

Х.М. Дейтел, П.Дж. Дейтел

КАК ПРОГРАММИРОВАТЬ НА

C

седьмое издание

Перевод
с английского
под редакцией
С.Н. Башникова



Москва
Издательство **БИНОМ**
2017

Дейтел, Х. М. Как программировать на С / Х. М. Дейтел, П. Дж. Дейтел ; пер. с англ. под ред. С. Н. Банникова. — 7-е изд. — Москва : БИНОМ, 2017. — 1000 с. : ил. — Парал. тит. англ. — Предм. указ. : с. 985—1000.

УДК 004.438С:004.42

ББК 32

Чит. зал № 1 — 1 экз.

Книга является общепризнанным руководством для изучения языка С, который широко распространен на различных платформах, включая Windows и UNIX/Linux. Приводится полное описание языка, ориентированное на его практическое применение (с учетом стандарта ISO/IEC 9899:2011 или СП). Кроме того, дополнительно излагается С++ — язык, являющийся логическим развитием С в сторону современных методологий программирования, таких, как объектно-ориентированное и обобщенное (на основе шаблонов) программирование.

Весь изучаемый материал сопровождается содержательными примерами «живого кода», позволяющими читателю детально разобраться в обсуждаемом вопросе и демонстрирующими широко применяемые приемы программирования, алгоритмы и структуры данных. В конце каждой главы даются рекомендации по безопасному программированию, предлагаются контрольные вопросы и упражнения для самостоятельного решения. Изложение построено таким образом, что книга позволяет изучать С/С++ в качестве первого языка программирования.

Адресована прежде всего изучающим программирование в учебных заведениях, но может с успехом использоваться и теми, кто хочет освоить языки С/С++ самостоятельно.

| | |
|--|-----------|
| Предисловие | 14 |
| Глава 1. Введение в компьютеры, Internet и World Wide Web | 21 |
| 1.1. Введение | 22 |
| 1.2. Что такое компьютер? | 23 |
| 1.3. Внутренняя организация компьютера | 24 |
| 1.4. Персональные вычисления, распределенные вычисления и вычисления в модели клиент/сервер | 25 |
| 1.5. Интернет и Всемирная паутина | 25 |
| 1.6. Машинные языки, языки ассемблера и языки высокого уровня | 26 |
| 1.7. История языка C | 27 |
| 1.8. Стандартная библиотека C | 28 |
| 1.9. C++ | 29 |
| 1.10. Java | 29 |
| 1.11. FORTRAN, COBOL, Pascal и Ada | 30 |
| 1.12. BASIC, Visual Basic, Visual C++, C# и .NET | 30 |
| 1.13. Ключевая тенденция в программировании: объектная технология | 31 |
| 1.14. Схема типичной среды разработки C | 32 |
| 1.15. Тенденции в области аппаратных средств | 35 |
| 1.16. Общие замечания о C и этой книге | 36 |
| Глава 2. Введение в программирование на C | 43 |
| 2.1. Введение | 44 |
| 2.2. Простая программа на C: печать строки текста | 44 |
| 2.3. Еще одна простая программа на C: сложение двух целых чисел. | 48 |
| 2.4. Общие понятия о памяти компьютера. | 53 |
| 2.5. Арифметика в C | 54 |
| 2.6. Принятие решений: операции равенства и отношения | 58 |
| Глава 3. Структурная разработка программ на C | 71 |
| 3.1. Введение | 72 |
| 3.2. Алгоритмы | 72 |
| 3.3. Псевдокод | 73 |
| 3.4. Управляющие структуры. | 74 |
| 3.5. Оператор выбора if | 76 |
| 3.6. Оператор выбора if...else | 77 |
| 3.7. Оператор повторения while | 81 |
| 3.8. Формулирование алгоритмов: пример 1 (повторение, управляемое счетчиком) | 82 |

