

# Содержание

Благодарности	20
Об авторе	22
От издательства	22
Введение	23
Принятый подход	23
Примеры кода, используемые в книге	24
Организация книги	24
Примечание для преподавателей	28
Соглашения, используемые в этой книге	29
Системы, на которых разрабатывались примеры для книги	30
<b>Глава 1. Начало работы с C++</b>	<b>31</b>
Изучение языка C++: с чем придется иметь дело	32
Истоки языка C++: немного истории	33
Язык программирования C	33
Философия программирования на языке C	34
Переход к C++: объектно-ориентированное программирование	35
C++ и обобщенное программирование	37
Происхождение языка программирования C++	37
Переносимость и стандарты	38
Развитие языка	41
Эта книга и стандарты C++	41
Порядок создания программы	41
Создание файла исходного кода	42
Компиляция и компоновка	43
Компиляция и связывание в Unix	44
Компиляция и связывание в Linux	45
Компиляторы командной строки для режима командной строки Windows	45
Компиляторы для Windows	46
C++ в Macintosh	48
Резюме	48
<b>Глава 2. Приступаем к изучению C++</b>	<b>49</b>
Первые шаги в C++	50
Возможности функции <code>main()</code>	51
Заголовок функции как интерфейс	52
Почему функции <code>main()</code> нельзя назначать другое имя	54
Комментарии в языке C++	54
Препроцессор C++ и файл <code>iostream</code>	55
Имена заголовочных файлов	56
Пространства имен	57
Вывод в C++ с помощью <code>cout</code>	58
Манипулятор <code>endl</code>	60
Символ новой строки	60
Форматирование исходного кода C++	61
Лексемы и пробельные символы в исходном коде	61
Стиль написания исходного кода C++	62

Операторы в языке C++	63
Операторы объявления и переменные	63
Операторы присваивания	65
Новый трюк с объектом cout	66
Другие операторы C++	67
Использование cin	67
Конкатенация с помощью cout	68
cin и cout: признак класса	68
Функции	70
Использование функции, имеющей возвращаемое значение	70
Разновидности функций	74
Функции, определяемые пользователем	75
Использование определяемых пользователем функций, имеющих возвращаемое значение	78
Местоположение директивы using в программах с множеством функций	80
Резюме	81
Вопросы для самоконтроля	82
Упражнения по программированию	83
<b>Глава 3. Работа с данными</b>	<b>85</b>
Простые переменные	86
Имена, назначаемые переменным	87
Целочисленные типы	88
Целочисленные типы short, int, long и long long	89
Типы без знаков	94
Выбор целочисленного типа	96
Целочисленные литералы	97
Определение компилятором C++ типа константы	99
Тип char: символы и короткие целые числа	100
Тип bool	109
Квалификатор const	109
Числа с плавающей точкой	111
Запись чисел с плавающей точкой	111
Типы чисел с плавающей точкой	112
Константы с плавающей точкой	114
Преимущества и недостатки чисел с плавающей точкой	115
Арифметические операции в C++	116
Порядок выполнения операций: приоритеты операций и ассоциативность	117
Операция нахождения остатка от деления	119
Преобразования типов	120
Объявления auto в C++11	126
Резюме	127
Вопросы для самоконтроля	128
Упражнения по программированию	129
<b>Глава 4. Составные типы</b>	<b>131</b>
Введение в массивы	132
Замечания по программе	134
Правила инициализации массивов	135

