

Содержание

Введение	17
Часть 1. Первое знакомство с C++	23
Глава 1. Написание вашей первой программы	25
Постигая концепции C++	25
Что такое программа	26
Как пишут программы	26
Инсталляция Dev-C++	27
Настройка Dev-C++	30
Создание первой программы	31
Введение кода	32
Построение вашей программы	34
Выполнение программы	36
Dev-C++ — это не Windows	36
Помощь в Dev-C++	36
Разбор программ	36
Определение структуры программ C++	37
Использование в исходном коде комментариев	37
Использование инструкций в программах	38
Объявления	38
Генерирование вывода	39
Вычисление выражений	39
Сохранение результатов выражения	39
Обзор программы <code>Convert</code> продолжается...	40
Глава 2. Премудрости объявления переменных	41
Объявление переменных	41
Объявление разных типов переменных	42
Ограничения, налагаемые на целые числа в C++	43
Округление до целых значений	43
Ограничения диапазона	43
Решение проблемы усечения дробной части	44
Ограничения, налагаемые на числа с плавающей точкой	44
Перечисление	44
Скорость вычислений	45
Потеря точности	45
Ограниченность диапазона	45
Объявления типов переменных	45
Константы	47
Специальные символы	47
Логические выражения	48
Выражения смешанного типа	48
Глава 3. Выполнение математических операций	50
Бинарная арифметика	50
Анализ выражений	51

Определение порядка операций	52
Выполнение унарных операций	53
Использование операторов присвоения	54
Глава 4. Выполнение логических операций	55
Зачем нужны логические операторы	55
Использование простых логических операторов	55
Хранение логических значений	57
Использование целых переменных в качестве логических	58
Логические операции и действительные переменные	58
Сокращенные вычисления в C++	59
Бинарные числа в C++	60
Десятичная система счисления	60
Другие системы счисления	60
Двоичная система счисления	60
Выполнение побитовых логических операций	62
Побитовые операции с одним битом	62
Использование побитовых операторов	63
Простой пример	64
Практическое применение логических вычислений	65
Глава 5. Операторы управления программой	66
Управление ходом программы с помощью команд ветвления	66
Выполнение циклов	68
Цикл while	68
Использование операторов инкремента и декремента	70
Использование цикла for	71
Избегайте бесконечных циклов	73
Специальные операторы управления циклом	73
Вложенные команды управления	76
Инструкция выбора	77
Часть 2. Становимся функциональными программистами	79
Глава 6. Создание функций	81
Написание и использование функций	81
Определение функции <code>sumSequence()</code>	83
Вызов функции <code>sumSequence()</code>	83
Разделяй и властвуй	83
Подробный анализ функций	84
Простые функции	85
Функции с аргументами	85
Функции с одним аргументом	85
Функции с несколькими аргументами	87
Функция <code>main()</code>	87
Перегрузка функций	88
Определение прототипов функций	89
Хранение переменных в памяти	90
Использование заголовочных файлов	91

Глава 7. Хранение последовательностей в массивах	92
Преимущества массивов	92
Работа с массивами	93
Инициализация массива	96
Выход за границы массива	97
Использовать ли массивы	97
Определение и использование массивов с элементами-массивами	98
Использование символьных массивов	98
Создание строки символов	99
Управление строками	100
Тип <code>string</code>	103
Глава 8. Первое знакомство с указателями в C++	105
Размер переменной	105
Что такое адрес	106
Адресные операторы	106
Использование указателей	108
Сравнение указателей и почтовых адресов	109
Использование разных типов указателей	109
Передача указателей функциям	111
Передача аргументов по значению	112
Передача значений указателей	112
Передача аргументов по ссылке	113
Использование кучи	113
Область видимости	113
Проблемы области видимости	114
Использование блока памяти	115
Глава 9. Второе знакомство с указателями	117
Операции с указателями	117
Повторное знакомство с массивами в свете указателей	118
Использование операций над указателями для адресации внутри массива	119
Использование указателей для работы со строками	120
Почему при работе со строками пользуются указателями	122
Операции с указателями других типов	122
Отличия между указателями и массивами	122
Объявление и использование массивов указателей	124
Использование массивов строк	124
Доступ к аргументам <code>main()</code>	126
Аргументы в DOS	127
Аргументы в Dev-C++	127
Аргументы в Windows	127
Глава 10. Отладка программ на C++	128
Определение типа ошибки	128
Использование отладочной печати	128
Выявление “жучка” № 1	130
Выявление “жучка” № 2	131
Использование отладчика	134
Что такое отладчик	134