| | |
|--|------------|
| 3.9. Формулирование алгоритмов методом нисходящего последовательного уточнения: пример 2 (повторение, управляемое контрольным значением) | 84 |
| 3.10. Формулирование алгоритмов методом нисходящего последовательного уточнения: пример 3 (вложенные управляющие структуры) | 91 |
| 3.11. Операции присваивания | 95 |
| 3.12. Операции инкремента и декремента | 96 |
| Глава 4. Управление программой в C | 113 |
| 4.1. Введение | 114 |
| 4.2. Основы структур повторения | 114 |
| 4.3. Повторение, управляемое счетчиком | 115 |
| 4.4. Оператор повторения for | 117 |
| 4.5. Оператор for: замечания и рекомендации | 119 |
| 4.6. Примеры структур for | 120 |
| 4.7. Оператор множественного выбора switch | 124 |
| 4.8. Оператор повторения do...while | 129 |
| 4.9. Операторы break и continue | 131 |
| 4.10. Логические операции | 133 |
| 4.11. Смешивание операций равенства (==) и присваивания (=). | 136 |
| 4.12. Резюме структурного программирования | 137 |
| Глава 5. Функции в C | 153 |
| 5.1. Введение | 154 |
| 5.2. Программные модули в C. | 154 |
| 5.3. Функции математической библиотеки | 155 |
| 5.4. Функции | 157 |
| 5.5. Определения функций | 157 |
| 5.6. Прототипы функций | 162 |
| 5.7. Стек вызовов и активационные записи | 164 |
| 5.8. Заголовочные файлы | 165 |
| 5.9. Вызов функций: вызов по значению и вызов по ссылке | 166 |
| 5.10. Генерация случайных чисел | 166 |
| 5.11. Пример: стохастическая игра | 172 |
| 5.12. Классы памяти | 175 |
| 5.13. Правила области действия | 178 |
| 5.14. Рекурсия | 181 |
| 5.15. Пример применения рекурсии: числа Фибоначчи | 185 |
| 5.16. Рекурсия в сравнении с итерацией | 188 |
| Глава 6. Массивы в C | 207 |
| 6.1. Введение | 208 |
| 6.2. Массивы | 208 |
| 6.3. Определение массивов | 210 |
| 6.4. Примеры действий над массивами | 210 |
| 6.5. Передача массивов функциям | 224 |
| 6.6. Сортировка массивов | 229 |
| 6.7. Пример: вычисление среднего, медианы и наиболее вероятного значения массива | 231 |
| 6.8. Поиск в массивах | 236 |
| 6.9. Многомерные массивы | 241 |
| Глава 7. Указатели в C | 263 |
| 7.1. Введение | 264 |
| 7.2. Определение и инициализация переменных-указателей | 264 |