## 8 Содержание

Инициализация массивов в C++11	136
Строки	136
Конкатенация строковых литералов	138
Использование строк в массивах	138
Риски, связанные с вводом строк	140
Построчное чтение ввода	141
Смешивание строкового и числового ввода	145
Введение в класс <code>string</code>	146
Инициализация строк в C++11	148
Другие формы строковых литералов	153
Введение в структуры	154
Использование структур в программах	155
Инициализация структур в C++11	158
Может ли структура содержать член типа <code>string</code> ?	158
Прочие свойства структур	158
Массивы структур	160
Битовые поля в структурах	161
Объединения	162
Перечисления	163
Установка значений перечислителей	165
Диапазоны значений перечислителей	165
Указатели и свободное хранилище	166
Объявление и инициализация указателей	169
Опасность, связанная с указателями	171
Указатели и числа	172
Выделение памяти с помощью операции <code>new</code>	172
Освобождение памяти с помощью операции <code>delete</code>	175
Использование операции <code>new</code> для создания динамических массивов	176
Указатели, массивы и арифметика указателей	179
Замечания по программе	180
Указатели и строки	184
Использование операции <code>new</code> для создания динамических структур	189
Автоматическое, статическое и динамическое хранилище	192
Комбинации типов	194
Альтернативы массивам	196
Шаблонный класс <code>vector</code>	196
Шаблонный класс <code>array</code> (C++11)	197
Сравнение массивов, объектов <code>vector</code> и объектов <code>array</code>	198
Резюме	199
Вопросы для самоконтроля	201
Упражнения по программированию	202
<b>Глава 5. Циклы и выражения отношений</b>	<b>205</b>
Введение в циклы <code>for</code>	206
Части цикла <code>for</code>	207
Возврат к циклу <code>for</code>	213
Изменение шага цикла	214
Доступ внутрь строк с помощью цикла <code>for</code>	215
Операции инкремента и декремента	216

Побочные эффекты и точки следования	217
Сравнение префиксной и постфиксной форм	218
Операции инкремента и декремента и указатели	219
Комбинация операций присваивания	220
Составные операторы, или блоки	221
Дополнительные синтаксические трюки — операция запятой	222
Выражения отношений	225
Присваивание, сравнение и вероятные ошибки	226
Сравнение строк в стиле C	228
Сравнение строк класса <code>string</code>	230
Цикл <code>while</code>	231
Замечания по программе	233
Сравнение циклов <code>for</code> и <code>while</code>	234
Построение цикла задержки	235
Цикл <code>do while</code>	237
Цикл <code>for</code> , основанный на диапазоне (C++11)	239
Циклы и текстовый ввод	240
Применение для ввода простого <code>cin</code>	240
Спасение в виде <code>cin.get(char)</code>	242
Выбор используемой версии <code>cin.get()</code>	243
Условие конца файла	243
Еще одна версия <code>cin.get()</code>	246
Вложенные циклы и двумерные массивы	250
Инициализация двумерного массива	251
Использование двумерного массива	252
Резюме	253
Вопросы для самоконтроля	254
Упражнения по программированию	255
<b>Глава 6. Операторы ветвления и логические операции</b>	257
Оператор <code>if</code>	258
Оператор <code>if else</code>	259
Форматирование операторов <code>if else</code>	261
Конструкция <code>if else if else</code>	262
Логические выражения	264
Логическая операция “ИЛИ”: <code>  </code>	264
Логическая операция “И”: <code>&amp;&amp;</code>	265
Логическая операция “НЕ”: <code>!</code>	270
Факты, связанные с логическими операциями	271
Альтернативные представления	272
Библиотека символьных функций <code>cctype</code>	272
Операция <code>?:</code>	275
Оператор <code>switch</code>	276
Использование перечислителей в качестве меток	279
Операторы <code>switch</code> и <code>if else</code>	280
Операторы <code>break</code> и <code>continue</code>	281
Замечания по программе	282
Циклы для чтения чисел	283
Замечания по программе	286

## 10 Содержание

Простой файловый ввод-вывод	287
Текстовый ввод-вывод и текстовые файлы	287
Запись в текстовый файл	288
Чтение текстового файла	292
Резюме	297
Вопросы для самоконтроля	298
Упражнения по программированию	300
<b>Глава 7. Функции как программные модули C++</b>	<b>303</b>
Обзор функций	304
Определение функции	305
Прототипирование и вызов функции	307
Аргументы функций и передача по значению	310
Множественные аргументы	312
Еще одна функция с двумя аргументами	314
Функции и массивы	316
Как указатели позволяют функциям обрабатывать массивы	318
Последствия использования массивов в качестве аргументов	318
Дополнительные примеры функций для работы с массивами	321
Функции, работающие с диапазонами массивов	327
Указатели и <code>const</code>	329
Функции и двумерные массивы	332
Функции и строки в стиле C	333
Функции с аргументами – строками в стиле C	334
Функции, возвращающие строки в стиле C	335
Функции и структуры	337
Передача и возврат структур	337
Еще один пример использования функций со структурами	339
Передача адресов структур	344
Функции и объекты класса <code>string</code>	346
Функции и объекты <code>array</code>	347
Замечания по программе	349
Рекурсия	349
Рекурсия с одиночным рекурсивным вызовом	349
Рекурсия с множественными рекурсивными вызовами	351
Указатели на функции	352
Основы указателей на функции	353
Пример с указателем на функцию	355
Вариации на тему указателей на функции	356
Упрощение объявлений с помощью <code>typedef</code>	360
Резюме	361
Вопросы для самоконтроля	362
Упражнения по программированию	363
<b>Глава 8. Дополнительные сведения о функциях</b>	<b>367</b>
Встроенные функции C++	368
Ссылочные переменные	371
Создание ссылочных переменных	371
Ссылки как параметры функций	374

