

# Содержание

Предисловие	21
Об авторе	22
Посвящение	22
Благодарности	22
От издательства	22

## **Глава 1. Предварительные сведения** **23**

Как появился язык C?	23
Почему язык C?	24
Конструктивные особенности	25
Эффективность	25
Переносимость	25
Мощь и гибкость	26
Ориентация на программистов	26
Недостатки	26
Откуда пошел язык C?	27
Как работают компьютеры	28
Языки программирования высокого уровня и компиляторы	29
Использование языка C: семь этапов	30
Этап 1: определение целей программы	31
Этап 2: проектирование программы	31
Этап 3: написание кода	32
Этап 4: компиляция	32
Этап 5: запуск программы на выполнение	33
Этап 6: тестирование и отладка программы	33
Этап 7: сопровождение и модификация программы	34
Комментарий	34
Механика программирования	35
Файлы объектного кода, исполняемые файлы и библиотеки	36
Операционная система Unix	37
Редактирование в системе Unix	37
Компиляция в системе Unix	38
Операционная система Linux	39
Интегрированная среда разработки (Windows)	39
Компиляторы DOS для персональных компьютеров IBM PC	41
Работа с языком C в системах Macintosh	42
Языковые стандарты	42
Первый стандарт ANSI/ISO C	42
Стандарт C99	43
Как организована эта книга	44

Соглашения, принятые в этой книге	44
Шрифты и начертание	44
Выходные данные программы	45
Специальные клавиши	45
Системы, использованные при подготовке данной книги	46
Требования к системе	46
Специальные элементы	46
Резюме	47
Вопросы для самоконтроля	47
Упражнения по программированию	48
<b>Глава 2. Введение в язык C</b>	<b>49</b>
Простой пример программы на языке C	49
Пояснение к программе	50
Проход 1: краткий обзор	50
Проход 2: детали программы	52
Структура простой программы	62
Советы касательно удобства чтения программы	63
Еще один шаг в использовании языка C	64
Документирование	64
Множественные объявления	64
Умножение	64
Распечатка нескольких значений	65
Множество функций	65
Предварительные сведения об отладке	67
Синтаксические ошибки	67
Семантические ошибки	68
Состояние программы	70
Ключевые слова и зарезервированные идентификаторы	71
Ключевые понятия	71
Резюме	72
Вопросы для самоконтроля	73
Упражнения по программированию	74
<b>Глава 3. Представление данных в языке C</b>	<b>75</b>
Демонстрационная программа	75
Что нового в этой программе?	77
Переменные и константы	78
Ключевые слова, обозначающие типы	79
Целочисленные данные и данные с плавающей запятой	80
Целые числа	80
Числа с плавающей запятой	81
Базовые типы данных языка C	82
Тип данных int	82
Другие целочисленные типы	86

## 8 Содержание

Использование символов: тип <code>char</code>	92
Тип <code>_Bool</code>	99
Переносимые типы: <code>inttypes.h</code>	99
Данные типа <code>float</code> , <code>double</code> и <code>long double</code>	101
Комплексные и мнимые типы	106
За пределами базовых типов	106
Размеры типов	108
Использование типов данных	110
Аргументы и ошибки при их использовании	111
Еще один пример: управляющие последовательности	113
Каким будет результат выполнения этой программы	113
Сброс буфера выходных данных	114
Ключевые понятия	115
Резюме	115
Вопросы для самоконтроля	116
Упражнения по программированию	118
<b>Глава 4. Символьные строки и форматированный ввод-вывод</b>	<b>121</b>
Вводная программа	121
Строки символов: введение	123
Массив значений типа <code>char</code> и нулевой символ	123
Использование строк	124
Функция <code>strlen()</code>	125
Константы и препроцессор C	127
Модификатор <code>const</code>	131
Работа с символическими константами	131
Исследование и использование функций <code>printf()</code> и <code>scanf()</code>	133
Функция <code>printf()</code>	134
Использование функции <code>printf()</code>	135
Модификаторы спецификации преобразования для функции <code>printf()</code>	137
Что преобразует спецификация преобразования	143
Использование функции <code>scanf()</code>	150
Модификатор <code>*</code> и его использование в функциях <code>printf()</code> и <code>scanf()</code>	155
Советы по использованию функции <code>printf()</code>	157
Ключевые понятия	157
Резюме	158
Вопросы для самоконтроля	159
Упражнения по программированию	161
<b>Глава 5. Операции, выражения и операторы</b>	<b>163</b>
Введение в циклы	163
Фундаментальные операции	166
Операция присваивания: <code>=</code>	166
Операция сложения: <code>+</code>	168
Операция вычитания: <code>-</code>	168

