

Содержание

Предисловие	15
Причины для изучения C++	15
Причины для чтения данной книги	16
Красавица и чудовище	16
Языки в науке и технике	18
Соглашения об оформлении	19
Благодарности	21
Об авторе	23
Ждем ваших отзывов!	24
Глава 1. Основы C++	25
1.1. Наша первая программа	25
1.2. Переменные	28
1.2.1. Константы	30
1.2.2. Литералы	31
1.2.3. Не сужающая инициализация в C++11	33
1.2.4. Области видимости	34
1.3. Операторы	36
1.3.1. Арифметические операторы	37
1.3.2. Булевы операторы	40
1.3.3. Побитовые операторы	41
1.3.4. Присваивание	42
1.3.5. Поток выполнения	42
1.3.6. Работа с памятью	43
1.3.7. Операторы доступа	43
1.3.8. Работа с типами	44
1.3.9. Обработка ошибок	44
1.3.10. Перегрузка	44
1.3.11. Приоритеты операторов	45
1.3.12. Избегайте побочных эффектов!	46
1.4. Выражения и инструкции	48
1.4.1. Выражения	48
1.4.2. Инструкции	48
1.4.3. Ветвление	49
1.4.4. Циклы	52
1.4.5. <code>goto</code>	55
1.5. Функции	56
1.5.1. Аргументы	56
1.5.2. Возврат результатов	58
1.5.3. Встраивание	59
1.5.4. Перегрузка	60
1.5.5. Функция <code>main</code>	62

1.6. Обработка ошибок	63
1.6.1. Утверждения	63
1.6.2. Исключения	65
1.6.3. Статические утверждения	70
1.7. Ввод-вывод	70
1.7.1. Стандартный вывод	70
1.7.2. Стандартный ввод	71
1.7.3. Ввод-вывод в файлы	71
1.7.4. Обобщенная концепция потоков	72
1.7.5. Форматирование	73
1.7.6. Обработка ошибок ввода-вывода	75
1.8. Массивы, указатели и ссылки	78
1.8.1. Массивы	78
1.8.2. Указатели	80
1.8.3. Интеллектуальные указатели	84
1.8.3.1. unique_ptr	84
1.8.4. Ссылки	88
1.8.5. Сравнение указателей и ссылок	88
1.8.6. Не ссылайтесь на устаревшие данные!	89
1.8.7. Контейнеры в качестве массивов	90
1.9. Структурирование программных проектов	92
1.9.1. Комментарии	92
1.9.2. Директивы препроцессора	94
1.10. Упражнения	98
1.10.1. Возраст	98
1.10.2. Массивы и указатели	98
1.10.3. Чтение заголовка файла Matrix Market	99
Глава 2. Классы	101
2.1. Программируйте универсальный смысл, а не технические детали	101
2.2. Члены	103
2.2.1. Переменные-члены	104
2.2.2. Доступность	104
2.2.3. Операторы доступа	107
2.2.4. Декларатор static в классах	108
2.2.5. Функции-члены	108
2.3. Установка значений. Конструкторы и присваивания	110
2.3.1. Конструкторы	110
2.3.2. Присваивание	120
2.3.3. Список инициализаторов	121
2.3.5. Семантика перемещения	125
2.4. Деструкторы	129
2.4.1. Правила реализации	130
2.4.2. Корректная работа с ресурсами	130
2.5. Резюме генерации методов	137
2.6. Доступ к переменным-членам	137

