

Содержание

Введение	20
Для кого написана эта книга	20
Что можно найти в этой книге	21
Как построена эта книга	22
Что нужно для работы с этой книгой	22
Соглашения, используемые в этой книге	22
Исходный код	23
Сигнализируйте об ошибках	23
Форумы	24
Посвящение	25
Благодарности	25
Об авторах	25
Ждем ваших отзывов!	26
Часть I. Введение в профессиональное C++-проектирование	27
Глава 1. Краткий курс C++	28
Основы C++	29
Ну, как же обойтись без “Привет, мир!”	29
Пространства имен	32
Переменные	34
Операторы	35
Типы	37
Инструкции и операторы условного выполнения	39
Циклы	42
Массивы	43
Функции	44
Разминка окончена	45
Приготовимся к погружению в C++	45
Указатели и динамическое распределение памяти	46
Строки в C++	49
Ссылки	51
Исключения	51
Использование модификатора const	53
Использование C++ как объектно-ориентированного языка программирования	54
Объявление класса	54
Первая реальная C++-программа	57
Система управления кадрами	57
Класс Employee	57

Класс Database	61
Интерфейс пользователя	65
Оценка программы	68
Резюме с перспективой	68
Глава 2. Разработка профессиональных C++-программ	69
Что такое проектирование программ?	70
Значимость проектирования программ	70
Что особенного в проектировании C++-программ?	73
Два правила C++-проектирования	74
Абстракция	74
Множественное использование кода	76
Проектирование шахматной программы	78
Требования к проекту	78
Этапы проектирования	79
Резюме	84
Глава 3. Проектирование с использованием объектов	85
Объектно-ориентированный взгляд на мир	86
О процедурном мышлении	86
Объектно-ориентированный подход к проектированию	87
Жизнь в мире объектов	89
Отношения между объектами	92
Абстракция	102
Резюме	106
Глава 4. Проектирование с использованием библиотек и шаблонов	107
Множественное использование кода	108
Быть или не быть многократно используемому коду	109
Стратегии использования готового кода	112
Использование приложений от сторонних организаций	117
Открытые библиотеки	118
Стандартная библиотека C++	120
Проектирование с использованием шаблонов и методов	133
Методы проектирования	134
Шаблоны проектов	135
Резюме	137
Глава 5. Проектирование с целью многократного использования кода	138
Принцип многократности использования кода	139
Как спроектировать многократно используемый код	140
Использование абстракций	141
Структурирование кода в расчете на его неоднократное использование	142
Проектирование удобных интерфейсов	147
Поддерживайте баланс между общностью и простотой применения	153
Резюме	154

Глава 6. Использование эффективных методов разработки программного обеспечения	156
Необходимость подчиняться технологическому процессу	157
Модели жизненных циклов разработки ПО	158
Ступенчатая и водопадная модели	158
Спиральный метод	161
Рациональный унифицированный процесс	163
Методологии разработки программного обеспечения	165
Экстремальное программирование (ЭП)	165
Сортировка ПО	170
Построение собственного процесса и методологии	171
Будьте готовы к восприятию новых идей	171
Перенесите новые идеи на бумагу	171
Разберитесь, что работает, а что — нет	171
Не сдавайтесь	172
Резюме	172
Часть II. Пишем C++-код профессионально	173
Глава 7. Кодлируем стильно	174
Красота — страшная сила	174
Думая о будущем...	175
... и настоящем	175
Элементы хорошего стиля	175
Документируйте свой код	175
Зачем писать комментарии	175
Стили комментариев	179
Комментарии, применяемые в этой книге	184
Декомпозиция	184
Декомпозиция посредством переделки	184
Декомпозиция посредством нисходящего проектирования	185
О декомпозиции в этой книге	186
Присвоение имен	186
Выбор хорошего имени	186
Соглашение о присвоении имен	187
Использование языковых средств	190
Применяйте константы	190
Используйте преимущество const-переменных	190
Замените указатели ссылками	190
Используйте собственные исключения	191
Форматирование	191
Договоритесь о размещении фигурных скобок	192
Договоритесь об использовании пробелов и круглых скобок	193
Пробелы и табуляция	193
Проблемы стилистической слаженности	194
Резюме	194