Свойства и особенности ссылок	377
Временные переменные, ссылочные аргументы и квалификатор <code>const</code>	379
Использование ссылок при работе со структурами	381
Использование ссылок на объект класса	388
Еще один урок ООП: объекты, наследование и ссылки	391
Когда целесообразно использовать ссылочные аргументы	394
Аргументы по умолчанию	395
Перегрузка функций	398
Пример перегрузки	400
Когда целесообразно использовать перегрузку функций	403
Шаблоны функций	404
Перегруженные шаблоны	407
Ограничения шаблонов	408
Явные специализации	409
Создание экземпляров и специализация	412
Какую версию функции выбирает компилятор?	414
Эволюция шаблонных функций	421
Резюме	424
Вопросы для самоконтроля	425
Упражнения по программированию	426
<b>Глава 9. Модели памяти и пространства имен</b>	<b>429</b>
Раздельная компиляция	430
Продолжительность хранения, область видимости и компоновка	435
Область видимости и связывание	436
Автоматическая продолжительность хранения	437
Переменные со статической продолжительностью хранения	442
Спецификаторы и классификаторы	452
Функции и связывание	455
Языковое связывание	456
Схемы хранения и динамическое выделение памяти	457
Инициализация с помощью операции <code>new</code>	458
Когда <code>new</code> дает сбой	458
<code>new</code> : операции, функции и заменяющие функции	458
Операция <code>new</code> с размещением	459
Пространства имен	463
Традиционные пространства имен C++	463
Новое средство пространств имен	465
Пример пространства имен	472
Пространства имен и будущее	475
Резюме	476
Вопросы для самоконтроля	477
Упражнения по программированию	479
<b>Глава 10. Объекты и классы</b>	<b>483</b>
Процедурное и объектно-ориентированное программирование	484
Абстракции и классы	485
Что такое тип?	486
Классы в C++	486

## 12 Содержание

Реализация функций-членов класса	492
Использование классов	496
Изменение реализации	498
Обзор ситуации на текущий момент	499
Конструкторы и деструкторы классов	500
Усовершенствование класса <code>Stock</code>	505
Обзор конструкторов и деструкторов	513
Изучение объектов: указатель <code>this</code>	514
Массив объектов	520
Область видимости класса	522
Абстрактные типы данных	526
Резюме	530
Вопросы для самоконтроля	531
Упражнения по программированию	531
<b>Глава 11. Работа с классами</b>	<b>535</b>
Перегрузка операций	537
Время в наших руках: разработка примера перегрузки операции	538
Добавление операции сложения	540
Ограничения перегрузки	543
Дополнительные перегруженные операции	545
Что такое друзья?	547
Создание друзей	549
Общий вид друга: перегрузка операции <code>&lt;&lt;</code>	550
Перегруженные операции: сравнение функций-членов и функций, не являющихся членами	556
Дополнительные сведения о перегрузке: класс <code>Vector</code>	557
Использование члена, хранящего состояние	564
Перегрузка арифметических операций для класса <code>Vector</code>	566
Комментарии к реализации	568
Использование класса <code>Vector</code> при решении задачи случайного блуждания	569
Автоматические преобразования и приведения типов в классах	572
Преобразования и друзья	583
Резюме	585
Вопросы для самоконтроля	587
Упражнения по программированию	587
<b>Глава 12. Классы и динамическое выделение памяти</b>	<b>591</b>
Динамическая память и классы	592
Простой пример и статические члены класса	592
Специальные функции-члены	601
Новый усовершенствованный класс <code>String</code>	609
О чем следует помнить при использовании операции <code>new</code> в конструкторах	619
Замечания о возвращаемых объектах	622
Использование указателей на объекты	625
Обзор технических приемов	634
Моделирование очереди	635
Класс <code>Queue</code>	636
Класс <code>Customer</code>	646

