

Содержание

Введение	15
Об этой книге	15
Использованные в этой книге пиктограммы	17
Дополнительные источники информации	18
Что дальше	18
Ждем ваших отзывов!	19
Часть I. Знакомство с C++	21
Глава 1. Написание вашей первой программы	23
Постигая концепции C++	23
Установка Code::Blocks	25
Windows	25
Ubuntu Linux	27
Macintosh	28
Создание первой программы на C++	32
Создание проекта	32
Ввод кода	32
Небольшая подсказка	35
Построение программы	36
Выполнение программы	37
Анализ программы	38
Определение структуры программ C++	38
Использование в исходном коде комментариев	39
Использование инструкций в программах	39
Объявления	40
Генерация вывода	41
Вычисление выражений	41
Сохранение результатов выражения	41
Обзор программы Convert продолжается...	41
Глава 2. Премудрости объявления переменных	43
Объявление переменных	43
Объявление разных типов переменных	44
Ограничения, налагаемые на целые числа в C++	45
Решение проблемы усечения дробной части	45
Ограничения чисел с плавающей точкой	46
Объявления типов переменных	47
Типы констант	49
Диапазоны числовых типов	50
Специальные символы	51
Широкая дорога для символов	52

Логичны ли эти вычисления?	52
Выражения смешанного типа	53
Автоматические объявления	54
Глава 3. Выполнение математических операций	55
Бинарная арифметика	55
Анализ выражений	56
Определение порядка операций	57
Выполнение унарных операций	58
Использование операторов присваивания	59
Глава 4. Выполнение логических операций	61
Зачем нужны логические операторы	61
Использование простых логических операторов	61
Хранение логических значений	62
Использование целых переменных в качестве логических	64
Логические операции с действительными переменными	64
Бинарные числа в C++	66
Десятичная система счисления	66
Другие системы счисления	66
Двоичная система счисления	67
Выполнение побитовых логических операций	68
Операции с отдельными битами	68
Использование побитовых операторов	70
Простой пример	70
Практическое применение логических вычислений	72
Глава 5. Управление потоком выполнения программы	73
Управление ходом программы с помощью команд ветвления	73
Выполнение циклов	75
Цикл while	75
Использование операторов инкремента и декремента	77
Использование цикла for	78
Избегайте бесконечных циклов	81
Проход по элементам в цикле	81
Специальные операторы управления циклом	82
Вложенные команды управления	85
Инструкция выбора	86
Часть II. Становимся функциональными программистами	89
Глава 6. Создание функций	91
Написание и использование функций	91
Определение первой функции	93
Определение функции <code>sumSequence()</code>	94
Вызов функции <code>sumSequence()</code>	94
Разделяй и властвуй	94

Подробный анализ функций	94
Простые функции	95
Функции с аргументами	96
Перегрузка имен функций	98
Определение прототипов функций	100
Аргументы по умолчанию	101
Передача по значению и передача по ссылке	102
Хранение переменных в памяти	103
Глава 7. Хранение последовательностей в массивах	105
Преимущества массивов	105
Работа с массивами	107
Инициализация массива	110
Выход за границы массива	110
Циклы для диапазонов и массивы	111
Определение и использование массивов с элементами-массивами	111
Использование символьных массивов	112
Создание массива символов	112
Создание строки символов	113
Управление строками	114
Библиотечные функции	116
Работа с широкими строками	117
Глава 8. Первое знакомство с указателями в C++	121
Размер переменной	121
Что такое адрес	122
Адресные операторы	123
Использование указателей	125
Использование разных типов указателей	125
Передача указателей функциям	126
Передача аргументов по значению	126
Передача значений указателей	127
Передача аргументов по ссылке	127
Постоянные проблемы с const	128
Использование кучи	129
Область видимости	130
Проблемы области видимости	131
Решение с применением кучи	131
Глава 9. Второе знакомство с указателями в C++	133
Операции с указателями	133
Повторное знакомство с массивами в свете указателей	134
Операции над указателями для адресации внутри массива	135
Указатели и работа со строками	136
Почему при работе со строками пользуются указателями	138
Операции с указателями других типов	138
Различия между указателями и массивами	139
Когда указатель никуда не указывает	141