Глава 8. Оттачиваем мастерство в использовании классов и объектов	195
Начнем с примера	196
Создание классов	196
Определение класса	196
Определение методов	199
Использование объектов	202
Жизненные циклы объектов	204
Создание объектов	204
Итак, что мы знаем о генерируемых компилятором конструкторах	214
Разрушение объектов	215
Присваивание объектов	216
Отличие копирования от присваивания	219
Резюме	221
Глава 9. Освоение классов и объектов	222
Динамическое выделение памяти в объектах	222
Класс Spreadsheet	223
Освобождение памяти с помощью деструкторов	224
Обработка операций копирования и присваивания	225
Различные виды членов данных	233
Статические члены данных	233
Константные члены данных	235
Ссылочные члены данных	236
Ссылочные const-члены данных	238
Подробнее о методах	238
Статические методы	238
Константные методы	239
Перегрузка методов	242
Параметры по умолчанию	242
Встраиваемые методы	244
Вложенные классы	245
“Друзья”	247
Перегрузка операторов	248
Реализация оператора сложения	249
Второй вариант: перегруженный метод operator+	250
Перегрузка арифметических операторов	253
Перегрузка операторов сравнения	256
Построение типов с помощью перегрузки операторов	257
Указатели на методы и члены классов	258
Построение абстрактных классов	259
Использование классов интерфейса и реализации	259
Резюме	262
Глава 10. Осваиваем механизм наследования	264
Построение классов с использованием наследования	265
Расширение классов	265
Переопределение методов	269

Наследование как средство многократного использования кода	272
Класс WeatherPrediction	272
Добавление в подкласс функций	273
Уважайте своих родителей	275
Конструкторы родительских классов	276
Деструкторы родительских классов	277
Обращение к данным родительского класса	279
Наследование ради полиморфизма	282
Возвращаясь к электронным таблицам	282
Проектирование полиморфного класса ячейки электронной таблицы	283
Базовый класс ячеек электронной таблицы	284
Подклассы	286
Усиление полиморфизма	288
Размышления о будущем	289
Множественное наследование	291
Наследование нескольких классов	291
Коллизии имен и неоднозначные базовые классы	292
Неоднозначные базовые классы	294
“Подводные рифы” наследования	296
Изменение характеристик переопределенного метода	296
Специальные случаи в переопределении методов	300
Конструкторы копии и оператор равенства	306
Правда о ключевом слове virtual	307
Динамические возможности преобразования типов и получения информации о типе	311
Неоткрытое наследование	312
Виртуальные базовые классы	313
Резюме	314
Глава 11. Пишем обобщенный код с помощью шаблонов	315
Представление о шаблонах	316
Шаблоны классов	317
Построение шаблона класса	317
Как компилятор обрабатывает шаблоны	325
Распределение кода шаблона между файлами	326
Шаблонные параметры	327
Шаблоны методов	330
Специализация шаблонных классов	336
Построение подклассов шаблонных классов	340
Наследование в сравнении со специализацией	341
Шаблоны функций	341
Специализация шаблонов функций	342
Перегрузка шаблонов функций	343
Шаблоны функций-“друзей” в шаблонах классов	344
Расширенные шаблоны	345
Подробнее о шаблонных параметрах	346
Частичная специализация шаблонного класса	355

Эмуляция частичной специализации функций механизмом перегрузки	360
Применение рекурсии к шаблонам	362
Резюме	370
Глава 12. Причуды и странности C++	371
Ссылки	372
Ссылочные переменные	372
Ссылочные члены данных	374
Параметры-ссылки	374
Использование ссылок в качестве значений, возвращаемых функциями или методами	376
Выбор между ссылками и указателями	376
Некоторые особенности ключевых слов	379
Ключевое слово <code>const</code>	379
Ключевое слово <code>static</code>	382
Порядок инициализации нелокальных переменных	385
Типы и использование операций приведения к типу	386
Использование ключевого слова <code>typedef</code>	386
Использование операций приведения к типу	387
Разрешение контекста	392
Заголовочные файлы	393
Утилиты языка C	394
Списки аргументов переменной длины	394
Макроопределения препроцессора	396
Резюме	397
Часть III. Освоение суперсредств C++	399
Глава 13. Эффективное управление памятью	400
Работа с динамической памятью	401
Как представить себе память	401
Выделение памяти и ее освобождение	403
Массивы	405
Работа с указателями	412
Дуализм массивов и указателей	414
Массивы — это те же указатели!	415
Не все указатели являются массивами!	416
Динамическая обработка строк	417
Строки в стиле языка C	417
Строковые литералы	418
C++-класс <code>string</code>	419
Низкоуровневые операции по управлению памятью	421
Арифметика указателей	422
Индивидуальное управление памятью	422
“Сбор мусора”	423
Накопители объектов	424
Указатели на функции	424