Операции знака: - и +	168
Операция умножения: *	169
Операция деления: /	171
Приоритеты операций	172
Приоритеты и порядок вычисления	174
Некоторые дополнительные операции	176
Операция sizeof и тип size_t	176
Операция деления по модулю: %	177
Операции инкремента и декремента: ++ и --	178
Декремент: --	182
Приоритеты операций	183
Не будьте слишком самоуверенными	184
Выражения и операторы	185
Выражения	185
Операторы	186
Составные операторы (блоки)	189
Преобразования типов	191
Операция приведения	193
Функции с аргументами	195
Демонстрационная программа	197
Ключевые понятия	198
Резюме	199
Вопросы для самоконтроля	200
Упражнения по программированию	203
<b>Глава 6. Управляющие операторы: циклы</b>	<b>205</b>
Продолжение изучения цикла while	206
Комментарии по программе	207
Цикл считывания в стиле C	208
Оператор while	209
Завершение цикла while	210
Когда цикл завершается?	210
Оператор while: цикл с предусловием	211
Синтаксические особенности	211
Что больше: использование операций и выражений отношения	213
Что такое истина?	215
Какой еще может быть истина?	216
Трудности при употреблении понятия “истина”	217
Новый тип _Bool	219
Приоритеты операций отношения	220
Неопределенные циклы и циклы со счетчиком	222
Цикл for	223
Использование цикла for с целью повышения гибкости	225
Дополнительные операции присваивания: +=, -=, *=, /=, %=	230
Операция запятой	230
Греческий философ Зенон и цикл for	233

## 10 Содержание

Цикл с постусловием: do while	235
Какой цикл выбрать?	238
Вложенные циклы	239
Анализ программы	239
Вариации вложенных циклов	240
Введение в массивы	241
Использование цикла for при работе с массивами	242
Пример цикла, использующего возвращаемое значение функции	244
Анализ программы	247
Использование функций с возвращаемыми значениями	248
Ключевые понятия	249
Резюме	249
Вопросы для самоконтроля	250
Упражнения по программированию	254

## **Глава 7. Управляющие операторы: ветвление и безусловные переходы** **259**

Оператор if	260
Добавление конструкции else к оператору if	262
Еще один пример: знакомство с функциями getchar() и putchar()	264
Семейство символьных функций ctype.h	266
Множественный выбор else if	268
Объединение else и if в пары	271
Большее число вложений операторов if	273
Давайте будем логичными	277
Альтернативное представление: заголовочный файл isob46.h	278
Приоритеты операций	279
Порядок вычисления выражений	279
Диапазон значений	281
Программа подсчета слов	282
Условная операция: ?:	285
Дополнительные средства организации цикла: continue и break	288
Оператор continue	288
Оператор break	291
Множественный выбор: операторы switch и break	293
Использование оператора switch	295
Считывание только первого символа строки	296
Множественные метки	297
Операторы switch и if else	299
Оператор goto	300
Избегайте использования оператора goto	300
Ключевые понятия	304
Резюме	304
Вопросы для самоконтроля	305
Упражнения по программированию	308