2.6.1. Функции доступа	137
2.6.2. Оператор индекса	139
2.6.3. Константные функции-члены	140
2.6.4. Ссылочная квалификация членов	141
2.7. Проектирование перегрузки операторов	143
2.7.1. Будьте последовательны	143
2.7.2. Вопросы приоритетов	144
2.7.3. Члены или свободные функции	145
2.8. Упражнения	147
2.8.1. Полиномы	147
2.8.2. Перемещающее присваивание	148
2.8.3. Список инициализаторов	148
2.8.4. Спасение ресурса	148
Глава 3. Обобщенное программирование	149
3.1. Шаблоны функций	149
3.1.1. Инстанцирование	150
3.1.2. Вывод типа параметров	152
3.1.3. Работа с ошибками в шаблонах	156
3.1.4. Смешение типов	157
3.1.5. Унифицированная инициализация	158
3.1.6. Автоматический возвращаемый тип	159
3.2. Пространства имен и поиск функций	159
3.2.1. Пространства имен	159
3.2.2. Поиск, зависящий от аргумента	162
3.2.3. Квалификация пространств имен или ADL	166
3.3. Шаблоны классов	168
3.3.1. Пример контейнера	168
3.3.2. Проектирование унифицированных интерфейсов классов и функций	170
3.4. Вывод и определение типа	177
3.4.1. Автоматический тип переменных	177
3.4.2. Тип выражения	178
3.4.3. <code>decltype(auto)</code>	179
3.4.4. Определение типов	180
3.5. Немного теории шаблонов: концепции	182
3.6. Специализация шаблонов	183
3.6.1. Специализация класса для одного типа	183
3.6.2. Специализация и перегрузка функций	186
3.6.3. Частичная специализация	187
3.6.4. Частично специализированные функции	189
3.7. Параметры шаблонов, не являющиеся типами	191
3.8. Функторы	194
3.8.1. Функциональные параметры	196
3.8.2. Составные функторы	197
3.8.3. Рекурсия	199
3.8.4. Обобщенное суммирование	202

3.9. Лямбда-выражения	203
3.9.1. Захват	204
3.9.2. Захват по значению	205
3.9.3. Захват по ссылке	206
3.9.4. Обобщенный захват	207
3.9.5. Обобщенные лямбда-выражения	208
3.10. Вариативные шаблоны	209
3.11. Упражнения	211
3.11.1. Строковое представление	211
3.11.2. Строковое представление кортежей	211
3.11.3. Обобщенный стек	212
3.11.4. Итератор вектора	212
3.11.5. Нечетный итератор	212
3.11.6. Нечетный диапазон	213
3.11.7. Стек <code>bool</code>	213
3.11.8. Стек с пользовательским размером	213
3.11.9. Вывод аргументов шаблона, не являющихся типами	213
3.11.10. Метод трапеций	213
3.11.11. Функтор	214
3.11.12. Лямбда-выражения	214
3.11.13. Реализация <code>make_unique</code>	214
Глава 4. Библиотеки	215
4.1. Стандартная библиотека шаблонов	216
4.1.1. Вводный пример	216
4.1.2. Итераторы	217
4.1.3. Контейнеры	223
4.1.4. Алгоритмы	232
4.1.5. За итераторами	239
4.2. Числовые алгоритмы	240
4.2.1. Комплексные числа	241
4.2.2. Генераторы случайных чисел	244
4.3. Метапрограммирование	256
4.3.1. Пределы	256
4.3.2. Свойства типов	258
4.4. Утилиты	260
4.4.1. <code>tuple</code>	260
4.4.2. <code>function</code>	264
4.4.3. Оболочка для ссылок	266
4.5. Время — сейчас!	267
4.6. Параллельность	270
4.7. Научные библиотеки за пределами стандарта	273
4.7.1. Иная арифметика	273
4.7.2. Арифметика интервалов	274
4.7.3. Линейная алгебра	274
4.7.4. Обычные дифференциальные уравнения	275

4.7.5. Дифференциальные уравнения в частных производных	275
4.7.6. Алгоритмы на графах	275
4.8. Упражнения	276
4.8.1. Сортировка по абсолютной величине	276
4.8.2. Контейнер STL	276
4.8.3. Комплексные числа	276
Глава 5. Метапрограммирование	279
5.1. Пусть считает компилятор	279
5.1.1. Функции времени компиляции	280
5.1.2. Расширенные функции времени компиляции	282
5.1.3. Простота	283
5.1.4. Насколько константны наши константы	285
5.2. Предоставление и использование информации о типах	287
5.2.1. Свойства типов	287
5.2.2. Условная обработка исключений	290
5.2.3. Пример применения константности	291
5.2.4. Стандартные свойства типов	299
5.2.5. Свойства типов, специфичные для предметной области	300
5.2.6. <code>enable_if</code>	301
5.2.7. Еще о вариативных шаблонах	305
5.2.7.1. Вариативный шаблон класса	305
5.3. Шаблоны выражений	308
5.3.1. Реализация простого оператора	309
5.3.2. Класс шаблона выражения	313
5.3.3. Обобщенные шаблоны выражений	315
5.4. Метанастройка: написание собственной оптимизации	317
5.4.1. Классическое развертывание фиксированного размера	319
5.4.2. Вложенное развертывание	322
5.4.3. Динамическое развертывание: разминка	328
5.4.4. Развертывание векторных выражений	330
5.4.5. Настройка шаблона выражения	332
5.4.6. Настройки операций сверток	335
5.4.7. Настройка вложенных циклов	343
5.4.8. Резюме	349
5.5. Упражнения	350
5.5.1. Свойства типов	350
5.5.2. Последовательность Фибоначчи	351
5.5.3. Метапрограммирование НОД	351
5.5.4. Шаблон векторного выражения	351
5.5.5. Метасписок	352
Глава 6. Объектно-ориентированное программирование	353
6.1. Фундаментальные принципы	354
6.1.1. Базовые и производные классы	354
6.1.2. Наследование конструкторов	358