Объявление и использование массивов указателей	141
Использование массивов строк	142
Доступ к аргументам <code>main()</code>	144
Глава 10. Препроцессор C++	147
Что такое препроцессор	147
Включение файлов	148
Директива <code>#define</code>	151
Как обойтись без <code>#define</code>	152
Перечисления	153
Использование <code>#if</code>	155
Предопределенные константы	156
Typedef	157
Часть III. Введение в классы	159
Глава 11. Знакомство с объектно-ориентированным программированием	161
Микроволновые печи и уровни абстракции	161
Приготовление блюд с помощью функций	162
Приготовление “объектно-ориентированных” блюд	162
Классификация микроволновых печей	162
Зачем нужна классификация	163
Глава 12. Классы в C++	165
Введение в классы	165
Формат класса	165
Обращение к членам класса	166
Активация наших объектов	167
Моделирование реальных объектов	167
Зачем нужны функции-члены	168
Добавление функции-члена	168
Вызов функций-членов	169
Доступ к членам из функции-члена	171
Разрешение области видимости	172
Определение функции-члена	173
Определение функций-членов вне класса	175
Перегрузка функций-членов	176
Глава 13. Указатели на объекты	179
Объявление массивов объектов	179
Объявление указателей на объекты	180
Разыменование указателей на объекты	181
Использование стрелок	181
Передача объектов функциям	182
Вызов функции с передачей объекта по значению	182
Вызов функции с передачей указателя на объект	183
Передача объекта по ссылке	185
Зачем использовать указатели и ссылки	186

Возврат к куче	186
Выделение из кучи памяти для кучи объектов	187
Управление памятью вместо вас	187
Использование связанных списков	188
Другие операции над связанным списком	189
Программа <code>LinkedListData</code>	189
Списки в стандартной библиотеке	192
Глава 14. Защищенные члены класса: не беспокоить!	193
Защищенные члены	193
Зачем нужны защищенные члены	193
Как устроены защищенные члены	194
Чем хороши защищенные члены	195
Защита внутреннего устройства класса	195
Классы с ограниченным интерфейсом	196
Обращение внешних функций к защищенным членам	196
Глава 15. Создание и удаление объектов	201
Создание объектов	201
Использование конструкторов	202
Конструирование одного объекта	202
Конструирование нескольких объектов	204
Конструирование составных объектов	205
Что такое деструктор	206
Зачем нужен деструктор	206
Работа с деструкторами	207
Глава 16. Аргументация конструирования	211
Как снабдить конструктор аргументами	211
Как использовать конструктор с аргументами	212
Перегрузка конструктора	213
Определение конструкторов по умолчанию	217
Конструирование членов класса	218
Конструирование сложных членов-данных	218
Конструкторы константных членов	223
Управление последовательностью конструирования	224
Локальные объекты создаются последовательно	224
Статические объекты создаются один раз	225
Все глобальные объекты создаются до вызова <code>main()</code>	226
Порядок создания глобальных объектов не определен	226
Члены создаются в порядке их объявления	227
Деструкторы удаляют объекты в порядке, обратном порядку их создания	228
Конструирование массивов	228
Конструкторы как разновидность преобразования типов	229
Глава 17. Копирующий/перемещающий конструктор	231
Копирование объекта	231
Зачем нужен копирующий конструктор	231
Использование копирующего конструктора	232