<b>Глава 8. Символьный ввод-вывод и верификация ввода</b>	<b>311</b>
Односимвольные функции ввода-вывода: <code>getchar()</code> и <code>putchar()</code>	312
Буферы	313
Завершение ввода с клавиатуры	315
Файлы, потоки и ввод данных с клавиатуры	315
Конец файла	316
Перенаправление и файлы	320
Перенаправление в Unix, Linux и DOS	320
Создание дружественного пользовательского интерфейса	325
Работа с буферизованным вводом	325
Смешивание числового и символьного ввода	327
Проверка допустимости ввода	330
Анализ программы	335
Поток ввода и числа	336
Просмотр меню	336
Задачи	337
Обеспечение устойчивого выполнения программ	337
Функция <code>get_choice()</code>	338
Смешивание символьного и числового ввода	339
Ключевые понятия	342
Резюме	343
Вопросы для самоконтроля	343
Упражнения по программированию	344
<b>Глава 9. Функции</b>	<b>347</b>
Обзор функций	347
Создание и использование простой функции	349
Анализ программы	349
Аргументы функции	352
Определение функции с аргументами: формальные параметры	354
Создание прототипа функции с аргументами	355
Вызов функции с аргументом: фактические аргументы	355
Представление функции в виде черного ящика	356
Возврат значения функцией с помощью оператора <code>return</code>	357
Типы функций	360
Прототипирование функций в стандарте ANSI C	361
Решение проблемы	361
Решение стандарта ANSI	363
Отсутствие аргументов и неопределенные аргументы	364
Да здравствуют прототипы	365
Рекурсия	365
Рекурсия в действии	366
Основы рекурсии	367
Хвостовая рекурсия	369

## 12 Содержание

Рекурсия и обратный порядок	371
Преимущества и недостатки рекурсии	373
Компиляция программ из двух или большего числа исходных файлов	374
Unix	374
Linux	375
Компиляторы командной строки DOS	375
Компиляторы Windows и Macintosh	375
Использование заголовочных файлов	375
Поиск адресов: операция &	379
Изменение переменных в вызывающей функции	381
Указатели: первое знакомство	383
Операция разыменования: *	383
Объявление указателей	384
Использование указателей для обмена данными между функциями	385
Ключевые понятия	389
Резюме	390
Вопросы для самоконтроля	390
Упражнения по программированию	391
<b>Глава 10. Массивы и указатели</b>	<b>393</b>
Массивы	393
Инициализация	394
Выделенные инициализаторы (стандарт C99)	398
Присваивание значений массивам	399
Границы массива	400
Указание размера массива	402
Многомерные массивы	403
Инициализация двумерного массива	406
Массивы с размерностями больше двух	407
Указатели и массивы	407
Функции, массивы и указатели	411
Использование параметров типа указатель	413
Комментарии: указатели и массивы	416
Операции с указателями	416
Защита содержимого массива	421
Использование const с формальными параметрами	422
Дополнительные сведения о ключевом слове const	424
Указатели и многомерные массивы	426
Указатели на многомерные массивы	429
Совместимость указателей	430
Функции и многомерные массивы	431
Массивы переменной длины	435
Составные литералы	439
Ключевые понятия	441
Резюме	442

Вопросы для самоконтроля	443
Упражнения по программированию	445
<b>Глава 11. Символьные строки и строковые функции</b>	<b>449</b>
Введение в строки и строковый ввод-вывод	449
Определение строк в программе	451
Указатели и строки	459
Ввод строк	460
Выделение пространства памяти под строки	460
Функция gets()	460
Функция fgets()	463
Функция scanf()	464
Вывод строк	465
Функция puts()	466
Функция fputs()	467
Функция printf()	468
Возможность создания собственных функций	468
Строковые функции	471
Функция strlen()	471
Функция strcat()	473
Функция strncat()	474
Функция strcmp()	475
Возвращаемое значение функции strcmp()	476
Варианты функции strncmp()	479
Функции strcmpu() и strncmpu()	480
Остальные свойства функции strcmpu()	482
Тщательный выбор: функция strncmpu()	482
Функция sprintf()	484
Другие строковые функции	485
Пример обработки строк: сортировка строк	488
Сортировка указателей вместо строк	489
Выбор алгоритма сортировки	489
Символьные функции ctype.h и строки	491
Аргументы командной строки	493
Аргументы командной строки в интегрированных средах	495
Аргументы командной строки в среде Macintosh	495
Преобразование строк в числа	496
Ключевые понятия	499
Резюме	499
Вопросы для самоконтроля	500
Упражнения по программированию	503
<b>Глава 12. Классы памяти, компоновка и управление памятью</b>	<b>505</b>
Классы памяти	505
Область видимости	506