Распространенные ошибки при управлении памятью	426
Выделение недостаточного объема памяти для строк	426
Утечка памяти	427
Двойное удаление и использование некорректных указателей	431
Доступ к “заграничной” памяти	431
Резюме	432
Глава 14. Использование C++-потоков ввода-вывода	433
Откуда взялись эти потоки	434
Что такое поток	434
Входные и выходные потоки	434
Вывод данных с помощью потоков	435
Ввод данных с использованием потоков	439
Ввод и вывод объектов	444
Строковые потоки	446
Файловые потоки	447
Использование методов seek() и tell()	448
Связывание потоков	450
Двунаправленные потоки ввода-вывода	451
Локализация	452
“Широкие” символы	453
Использование стандарта Unicode	453
Местная специфика и аспекты локализации	454
Резюме	456
Глава 15. Обработка ошибок	457
Ошибки и исключения	458
Что такое исключения	458
Почему поддержка исключений в C++ — это “плюс”	459
Почему поддержка исключений в C++ — это “минус”	460
Наши рекомендации	461
Механизм исключений	461
Генерирование и перехват исключений	462
Типы исключений	463
Генерирование и перехват множественных исключений	465
Неперехваченные исключения	468
Списки типов генерируемых исключений	469
Исключения и полиморфизм	474
Иерархия стандартных классов исключений	474
Перехват исключений в иерархии классов	475
Создание собственных классов исключений	477
“Раскручивание” и очистка стека	480
Перехват, очистка и повторное генерирование исключений	482
Использование интеллектуальных указателей	482
Распространенные проблемы обработки ошибок	483
Ошибки, связанные с распределением памяти	483
Ошибки в конструкторах	486
Ошибки в деструкторах	487

Теперь соберем все в одну кучу	488
Резюме	490
Часть IV. Как создать код без ошибок	491
Глава 16. Перегрузка C++-операторов	492
Понятие о перегрузке операторов	493
Зачем перегружать операторы	493
Ограничения для перегрузки операторов	493
Рассмотрение альтернатив при перегрузке операторов	494
Операторы, не подлежащие перегрузке	497
Резюме о перегружаемых операторах	497
Перегрузка арифметических операторов	500
Перегрузка унарных операторов “минус” и “плюс”	500
Перегрузка операторов инкремента и декремента	501
Перегрузка поразрядных и бинарных логических операторов	503
Перегрузка операторов ввода-вывода данных	503
Перегрузка оператора индексации	505
Обеспечение с помощью оператора <code>operator[]</code> доступа “только для чтения”	508
Использование для массивов нецелочисленных индексов	510
Перегрузка оператора вызова функций	510
Перегрузка операторов разыменования	512
Реализация оператора <code>operator*</code>	514
Реализация оператора <code>operator-></code>	514
Что это за <code>operator->*</code> ?	515
Создание операторов преобразования	516
Проблемы неоднозначности при использовании операторов преобразования	517
Преобразования для булевых выражений	518
Перегрузка операторов выделения и освобождения памяти	520
Как в действительности работают операторы <code>new</code> и <code>delete</code>	521
Перегрузка операторов <code>new</code> и <code>delete</code>	522
Перегрузка операторов <code>operator new</code> и <code>operator delete</code> с дополнительными параметрами	525
Резюме	527
Глава 17. Создание эффективных C++-программ	528
Немного о производительности и эффективности	529
Два способа достижения эффективности	529
Два вида программ	529
Разве C++ — не эффективный язык программирования?	530
Эффективность на уровне языка	531
Эффективная обработка объектов	531
Не злоупотребляйте дорогостоящими языковыми средствами	535
Использование встраиваемых методов и функций	536
Как позаботиться об эффективности на уровне проектирования	536
Кеш как самое эффективное средство	537