Автоматический копирующий конструктор	233
“Мелкие” и “глубокие” копии	235
Временные объекты	238
Как избегать временных объектов	240
Перемещающий конструктор	240
Глава 18. Статические члены	243
Определение статических членов	243
Зачем нужны статические члены	243
Использование статических членов	244
Обращение к статическим данным-членам	245
Применение статических данных-членов	246
Объявление статических функций-членов	246
Что такое <code>this</code>	248
Часть IV. Наследование	251
Глава 19. Наследование классов	253
Зачем нужно наследование	254
Как наследуется класс	254
Использование подкласса	256
Конструирование подкласса	257
Деструкция подкласса	258
Наследование конструкторов	258
Отношение СОДЕРЖИТ	259
Глава 20. Знакомство с виртуальными функциями-членами: настоящие ли они	261
Зачем нужен полиморфизм	264
Как работает полиморфизм	264
Когда функция не будет виртуальной	265
Виртуальные особенности	267
Глава 21. Разложение классов	269
Разложение	269
Реализация абстрактных классов	272
Концепция абстрактных классов	273
Создание полноценного класса из абстрактного	274
Передача абстрактных классов	274
Часть V. Безопасность	277
Глава 22. Оператор присваивания	279
Сравнение операторов и функций	279
Потоковые операторы	280
Мелкое копирование — глубокие проблемы	280
Переопределение оператора присваивания	281
Перегрузка оператора индексирования	285
Перемещающие конструктор и оператор присваивания	286

Глава 23. Использование потоков ввода-вывода	291
Как работают потоки ввода-вывода	291
Потоковые объекты по умолчанию	292
Потоковый ввод-вывод	293
Режимы открытия файлов	294
Состояние файла	294
Небольшой пример	295
Методы классов потоков	298
Прямое чтение и запись потока	299
Управление форматированием	301
Что такое endl	302
Позиционирование в файле	303
Подклассы stringstream	303
Работа с манипуляторами	306
Глава 24. Обработка ошибок и исключения	309
Зачем нужен новый механизм обработки ошибок	311
Механизм исключительных ситуаций	311
Что же мы будем бросать?	314
Передай дальше	316
Глава 25. Множественное наследование	319
Механизм множественного наследования	319
Устранение неоднозначностей множественного наследования	321
Виртуальное наследование	322
Конструирование объектов	327
Отрицательные стороны множественного наследования	327
Глава 26. Шаблоны C++	329
Обобщение функции в шаблон	330
Шаблоны классов	331
Советы по использованию шаблонов	334
Внешнее инстанцирование шаблонов	335
Реализация списка инициализации	335
Глава 27. Стандартная библиотека шаблонов	337
Контейнер string	337
Контейнер list	341
Огласите весь список, пожалуйста!	343
Операции над всем списком	344
Приведите пример	344
Глава 28. Как защититься от хакеров	347
Мотивы хакеров	347
Что такое инъекция кода	349
Пример SQL-инъекции	349
Как избежать инъекции кода	351

Переполнение буфера и как им пользуются хакеры	352
Приведите пример!	352
Что происходит со стеком	354
Использование переполнения буфера	358
Первая попытка избежать переполнения буфера	361
Вторая попытка избежать переполнения буфера	363
Еще один аргумент в пользу класса <code>string</code>	365
Почему строковые функции не используются постоянно?	366
Часть VI. Великолепные десятки	369
Глава 29. Десять способов избежать ошибок	371
Включение всех предупреждений и сообщений об ошибках	371
Используйте последовательный стиль программирования	371
Ограничивайте видимость	372
Комментируйте свою программу	373
Хотя бы один раз выполните программу пошагово	374
Избегайте перегрузки операторов	374
Последовательная работа с кучей	374
Для обработки ошибок используйте исключения	374
Делайте деструкторы виртуальными	375
Избегайте множественного наследования	376
Глава 30. Десять способов защититься от взлома	377
Не делайте предположений о вводе пользователя	377
Тщательно обрабатывайте все ошибки	378
Ведение журнала программы	379
Следование определенным правилам разработки	380
Важность контроля версий	381
Безопасная аутентификация пользователей	382
Управление удаленными сеансами	384
Запутывание кода	385
Использование цифровой подписи	387
Используйте безопасное шифрование, где это необходимо	387
Предметный указатель	389