## 14 Содержание

Связывание	508
Продолжительность хранения	508
Автоматические переменные	510
Регистровые переменные	514
Статические переменные с областью видимости в пределах блока	514
Статические переменные с внешним связыванием	516
Статическая переменная с внешним связыванием	521
Множественные файлы	521
Спецификаторы классов памяти	522
Классы памяти и функции	525
Какой класс памяти следует выбрать?	526
Функция генерации случайных чисел и статическая переменная	526
Игра в кости	530
Распределение памяти: функции malloc() и free()	534
Важность функции free()	538
Функция calloc()	539
Распределение динамической памяти и массивы переменной длины	539
Классы памяти и динамическое распределение памяти	541
Квалификаторы типов в стандарте ANSI C	541
Квалификатор типа const	542
Квалификатор типа volatile	544
Квалификатор типа restrict	545
Новые места для старых ключевых слов	547
Ключевые понятия	547
Резюме	548
Вопросы для самоконтроля	549
Упражнения по программированию	551
<b>Глава 13. Файловый ввод-вывод</b>	<b>555</b>
Обмен данными с файлами	555
Что такое файл?	556
Текстовое и двоичное представление файлов	556
Уровни ввода-вывода	557
Стандартные файлы	558
Стандартный ввод-вывод	558
Проверка наличия аргументов командной строки	559
Функция fopen()	560
Функции getc() и putc()	561
Признак конца файла	562
Функция fclose()	563
Указатели на стандартные файлы	564
Простая программа сжатия файлов	564
Функции ввода-вывода: fprintf(), fscanf(), fgets() и fputs()	566
Функции fprintf() и fscanf()	566
Функции fgets() и fputs()	568

Комментарий: функции gets() и fgets()	569
Произвольный доступ: функции fseek() и ftell()	570
Как работают функции fseek() и ftell()	571
Сравнение двоичного и текстового режимов	573
Переносимость	573
Функции fgetpos() и fsetpos()	574
За кулисами стандартного ввода-вывода	575
Другие стандартные функции ввода-вывода	576
Функция int ungetc(int c, FILE *fp)	576
Функция int fflush()	576
Функция int setvbuf()	577
Двоичный ввод-вывод: fread() и fwrite()	577
Функция size_t fwrite()	579
Функция size_t fread()	580
Функции int feof(FILE *fp) и int ferror(FILE *fp)	580
Пример использования функций fread() и fwrite()	580
Произвольный доступ с двоичным вводом-выводом	583
Ключевые понятия	585
Резюме	586
Вопросы для самоконтроля	587
Упражнения по программированию	588
<b>Глава 14. Структуры и другие формы данных</b>	<b>593</b>
Учебная задача: создание каталога книг	593
Объявление структуры	595
Объявление переменной типа структуры	596
Инициализация структур	597
Доступ к элементам структуры	598
Выделенные инициализаторы структур	599
Массивы структур	599
Объявление массива структур	602
Идентификация элементов массива структур	602
Анализ программы	603
Вложенные структуры	604
Указатели на структуры	605
Объявление и инициализация указателя на структуру	607
Доступ к элементам структуры через указатели	607
Взаимодействие функций и структур	608
Передача элементов структуры	608
Использование адреса структуры	609
Передача структуры в качестве аргумента	610
Дальнейший анализ свойств структур	611
Структуры или указатели на структуры?	615
Символьные массивы или указатели на символы в структурах	616
Структура, указатели и функция malloc()	617