Использование пула (накопителя) объектов	538
Использование пула потоков	543
Протоколирование программ	543
Пример протоколирования программы с помощью средства <i>gprof</i>	544
Резюме	553
Глава 18. Разработка межплатформенных приложений	554
Межплатформенная разработка	555
Проблемы архитектуры	555
Проблемы реализации	558
Средства языка, зависящие от платформы	559
Использование в разработке нескольких языков программирования	560
Смешанное использование языков C и C++	561
Смещение парадигм	561
Компоновка с C-кодом	564
Смешанное выполнение Java- и C++-кода с помощью JNI-интерфейса	565
Объединение C++ с языком Perl и сценариями для оболочки	568
Совместное выполнение C++ и языка ассемблера	571
Резюме	572
Глава 19. Становимся экспертами в области тестирования программ	573
Контроль качества	574
Кто отвечает за тестирование	574
Жизненный цикл ошибок	574
Средства отслеживания ошибок	575
Блочное тестирование	577
Методы поэлементного тестирования	578
Процесс поэлементного тестирования	579
Поэлементное тестирование в действии	583
Тестирование более высокого уровня	592
Комплексные испытания	592
Системные тесты	594
Регрессивные тесты	594
Рекомендации по успешному тестированию	595
Резюме	596
Глава 20. Что нужно знать об отладке	597
Основной закон отладки	598
Систематика ошибок	598
Как избежать попадания ошибок в код	598
Планирование работы над ошибками	599
Регистрация ошибок	599
Трассировка программы	600
Использование макросов <code>assert</code>	612
Методы отладки	613
Репродуцирование ошибок	613
Отладка воспроизводимых ошибок	614

Отладка невоспроизводимых ошибок	615
Отладка ошибок, связанных с управлением памятью	616
Отладка многопоточных программ	620
Пример отладки: поиск цитат	620
Делаем выводы	631
Резюме	632
Часть V. Использование библиотек и шаблонов	633
Глава 21. Библиотека STL: контейнеры и итераторы	634
Обзор контейнеров	635
Исключения и контроль за ошибками	637
Итераторы	637
Последовательные контейнеры	639
Вектор	640
Специализация <code>vector<bool></code>	658
Очередь с двусторонним доступом (дек)	659
Список	659
Контейнеры-адаптеры	663
Очередь	663
Очередь по приоритету	666
Стек	669
Ассоциативные контейнеры	670
Вспомогательный класс <code>pair</code>	670
Отображение	672
Мультиотображение	680
Множество	683
Мультимножество	686
Другие контейнеры	686
Массивы как STL-контейнеры	686
Строки как STL-контейнеры	687
Потоки как STL-контейнеры	688
Битовое множество	688
Резюме	693
Глава 22. Освоение STL-алгоритмов и функциональных объектов	694
Обзор алгоритмов	695
Алгоритмы <code>find()</code> и <code>find_if()</code>	695
Алгоритм <code>accumulate()</code>	698
Функциональные объекты	699
Функциональные объекты арифметических операторов	699
Функциональные объекты операторов сравнения	700
Логические функциональные объекты	702
Адаптеры функциональных объектов	702
Создание собственных функциональных объектов	706
Алгоритмы в деталях	707
Вспомогательные алгоритмы	708
Немодифицирующие алгоритмы	708

Модифицирующие алгоритмы	714
Алгоритмы сортировки	719
Алгоритмы выполнения операций над множествами	721
Пример использования алгоритмов и функциональных объектов: проверка регистрации участников голосования	724
Постановка задачи проверки регистрации участников голосования	724
Функция auditVoterRolls()	724
Функция getDuplicates()	725
Функтор RemoveNames	726
Функтор NameInList	727
Тестирование функции auditVoterRolls()	728
Резюме	729
Глава 23. Использование и расширение возможностей STL	731
Распределители памяти	732
Итераторные адаптеры	732
Реверсивные итераторы	733
Потоковые итераторы	734
Итераторы вставки	735
Расширение библиотеки STL	737
Зачем расширять библиотеку STL	737
Написание STL-алгоритма	737
Написание STL-контейнера	739
Резюме	770
Глава 24. Исследование распределенных объектов	772
В чем притягательность распределенных вычислений	772
Распределение ради расширяемости	773
Распределение ради надежности	774
Распределение ради центрированности	774
Распределенное содержимое	774
Сравнение распределенного приложения с сетевым	774
Распределенные объекты	776
Сериализация и маршalling	776
Удаленные вызовы процедур	780
Архитектура CORBA	782
Язык определения интерфейсов (IDL)	782
Реализация класса	785
Язык XML	789
Ускоренный курс по XML	789
Использование языка XML в качестве технологии распределенных объектов	792
Формирование и анализ XML-кода в C++	793
Аттестация языка XML	801
Построение распределенных объектов с помощью языка XML	803
Протокол SOAP (Simple Object Access Protocol)	806
Резюме	808