| | |
|---|------------|
| 7.3. Операции над указателями | 265 |
| 7.4. Вызов по ссылке | 268 |
| 7.5. Использование квалификатора const с указателями | 272 |
| 7.6. Пузырьковая сортировка, использующая вызов по ссылке | 278 |
| 7.7. Операция sizeof | 282 |
| 7.8. Выражения и арифметические операции с указателями | 284 |
| 7.9. Связь между указателями и массивами | 287 |
| 7.10. Массивы указателей | 291 |
| 7.11. Пример: программа тасования и сдачи карт | 292 |
| 7.11. Указатели на функции | 297 |
| Глава 8. Символы и строки в C | 320 |
| 8.1. Введение | 321 |
| 8.2. Элементарные сведения о строках и символах | 321 |
| 8.3. Библиотека обработки символов | 323 |
| 8.4. Функции преобразования строк | 329 |
| 8.5. Функции стандартной библиотеки ввода/вывода | 333 |
| 8.6. Функции операций над строками из библиотеки обработки строк | 337 |
| 8.7. Функции сравнения из библиотеки обработки строк | 340 |
| 8.8. Функции поиска из библиотеки обработки строк | 342 |
| 8.9. Функции памяти из библиотеки обработки строк | 348 |
| 8.10. Другие функции из библиотеки обработки строк | 352 |
| Глава 9. Форматированный ввод/вывод в C | 364 |
| 9.1. Введение | 365 |
| 9.2. Потoki | 365 |
| 9.3. Форматированный вывод функцией printf | 366 |
| 9.4. Печать целых чисел | 366 |
| 9.5. Печать чисел с плавающей точкой | 368 |
| 9.6. Печать строк и символов | 370 |
| 9.7. Другие спецификаторы преобразования | 371 |
| 9.8. Печать с заданием ширины поля и точности представления | 372 |
| 9.9. Использование флагов в строке управления форматом printf | 375 |
| 9.10. Печать литералов и esc-последовательностей | 378 |
| 9.11. Форматированный ввод функцией scanf | 378 |
| Глава 10. Структуры, объединения, перечисления и операции с битами в C | 392 |
| 10.1. Введение | 393 |
| 10.2. Определение структур | 393 |
| 10.3. Инициализация структур | 396 |
| 10.4. Доступ к элементам структур | 396 |
| 10.5. Использование структур с функциями | 398 |
| 10.6. Ключевое слово typedef | 398 |
| 10.7. Пример: моделирование высокоэффективного тасования и сдачи карт | 399 |
| 10.8. Объединения | 402 |
| 10.9. Поразрядные операции | 404 |
| 10.10. Битовые поля | 413 |
| 10.11. Перечисляемые константы | 416 |
| Глава 11. Работа с файлами в C | 426 |
| 11.1. Введение | 427 |
| 11.2. Иерархия данных | 427 |
| 11.3. Файлы и потоки | 429 |

| | | |
|------------------|--|------------|
| 11.4. | Создание файла последовательного доступа | 430 |
| 11.5. | Чтение данных из файла последовательного доступа | 435 |
| 11.6. | Файлы произвольного доступа | 440 |
| 11.7. | Создание файла произвольного доступа | 441 |
| 11.8. | Произвольная запись данных в файл произвольного доступа | 443 |
| 11.9. | Чтение данных из файла произвольного доступа. | 445 |
| 11.10. | Пример: программа обработки транзакций. | 447 |
| Глава 12. | Структуры данных в C | 461 |
| 12.1. | Введение | 462 |
| 12.2. | Автореферентные структуры | 463 |
| 12.3. | Динамическое распределение памяти | 464 |
| 12.4. | Связанные списки | 465 |
| 12.5. | Стеки. | 474 |
| 12.6. | Очереди | 479 |
| 12.7. | Деревья | 485 |
| Глава 13. | Препроцессор C | 513 |
| 13.1. | Введение | 514 |
| 13.2. | Директива препроцессора #include | 514 |
| 13.3. | Директива #define: символические константы | 515 |
| 13.4. | Директива #define: макросы | 515 |
| 13.5. | Условная компиляция | 517 |
| 13.6. | Директивы #error и #pragma. | 518 |
| 13.7. | Операции # и ## | 518 |
| 13.8. | Нумерация строк | 519 |
| 13.9. | Предопределенные символические константы. | 519 |
| 13.10. | Макрос подтверждения | 520 |
| Глава 14. | Специальные вопросы языка C. | 524 |
| 14.1. | Введение | 525 |
| 14.2. | Переадресация ввода/вывода | 525 |
| 14.3. | Списки аргументов переменной длины | 526 |
| 14.4. | Аргументы командной строки. | 529 |
| 14.5. | Замечания относительно компиляции программ из нескольких исходных файлов | 530 |
| 14.6. | Выход из программы с помощью exit и atexit | 532 |
| 14.7. | Квалификатор типа volatile | 533 |
| 14.8. | Суффиксы для целых констант и констант с плавающей точкой | 533 |
| 14.9. | Еще раз о файлах | 534 |
| 14.10. | Обработка сигналов | 536 |
| 14.11. | Динамическое выделение памяти: функции calloc и realloc | 538 |
| 14.12. | Безусловный переход: goto. | 539 |
| Глава 15. | C++ как «улучшенный C. | 545 |
| 15.1. | Введение | 546 |
| 15.2. | C++ | 546 |
| 15.3. | Простая программа: сложение двух целых | 547 |
| 15.4. | Стандартная библиотека C++ | 549 |
| 15.5. | Заголовочные файлы | 550 |
| 15.6. | Inline-функции | 552 |
| 15.7. | Ссылки и параметры-ссылки | 554 |
| 15.8. | Пустые списки параметров. | 559 |
| 15.9. | Аргументы по умолчанию | 559 |
| 15.10. | Унарная операция разрешения области действия | 561 |