## 16 Содержание

Составные литералы и структуры (C99)	619
Элементы типа гибких массивов (C99)	621
Функции, использующие массив структур	623
Сохранение содержимого структур в файле	625
Пример сохранения структуры	626
Анализ программы	628
Структуры: что дальше?	629
Объединения: краткое знакомство	630
Перечислимые типы	633
Константы типа enum	634
Значения по умолчанию	634
Присваиваемые значения	634
Использование ключевого слова enum	635
Совместно используемые пространства имен	636
Оператор typedef: краткое знакомство	637
Фиктивные объявления	639
Функции и указатели	641
Ключевые понятия	648
Резюме	649
Вопросы для самоконтроля	650
Упражнения по программированию	653
<b>Глава 15. Операции с разрядами</b>	<b>657</b>
Двоичные числа, биты и байты	657
Двоичная запись целочисленных значений	658
Целочисленные значения со знаком	659
Двоичное представление чисел с плавающей точкой	660
Другие основания систем счисления	660
Восьмеричная система счисления	661
Шестнадцатеричная система счисления	661
Поразрядные операции	662
Поразрядные логические операции	663
Область применения: маски	664
Область применения: включение разрядов	666
Область применения: отключение разрядов	666
Область применения: переключение разрядов	666
Область применения: проверка значения разряда	667
Поразрядные операции сдвига	667
Пример программы	669
Еще один пример программы	671
Разрядные поля	673
Пример использования разрядных полей	674
Разрядные поля и поразрядные операции	678
Ключевые понятия	685
Резюме	685

Вопросы для самоконтроля	686
Упражнения по программированию	688
<b>Глава 16. Препроцессор и библиотека языка C</b>	<b>691</b>
Первые шаги трансляции программы	691
Именованные константы: #define	692
Лексемы	696
Переопределение констант	697
Использование аргументов в директиве #define	697
Создание строк из аргументов макроса: операция #	700
Средство объединения препроцессора: операция ##	701
Варьируемые макросы: ... и __VA_ARGS__	702
Макрос или функция?	703
Включение файлов: директива #include	704
Пример заголовочного файла	705
Область применения заголовочных файлов	707
Остальные директивы	708
Директива #undef	709
Определенность с точки зрения препроцессора	709
Условная компиляция	710
Предопределенные макросы	715
Директивы #line и #error	716
Директива #pragma	716
Встраиваемые функции	717
Библиотека C	720
Доступ к библиотеке C	720
Использование описаний библиотеки	721
Библиотека математических функций	722
Библиотека утилит общего назначения	725
Функции exit() и atexit()	725
Функция qsort()	727
Библиотека assert	732
Функции memcpy() и memmove() из библиотеки string.h	734
Переменные аргументы: файл stdarg.h	736
Ключевые понятия	738
Резюме	738
Вопросы для самоконтроля	739
Упражнения по программированию	740
<b>Глава 17. Расширенное представление данных</b>	<b>743</b>
Исследование темы представления данных	744
От массива к связанному списку	747
Использование связанного списка	750
Дополнительные соображения	754
Абстрактные типы данных	755

## 18 Содержание

Получение абстракции	756
Построение интерфейса	757
Использование интерфейса	762
Реализация интерфейса	764
Создание очереди с помощью ADT	771
Определение абстрактного типа данных очереди	771
Определение интерфейса	772
Реализация представления данных интерфейса	773
Тестирование очереди	781
Имитация реальной очереди	783
Сравнение связного списка и массива	789
Деревья бинарного поиска	793
Тип ADT бинарного дерева	795
Интерфейс дерева бинарного поиска	795
Реализация бинарного дерева	798
Тестирование дерева	813
Соображения по поводу дерева	817
Другие направления	819
Ключевые понятия	820
Резюме	820
Вопросы для самоконтроля	820
Упражнения по программированию	821
<b>Приложение А. Ответы на вопросы для самоконтроля</b>	<b>823</b>
Ответы на вопросы для самоконтроля из главы 1	823
Ответы на вопросы для самоконтроля из главы 2	823
Ответы на вопросы для самоконтроля из главы 3	825
Ответы на вопросы для самоконтроля из главы 4	828
Ответы на вопросы для самоконтроля из главы 5	830
Ответы на вопросы для самоконтроля из главы 6	833
Ответы на вопросы для самоконтроля из главы 7	836
Ответы на вопросы для самоконтроля из главы 8	840
Ответы на вопросы для самоконтроля из главы 9	841
Ответы на вопросы для самоконтроля из главы 10	843
Ответы на вопросы для самоконтроля из главы 11	845
Ответы на вопросы для самоконтроля из главы 12	847
Ответы на вопросы для самоконтроля из главы 13	848
Ответы на вопросы для самоконтроля из главы 14	851
Ответы на вопросы для самоконтроля из главы 15	855
Ответы на вопросы для самоконтроля из главы 16	856
Ответы на вопросы для самоконтроля из главы 17	858
<b>Приложение Б. Справочный раздел</b>	<b>861</b>
Раздел I. Дополнительные источники информации	861
Журнал	861

