Шаблоны выражений	347
18.1. Временные объекты и отдельные циклы	348
18.2. Программирование выражений в аргументах шаблонов	353
18.2.1. Операнды шаблонов выражений	354
18.2.2. Тип <code>Aggr</code>	357
18.2.3. Операторы	359
18.2.4. Подведем итог	361
18.2.5. Присвоение шаблонов выражений	363
18.3. Производительность и ограничения шаблонов выражений	364
18.4. Заключение	365
Часть IV. Нетрадиционное использование шаблонов	369
Глава 19. Классификация типов	371
19.1. Определение фундаментальных типов	371

19.2. Определение составных типов	373
19.3. Определение типов функций	376
19.4. Классификация перечислений с помощью разрешения перегрузки	380
19.5. Определение типов классов	382
19.6. Окончательное решение	383
19.7. Заключение	386
Глава 20. Интеллектуальные указатели	387
20.1. Holder и Trule	387
20.1.1. Защита от исключений	388
20.1.2. Holder	390
20.1.3. Holder в качестве члена класса	392
20.1.4. Захват ресурса есть инициализация	394
20.1.5. Ограничения Holder	394
20.1.6. Копирование Holder	396
20.1.7. Копирование Holder при вызовах функций	397
20.1.8. Trule	397
20.2. Счетчики ссылок	400
20.2.1. Где находится счетчик	401
20.2.2. Параллельный доступ к счетчику	402
20.2.3. Деструкция и освобождение памяти	403
20.2.4. Шаблон CountingPtr	404
20.2.5. Простой незахватывающий счетчик	407
20.2.6. Шаблон простого захватывающего счетчика	409
20.2.7. Константность	410
20.2.8. Неявные преобразования типов	411
20.2.9. Сравнения	414
20.3. Заключение	415
Глава 21. Кортежи	417
21.1. Класс Duo	417
21.2. Рекурсивное вложение объектов класса Duo	422
21.2.1. Количество полей	423
21.2.2. Типы полей	424
21.2.3. Значения полей	425
21.3. Создание класса Tuple	430
21.4. Заключение	435
Глава 22. Объекты-функции и обратные вызовы	437
22.1. Прямые, не прямые и встраиваемые вызовы	438
22.2. Указатели и ссылки на функции	441
22.3. Указатели на функции-члены	444
22.4. Функторы-классы	447

22.4.1. Первый пример функторов-классов	447
22.4.2. Типы функторов-классов	448
22.5. Определение функторов	450
22.5.1. Функторы в роли аргументов типа шаблонов	450
22.5.2. Функторы в роли аргументов функций	451
22.5.3. Сочетание параметров функции и параметров типа шаблона	452
22.5.4. Функторы в роли не являющихся типами аргументов шаблонов	453
22.5.5. Инкапсуляция указателей на функции	454
22.6. Самотестирование	457
22.6.1. Анализ типа функтора	457
22.6.2. Доступ к типам параметров	458
22.6.3. Инкапсуляция указателей на функции	460
22.7. Композиции объектов-функций	465
22.7.1. Простая композиция	466
22.7.2. Композиция разных типов	470
22.7.3. Функторы с несколькими параметрами	473
22.8. Связывание значений	476
22.8.1. Выбор параметров связывания	477
22.8.2. Сигнатура связывания	479
22.8.3. Выбор аргументов	480
22.8.4. Вспомогательные функции	486
22.9. Операции с функторами: полная реализация	489
22.10. Заключение	491
Приложение А. Правило одного определения	493
А.1. Единицы трансляции	493
А.2. Объявления и определения	494
А.3. Детали правила одного определения	495
А.3.1. Ограничения “одно на программу”	495
А.3.2. Ограничения “одно на единицу трансляции”	498
А.3.3. Ограничения эквивалентности единиц перекрестной трансляции	499
Приложение Б. Разрешение перегрузки	505
Б.1. Когда используется разрешение перегрузки	506
Б.2. Упрощенное разрешение перегрузки	506
Б.2.1. Неявный аргумент для функций-членов	508
Б.2.2. Улучшение точного соответствия	510
Б.3. Детали перегрузки	511
Б.3.1. Предпочтение нешаблонных функций	511
Б.3.2. Последовательности преобразований	512
Б.3.3. Преобразования указателей	513
Б.3.4. Функторы и функции-суррогаты	514
Б.3.5. Другие контексты перегрузки	515

Содержание	15
<hr/>	
Библиография	517
Группы новостей	517
Книги и Web-узлы	517
Глоссарий	521
Предметный указатель	532