| | |
|--|------------|
| 15.11. Перегрузка функций | 562 |
| 15.12. Шаблоны функций | 565 |
| 15.13. Введение в объектную технологию и UML | 567 |
| 15.14. Заключение | 571 |
| Глава 16. Введение в классы и объекты | 576 |
| 16.1. Введение | 577 |
| 16.2. Классы, объекты, элемент-функции и элементы данных | 577 |
| 16.3. Определение класса с элемент-функцией | 579 |
| 16.4. Определение элемент-функции с параметром | 582 |
| 16.5. Элементы данных, <i>set</i> -функции и <i>get</i> -функции | 585 |
| 16.6. Инициализация объектов при помощи конструкторов | 592 |
| 16.7. Размещение класса в отдельном файле | 596 |
| 16.8. Отделение интерфейса от реализации | 600 |
| 16.9. Верификация данных посредством <i>set</i> -функций | 606 |
| 16.10. Заключение | 611 |
| Глава 17. Классы: часть 1 | 618 |
| 17.1. Введение | 619 |
| 17.2. Пример: класс <i>Time</i> | 620 |
| 17.3. Область действия класса и доступ к элементам класса | 627 |
| 17.4. Отделение интерфейса от реализации | 629 |
| 17.5. Функции доступа и сервисные функции | 630 |
| 17.6. Пример: класс <i>Time</i> . Конструкторы с аргументами по умолчанию | 633 |
| 17.7. Деструкторы | 638 |
| 17.8. Когда вызываются конструкторы и деструкторы | 639 |
| 17.9. Пример: класс <i>Time</i> . Скрытая ошибка — возвращение ссылки на закрытый элемент данных | 642 |
| 17.10. Поэлементное присваивание по умолчанию | 645 |
| 17.13. Заключение | 647 |
| Глава 18. Классы: часть 2. | 654 |
| 18.1. Введение | 655 |
| 18.2. Константные объекты и константные элемент-функции | 655 |
| 18.3. Композиция: объекты в качестве элементов класса | 665 |
| 18.4. Дружественные функции и дружественные классы | 671 |
| 18.5. Указатель <i>this</i> | 673 |
| 18.6. Статические элементы класса | 679 |
| 18.7. Абстракция данных и сокрытие информации | 683 |
| 18.8. Заключение | 685 |
| Глава 19. Перегрузка операций; объекты <i>Array</i> и <i>String</i> | 690 |
| 19.1. Введение | 691 |
| 19.2. Основы перегрузки операций | 692 |
| 19.3. Ограничения на перегрузку операций | 693 |
| 19.4. Функции-операции как элементы класса и как глобальные функции | 695 |
| 19.5. Перегрузка операций передачи в поток и извлечения из потока | 696 |
| 19.6. Перегрузка одноместных операций | 700 |
| 19.7. Перегрузка двухместных операций | 700 |
| 19.6. Динамическое управление памятью с помощью операций <i>new</i> и <i>delete</i> | 701 |
| 19.9. Пример: класс <i>Array</i> | 703 |
| 19.10. Преобразование типов | 715 |
| 19.11. Пример: класс <i>String</i> | 716 |

| | |
|--|-----|
| 19.12. Перегрузка ++ и -- | 728 |
| 19.13. Пример: класс Date | 730 |
| 19.14. Класс string стандартной библиотеки | 735 |
| 19.15. explicit-конструкторы | 738 |
| 19.16. Классы-посредники | 742 |
| 19.17. Заключение | 745 |