Глава 25. Объединим возможности технологий и оболочек	809
“Я все время забываю, как . . .”	810
. . . создать класс	810
. . . вывести подкласс из существующего класса	811
. . . сгенерировать и перехватить исключения	812
. . . считывать данные из файла	813
. . . записать данные в файл	813
. . . создать шаблонный класс	814
Должно быть, есть способ получше	815
Интеллектуальные указатели, выполняющие подсчет ссылок	816
Метод двойной диспетчеризации	821
Смешанные классы	827
Объектно-ориентированные оболочки	830
Работа с оболочками	830
Парадигма “модель-представление-контроллер”	831
Резюме	832
Глава 26. Применение шаблонов проектирования	833
Одноэлементное множество	834
Пример: механизм регистрации	834
Реализация одноэлементного множества	835
Использование одноэлементного множества	839
Фабрика объектов	840
Пример: имитация автомобильного производства	840
Реализация фабрики объектов	842
Использование фабрики объектов	845
Другие ситуации для использования фабрики объектов	846
Шаблон посредника	846
Пример: сокрытие проблем сетевого подключения	847
Реализация шаблона посредника	847
Использование проху-шаблона	848
Шаблон адаптера	848
Пример: адаптация библиотеки XML	849
Реализация адаптера	849
Использование адаптера	853
Шаблон дизайнера	853
Пример: определение стилей в Web-страницах	854
Реализация шаблона дизайнера	854
Использование шаблона дизайнера	856
Шаблон цепочки ответственности	857
Пример: обработка событий	857
Реализация шаблона цепочки ответственности	858
Использование цепочки ответственности	859
Шаблон наблюдателя	859
Пример: обработка событий	859
Реализация наблюдателя	859
Использование наблюдателя	861
Резюме	862

Часть VI. Приложения	863
Приложение А. Готовимся к С++-интервью	864
Глава 1: краткий курс С++	865
О чем не следует забывать	865
Типы вопросов	865
Глава 2: разработка профессиональных С++-программ	866
О чем не следует забывать	866
Типы вопросов	866
Глава 3: проектирование с использованием объектов	866
О чем не следует забывать	866
Типы вопросов	867
Глава 4: проектирование с использованием библиотек и шаблонов	867
О чем не следует забывать	868
Типы вопросов	868
Глава 5: проектирование с целью многократного использования кода	868
О чем не следует забывать	869
Типы вопросов	869
Глава 6: использование эффективных методов разработки программного обеспечения	869
О чем не следует забывать	869
Типы вопросов	869
Глава 7: кодируем стильно	870
О чем не следует забывать	870
Типы вопросов	870
Главы 8 и 9: классы и объекты	871
О чем не следует забывать	871
Типы вопросов	871
Глава 10: осваиваем механизм наследования	874
О чем не следует забывать	874
Типы вопросов	875
Глава 11: пишем обобщенный код с помощью шаблонов	875
О чем не следует забывать	875
Типы вопросов	875
Глава 12: причуды и странности С++	876
О чем не следует забывать	876
Типы вопросов	876
Глава 13: эффективное управление памятью	877
О чем не следует забывать	877
Типы вопросов	877
Глава 14: использование С++-потоков ввода-вывода	878
О чем не следует забывать	878
Типы вопросов	878
Глава 15: обработка ошибок	878
О чем не следует забывать	879
Типы вопросов	879
Глава 16: перегрузка С++-операторов	879
О чем не следует забывать	879

Типы вопросов	879
Глава 17: создание эффективных C++-программ	880
О чем не следует забывать	880
Типы вопросов	880
Глава 18: разработка межплатформенных приложений	881
О чем не следует забывать	881
Типы вопросов	881
Глава 19: становимся экспертами в области тестирования программ	881
О чем не следует забывать	881
Типы вопросов	882
Глава 20: что нужно знать об отладке	882
О чем не следует забывать	882
Типы вопросов	882
Главы 21, 22 и 23: стандартная библиотека шаблонов	882
О чем не следует забывать	883
Типы вопросов	883
Глава 24: исследование распределенных объектов	883
О чем не следует забывать	883
Типы вопросов	883
Глава 25: объединим возможности технологий и оболочек	884
Глава 26: применение шаблонов проектирования	884
О чем не следует забывать	884
Типы вопросов	884
Приложение Б. Аннотированная библиография	885
C++	885
Начальный курс по C++	885
Общий курс по C++	886
Потоки ввода-вывода	887
Стандартная библиотека C++	888
C++-шаблоны	888
Язык С	888
Интеграция C++ и других языков программирования	889
Алгоритмы и структуры данных	889
Открытые программные средства	889
Методология разработки программного обеспечения	890
Стиль программирования	891
Архитектура вычислительных систем	891
Эффективность	891
Тестирование	892
Отладка	892
Распределенные объекты	892
CORBA	892
XML и SOAP	893
Шаблоны проектирования	893
Предметный указатель	894