Сетевые ресурсы	861
Книги по языку С	862
Книги по программированию	863
Справочные руководства	863
Книги по С++	864
Раздел II. Операции С	864
Арифметические операции	865
Операции отношений	865
Операции присваивания	866
Логические операции	866
Условная операция	867
Операции, связанные с указателями	867
Операции знаков	868
Операции структур и объединений	868
Поразрядные операции	869
Прочие операции	870
Раздел III. Базовые типы и классы памяти	870
Обзор: базовые типы данных	870
Обзор: объявление простой переменной	872
Обзор: квалификаторы	874
Раздел IV. Выражения, операторы и поток управления программы	875
Обзор: выражения и операторы	875
Обзор: оператор while	876
Обзор: оператор for	876
Обзор: оператор do while	877
Обзор: использование операторов if для реализации выбора	877
Обзор: множественный выбор с помощью switch	878
Обзор: переходы в программе	880
Раздел V. Стандартная библиотека ANSI C с дополнениями C99	881
Диагностика: assert.h	881
Комплексные числа: complex.h (C99)	881
Обработка символов: ctype.h	883
Сообщения об ошибках: errno.h	884
Среда плавающей запятой: fenv.h (C99)	885
Преобразование формата целочисленных типов: inttypes.h (C99)	887
Локализация: locale.h	888
Математическая библиотека: math.h	891
Нелокальные переходы: setjmp.h	896
Обработка сигналов: signal.h	897
Переменное количество аргументов: stdarg.h	898
Поддержка булевских значений: stdbool.h (C99)	899
Общие определения: stddef.h	899
Целочисленные типы: stdint.h	900
Стандартная библиотека ввода-вывода: stdio.h	903
Общие утилиты: stdlib.h	906

## 20 Содержание

Обработка строк: string.h	912
Математические функции для общих типов: tgmth.h (C99)	915
Дата и время: time.h	916
Утилиты для работы с многобайтными и расширенными символами: wchar.h (C99)	920
Утилиты классификации и отображения расширенных символов: wctype.h (C99)	926
Раздел VI. Расширенные целочисленные типы	928
Типы строгой ширины	929
Типы минимальной ширины	929
Быстрые типы минимальной ширины	930
Типы максимальной ширины	930
Целые, которые могут хранить указатели	931
Расширенные целочисленные константы	931
Раздел VII. Расширенная поддержка символов	931
Триграфы	932
Диграфы	932
Альтернативная орфография: isob46.h	933
Многобайтные символы	933
Универсальные имена символов (UCN)	934
Расширенные символы	934
Расширенные и многобайтные символы	936
Раздел VIII. Расширенные средства вычислений C99	936
Стандарт плавающей запятой IEC	936
Заголовочный файл fenv.h	937
Указание компилятору STDC FP_CONTRACT	938
Дополнения к библиотеке math.h	938
Поддержка комплексных чисел	939
Раздел IX. Различия между C и C++	940
Прототипы функций	940
Константы char	941
Модификатор const	942
Структуры и объединения	943
Перечисления	943
Указатель на void	944
Булевские типы	944
Альтернативная орфография	944
Поддержка расширенных символов	944
Комплексные типы	945
Встраиваемые функции	945
Средства C99, которых нет в C++	945
<b>Приложение В. Набор символов ASCII</b>	<b>946</b>
<b>Предметный указатель</b>	<b>950</b>