6.1.3. Виртуальные функции и полиморфные классы	359
6.1.4. Функторы и наследование	365
6.2. Устранение избыточности	367
6.3. Множественное наследование	368
6.3.1. Множественные родители	368
6.3.2. Общие прародители	369
6.4. Динамический выбор с использованием подтипов	375
6.5. Преобразования	378
6.5.1. Преобразование между базовыми и производными классами	379
6.5.2. <code>const_cast</code>	383
6.5.3. <code>reinterpret_cast</code>	384
6.5.4. Преобразования в стиле функций	384
6.5.5. Неявные преобразования	386
6.6. CRTP	387
6.6.1. Простой пример	387
6.6.2. Повторно используемый оператор доступа	389
6.7. Упражнения	391
6.7.1. Ромбовидное наследование без избыточности	391
6.7.2. Наследование класса вектора	392
6.7.3. Функция клонирования	392
Глава 7. Научные проекты	393
7.1. Реализация решателей ОДУ	393
7.1.1. Обыкновенные дифференциальные уравнения	394
7.1.2. Алгоритмы Рунге–Кутты	396
7.1.3. Обобщенная реализация	398
7.1.4. Дальнейшее развитие	405
7.2. Создание проектов	406
7.2.1. Процесс построения	406
7.2.2. Инструменты для построения приложений	411
7.2.3. Раздельная компиляция	415
7.3. Несколько заключительных слов	421
Приложение А. Скучные детали	423
А.1. О хорошем и плохом научном программном обеспечении	423
А.2. Детали основ	430
А.2.1. О квалифицирующих литералах	430
А.2.2. Статические переменные	431
А.2.3. Еще немного об <code>if</code>	432
А.2.4. Метод Даффа	434
А.2.5. Еще немного о функции <code>main</code>	434
А.2.6. Утверждения или исключения?	435
А.2.7. Бинарный ввод-вывод	437
А.2.8. Ввод-вывод в стиле C	438
А.2.9. Сборка мусора	439
А.2.10. Проблемы с макросами	440

А.3. Реальный пример: обращение матриц	442
А.4. Больше о классах	453
А.4.1. Указатель на член	453
А.4.2. Примеры инициализации	453
А.4.3. Обращение к многомерным массивам	454
А.5. Генерация методов	457
А.5.1. Управление генерацией	459
А.5.2. Правила генерации	460
А.5.3. Ловушки и советы по проектированию	465
А.6. Подробнее о шаблонах	469
А.6.1. Унифицированная инициализация	469
А.6.2. Какая функция вызвана?	470
А.6.3. Специализация для определенного аппаратного обеспечения	473
А.6.4. Бинарный ввод-вывод с переменным числом аргументов	474
А.7. Использование <code>std::vector</code> в C++03	475
А.8. Динамический выбор в старом стиле	476
А.9. Подробности метапрограммирования	476
А.9.1. Первая метапрограмма в истории	476
А.9.2. Метафункции	478
А.9.3. Обратно совместимые статические утверждения	480
А.9.4. Анонимные параметры типа	481
А.9.5. Проверка производительности динамического развертывания	484
А.9.6. Производительность умножения матриц	485
Приложение Б. Инструментарий для программирования	487
Б.1. <code>gcc</code>	487
Б.2. Отладка	488
Б.2.1. Текстовая отладка	489
Б.2.2. Отладка с графическим интерфейсом: DDD	491
Б.3. Анализ памяти	493
Б.4. <code>gnuplot</code>	494
Б.5. Unix, Linux и Mac OS	496
Приложение В. Определения языка	499
В.1. Категории значений	499
В.2. Обзор операторов	499
В.3. Правила преобразования	502
В.3.1. Повышение	503
В.3.2. Другие преобразования	503
В.3.3. Обычные арифметические преобразования	504
В.3.4. Сужение	505
Библиография	506
Предметный указатель	509