Глава 20. Объектно-ориентированное программирование:

| | |
|---|------------|
| наследование | 758 |
| 20.1. Введение | 759 |
| 20.2. Базовые и производные классы | 761 |
| 20.3. Защищенные элементы | 764 |
| 20.4. Отношения между базовыми и производными классами | 764 |
| 20.4.1. Создание и тестирование класса CommissionEmployee | 765 |
| 20.4.2. Создание класса BasePlusCommissionEmployee без наследования | 770 |
| 20.4.3. Создание иерархии наследования CommissionEmployee — BasePlusCommissionEmployee | 775 |
| 20.4.4. Иерархия наследования CommissionEmployee — BasePlusCommissionEmployee с защищенными данными | 780 |
| 20.4.5. Иерархия наследования CommissionEmployee — BasePlusCommissionEmployee с закрытыми данными | 787 |
| 20.5. Конструкторы и деструкторы в производных классах | 795 |
| 20.6. Открытое, защищенное и закрытое наследование | 802 |
| 20.7. Наследование в конструировании программного обеспечения | 803 |
| 20.8. Заключение | 805 |

Глава 21. Объектно-ориентированное программирование: полиморфизм 810

| | |
|--|-----|
| 21.1. Введение | 812 |
| 21.2. Примеры полиморфизма | 813 |
| 21.3. Отношения между объектами в иерархии наследования | 814 |
| 21.3.1. Вызов функций базового класса из объектов производного класса | 815 |
| 21.3.2. Установка указателей производного класса на объекты базового класса | 822 |
| 21.3.3. Вызов элемент-функций производного класса через указатели базового класса | 823 |
| 21.3.4. Виртуальные функции | 825 |
| 21.3.5. Сводка допустимых присваиваний объектов указателям базового и производного классов | 831 |
| 21.4. Поля типа и операторы switch | 832 |
| 21.5. Абстрактные классы и чисто виртуальные функции | 832 |
| 21.6. Пример. Система начисления заработной платы, использующая полиморфизм | 835 |
| 21.6.1. Создание абстрактного базового класса Employee | 836 |
| 21.6.2. Создание конкретного производного класса SalariedEmployee | 840 |
| 21.6.3. Создание конкретного производного класса HourlyEmployee | 842 |
| 21.6.4. Создание конкретного производного класса CommissionEmployee | 844 |
| 21.6.5. Создание косвенного конкретного производного класса BasePlusCommissionEmployee | 846 |
| 21.6.6. Демонстрация полиморфной обработки | 848 |
| 21.7. Дополнительный раздел. Техническая сторона полиморфизма, виртуальных функций и динамического связывания | 852 |
| 21.8. Пример. Система начисления заработной платы, использующая полиморфизм и информацию о типе времени выполнения с нисходящими приведениями типа, dynamic_cast, typeid и type_info | 856 |