Работа с отладчиком	134
Запуск тестовой программы	135
Пошаговое выполнение программы	136
Часть 3. Введение в классы	143
Глава 11. Знакомство с объектно-ориентированным программированием	145
Микроволновые печи и уровни абстракции	145
Приготовление блюд с помощью функций	146
Приготовление “объектно-ориентированных” блюд	146
Классификация микроволновых печей	146
Зачем нужна классификация	147
Глава 12. Классы в C++	149
Введение в классы	149
Формат класса	149
Обращение к членам класса	150
Глава 13. Работа с классами	154
Активизация объектов	154
Моделирование реальных объектов	155
Зачем нужны функции-члены	155
Добавление функции-члена	156
Создание функции-члена	156
Именованное членом класса	157
Вызов функций-членов	157
Обращение к функциям-членам	158
Доступ к членам из функции-члена	159
Разрешение области видимости	161
Определение функции-члена	162
Определение функций-членов вне класса	164
Перегрузка функций-членов	165
Глава 14. Указатели на объекты	167
Определение массивов и указателей	167
Объявление массивов объектов	168
Объявление указателей на объекты	169
Разыменованное указателей на объекты	169
Использование стрелок	170
Передача объектов функциям	171
Вызов функции с передачей объекта по значению	171
Вызов функции с передачей указателя	172
Передача объекта по ссылке	173
Зачем использовать указатели и ссылки	174
Возврат к куче	175
Сравнение указателей и ссылок	175
Почему ссылки не используются вместо указателей	175
Использование связанных списков	176
Другие операции над связанным списком	177
Программа LinkedListData	178
Списки в стандартной библиотеке	180

Глава 15. Защищенные члены класса: не беспокоить!	181
Защищенные члены	181
Зачем нужны защищенные члены	181
Как устроены защищенные члены	182
Чем хороши защищенные члены	183
Защита внутреннего устройства класса	183
Классы с ограниченным интерфейсом	184
Обращение к защищенным членам	184
Глава 16. Создание и удаление объектов	188
Создание объектов	188
Использование конструкторов	189
Зачем нужны конструкторы	189
Работа с конструкторами	190
Что такое деструктор	194
Зачем нужен деструктор	194
Работа с деструкторами	194
Глава 17. Аргументация конструирования	198
Как снабдить конструктор аргументами	198
Зачем конструкторам нужны аргументы	198
Как использовать конструктор с аргументами	199
Перегрузка конструктора	200
Определение конструкторов по умолчанию	203
Конструирование членов класса	204
Конструкторы константных членов	208
Управление последовательностью конструирования	208
Локальные объекты создаются последовательно	209
Статические объекты создаются один раз	209
Все глобальные объекты создаются до вызова <code>main()</code>	210
Порядок создания глобальных объектов не определен	210
Члены создаются в порядке их объявления	211
Деструкторы удаляют объекты в порядке, обратном порядку их создания	212
Глава 18. Копирующий конструктор	213
Копирование объекта	213
Зачем нужен копирующий конструктор	213
Использование конструктора копирования	214
Автоматический конструктор копирования	215
“Мелкие” и “глубокие” копии	217
Временные объекты	221
Как избегать временных объектов	222
Аргумент копирующего конструктора	223
Глава 19. Статические члены	224
Определение статических членов	224
Зачем нужны статические члены	224
Использование статических членов	225
Обращение к статическим данным-членам	226
Применение статических данных-членов	227