| | |
|--|------------|
| 21.9. Виртуальные деструкторы | 860 |
| 21.10. Заключение | 860 |
| Глава 22. Шаблоны | 866 |
| 22.1. Введение | 867 |
| 22.2. Шаблоны функций | 868 |
| 22.3. Перегрузка шаблонов функции | 871 |
| 22.4. Шаблоны классов | 871 |
| 22.5. Нетиповые параметры и типы по умолчанию для шаблонов класса. | 878 |
| 22.6. Замечания о шаблонах и наследовании | 879 |
| 22.7. Замечания о шаблонах и друзьях | 879 |
| 22.8. Замечания о шаблонах и статических элементах | 880 |
| 22.9. Заключение | 880 |
| Глава 23. Поточковый ввод/вывод | 885 |
| 23.1. Введение | 887 |
| 23.2. Потoki | 887 |
| 23.2.1. Классические и стандартные потоки. | 888 |
| 23.2.2. Заголовочные файлы библиотеки <code>iostream</code> | 888 |
| 23.2.3. Классы и объекты потокового ввода/вывода | 889 |
| 23.3. Поточковый вывод | 891 |
| 23.3.1. Вывод переменных типа <code>char *</code> | 892 |
| 23.3.2. Вывод символов с помощью элемент-функции <code>put</code> | 892 |
| 23.4. Поточковый ввод | 893 |
| 23.4.1. Элемент-функции <code>get</code> и <code>getline</code> | 893 |
| 23.4.2. Элемент-функции <code>peek</code> , <code>putback</code> и <code>ignore</code> класса <code>istream</code> | 896 |
| 23.4.3. Безопасный по типу ввод/вывод | 896 |
| 23.5. Бесформатный ввод/вывод с помощью <code>read</code> , <code>gcount</code> и <code>write</code> | 896 |
| 23.6. Введение в манипуляторы потоков | 898 |
| 23.6.1. Основание целых чисел: <code>dec</code> , <code>oct</code> , <code>hex</code> и <code>setbase</code> | 898 |
| 23.6.2. Точность чисел с плавающей точкой (<code>precision</code> , <code>setprecision</code>) | 899 |
| 23.6.3. Ширина поля (<code>setw</code> , <code>width</code>) | 900 |
| 23.6.4. Определяемые пользователем манипуляторы выходного потока. | 902 |
| 23.7. Состояния формата потока и потоковые манипуляторы | 903 |
| 23.7.1. Конечные нули и десятичные точки (<code>showpoint</code>) | 904 |
| 23.7.2. Выравнивание (<code>left</code> , <code>right</code> , <code>internal</code>) | 905 |
| 23.7.3. Заполнение (<code>fill</code> , <code>setfill</code>) | 907 |
| 23.7.4. Основание целых чисел (<code>dec</code> , <code>oct</code> , <code>hex</code> , <code>showbase</code>) | 908 |
| 23.7.5. Числа с плавающей точкой: научная и фиксированная нотация (<code>scientific</code> , <code>fixed</code>) | 909 |
| 23.7.6. Управление верхним/нижним регистрами (<code>uppercase</code>) | 910 |
| 23.7.7. Спецификация булева формата (<code>boolalpha</code>) | 911 |
| 23.7.8. Установка и сброс состояний формата с помощью элемент-функции <code>flags</code> | 912 |
| 23.8. Состояния ошибки потоков | 913 |
| 23.9. Привязка потока вывода к потоку ввода | 915 |
| Глава 24. Управление исключениями | 925 |
| 24.1. Введение | 926 |
| 24.2. Обзор управления исключениями | 927 |
| 24.3. Пример: обработка попытки деления на ноль | 928 |
| 24.4. Когда следует применять управление исключениями | 934 |
| 24.5. Перебрасывание исключений | 935 |
| 24.6. Спецификации исключений | 937 |
| 24.7. Обработка непредусмотренных исключений | 938 |

| | |
|---|------------|
| 24.8. Разматывание стека | 938 |
| 24.9. Конструкторы, деструкторы и управление исключениями | 940 |
| 24.10. Исключения и наследование | 941 |
| 24.11. Обработка отказов операции new | 941 |
| 24.12. Класс auto_ptr и динамическое выделение памяти | 945 |
| 24.13. Иерархия исключений стандартной библиотеки | 948 |
| 24.14. Другие методы обработки ошибок | 950 |
| 24.15. Заключение | 950 |
| Приложение А. Таблицы старшинства и ассоциативности операций | 957 |
| Приложение Б. Набор символов ASCII | 961 |
| Приложение В. Системы счисления | 962 |
| В.1. Введение | 963 |
| В.2. Сокращенная запись двоичных чисел в восьмеричной и шестнадцатеричной системах счисления | 966 |
| В.3. Преобразование восьмеричных и шестнадцатеричных чисел в двоичные | 968 |
| В.4. Преобразование двоичных, восьмеричных и шестнадцатеричных чисел в десятичные | 968 |
| В.5. Преобразование десятичных чисел в двоичные, восьмеричные и шестнадцатеричные | 969 |
| В.6. Представление отрицательных двоичных чисел: дополнение до двух | 970 |
| Приложение Г. Безопасное программирование | 976 |
| Предметный указатель | 985 |