Объявление статических функций-членов	228
Что такое <code>this</code>	230
Часть 4. Наследование	231
Глава 20. Наследование классов	233
Зачем нужно наследование	234
Как наследуется класс	234
Использование подкласса	236
Конструирование подкласса	237
Деструкция подкласса	238
Отношение СОДЕРЖИТ	238
Глава 21. Знакомство с виртуальными функциями-членами: настоящие ли они	240
Зачем нужен полиморфизм	243
Как работает полиморфизм	245
Когда функция не является виртуальной	246
Виртуальные особенности	247
Глава 22. Разложение классов	249
Разложение	249
Реализация абстрактных классов	253
Концепция абстрактных классов	254
Создание полноценного класса из абстрактного	255
Передача абстрактных классов	257
Нужны ли чисто виртуальные функции	257
Разделение исходного кода C++	259
Разделение программы — класс <code>Student</code>	260
Определение пространства имен	261
Реализация класса <code>Student</code>	261
Разделение программы — класс <code>GraduateStudent</code>	262
Реализация приложения	263
Файл проекта	264
Создание файла проекта в Dev-C++	265
Часть 5. Полезные особенности	269
Глава 23. Оператор присвоения	271
Сравнение операторов и функций	271
Потоковые операторы	272
Мелкое копирование — глубокие проблемы	272
Переопределение оператора присвоения	273
Защита от копирования	276
Глава 24. Использование потоков ввода-вывода	277
Как работают потоки ввода-вывода	277
Знакомство с подклассами <code>fstream</code>	278
Прямое чтение из потока	282
Что такое <code>endl</code>	284
Подклассы <code>stringstream</code>	285
Работа с манипуляторами	287

Глава 25. Обработка ошибок и исключения	290
Зачем нужен новый механизм обработки ошибок	291
Механизм исключительных ситуаций	293
Так что же мы будем бросать?	295
Глава 26. Множественное наследование	298
Механизм множественного наследования	298
Устранение неоднозначностей множественного наследования	300
Виртуальное наследование	301
Конструирование объектов	306
Отрицательные стороны множественного наследования	306
Глава 27. Шаблоны C++	308
Обобщение функции в шаблон	309
Шаблоны классов	311
Зачем нужны шаблоны классов	314
Советы по использованию шаблонов	316
Глава 28. Стандартная библиотека шаблонов	317
Контейнер <code>string</code>	317
Контейнер <code>list</code>	320
Итераторы	321
Использование контейнера <code>map</code>	324
Часть 6. Великолепная десятка	329
Глава 29. Десять способов избежать ошибок	331
Включение всех предупреждений и сообщений об ошибках	331
Добейтесь чистой компиляции	332
Используйте последовательный стиль программирования	332
Ограничивайте видимость	332
Комментируйте свою программу	334
Хотя бы один раз выполните программу пошагово	334
Избегайте перегрузки операторов	334
Работа с кучей	334
Используйте для обработки ошибок исключительные ситуации	335
Избегайте множественного наследования	335
Глава 30. Десять основных возможностей Dev-C++	336
Настройка редактора по вашему вкусу	336
Подсветка парных скобок	337
Включение обработки исключений	337
Включение отладочной информации	338
Создание файла проекта	338
Настройка справки	338
Переустановка точек останова после редактирования файла	339
Избегайте некорректных имен файлов	339
Включите заголовочные файлы в ваш проект	339
Работа с профайлером	339

Глава 31. Программа BUDGET	343
BUDGET1	343
BUDGET2	348
BUDGET3	355
Реализация модуля со связанным списком	356
Работа со счетами	358
Классы связанных списков	363
Оценка бюджета	365
BUDGET4	366
Реализация связанного списка в виде шаблона класса	366
Исходный код BUDGET4	368
Подведение итогов	373
BUDGET5	373
Использование шаблона класса из STL	373
Создание списка счетов	378
Приложение. Содержимое прилагаемого компакт-диска	379
Предметный указатель	380