Моделирование работы банкомата	649
Резюме	653
Вопросы для самоконтроля	655
Упражнения по программированию	656
<b>Глава 13. Наследование классов</b>	659
Начало: простой базовый класс	661
Порождение класса	663
Конструкторы: варианты доступа	664
Использование производного класса	667
Особые отношения между производным и базовым классами	669
Наследование: отношение <i>является</i>	671
Полиморфное открытое наследование	673
Разработка классов Brass и BrassPlus	674
Статическое и динамическое связывание	685
Совместимость типов указателей и ссылок	685
Виртуальные функции-члены и динамическое связывание	687
Что следует знать о виртуальных методах	690
Управление доступом: protected	693
Абстрактные базовые классы	694
Применение концепции абстрактных базовых классов	697
Философия АБК	702
Наследование и динамическое выделение памяти	703
Случай 1: производный класс не использует операцию new	703
Случай 2: производный класс использует операцию new	704
Пример наследования с динамическим выделением памяти и дружественными функциями	706
Обзор структуры класса	711
Функции-члены, генерируемые компилятором	711
Другие соображения относительно методов класса	713
Соображения по поводу открытого наследования	716
Сводка функций классов	720
Резюме	721
Вопросы для самоконтроля	721
Упражнения по программированию	723
<b>Глава 14. Повторное использование кода в C++</b>	727
Классы с объектами-членами	728
Класс valarray: краткий обзор	729
Проект класса Student	730
Пример класса Student	731
Закрытое наследование	738
Новый вариант класса Student	738
Включение или закрытое наследование?	745
Защищенное наследование	746
Переопределение доступа с помощью using	746
Множественное наследование	748
Краткий обзор множественного наследования	764
Шаблоны классов	765

## 14 Содержание

Определение шаблона класса	766
Более внимательный взгляд на шаблонные классы	771
Пример шаблона массива и нетипизированные аргументы	776
Универсальность шаблонов	778
Специализации шаблона	781
Шаблоны-члены	784
Шаблонные классы и друзья	788
Резюме	795
Вопросы для самоконтроля	797
Упражнения по программированию	799

## Глава 15. Друзья, исключения и многое другое 805

Друзья	806
Дружественные классы	806
Дружественные функции-члены	811
Другие дружественные отношения	813
Вложенные классы	815
Вложенные классы и доступ	817
Вложение в шаблонах	818
Исключения	821
Вызов <code>abort()</code>	822
Возврат кода ошибки	823
Механизм исключений	825
Использование объектов в качестве исключений	827
Спецификации исключений в C++11	831
Раскручивание стека	832
Дополнительные свойства исключений	837
Класс <code>exception</code>	839
Исключения, классы и наследование	843
Потеря исключений	847
Предостережения относительно использования исключений	850
Динамическая идентификация типов	852
Для чего нужен механизм RTTI	852
Как работает механизм RTTI	852
Операция <code>dynamic_cast</code>	853
Операции приведения типов	860
Резюме	863
Вопросы для самоконтроля	864
Упражнения по программированию	865

## Глава 16. Класс `string` и стандартная библиотека шаблонов 867

Класс <code>string</code>	868
Создание объекта <code>string</code>	868
Ввод для класса <code>string</code>	872
Работа со строками	875
Другие возможности, предлагаемые классом <code>string</code>	880
Разновидности строк	881
Классы шаблонов интеллектуальных указателей	882
Использование интеллектуальных указателей	883

Соображения по поводу интеллектуальных указателей	886
Выбор интеллектуального указателя	890
Стандартная библиотека шаблонов (STL)	892
Класс шаблона <code>vector</code>	892
Что еще можно делать с помощью векторов	894
Дополнительные возможности векторов	899
Цикл <code>for</code> , основанный на диапазоне (C++11)	903
Обобщенное программирование	903
Предназначение итераторов	904
Виды итераторов	908
Иерархия итераторов	911
Концепции, уточнения и модели	912
Виды контейнеров	918
Ассоциативные контейнеры	929
Неупорядоченные ассоциативные контейнеры (C++11)	935
Функциональные объекты (функторы)	936
Концепции функторов	937
Предопределенные функторы	939
Адаптируемые функторы и функциональные адаптеры	941
Алгоритмы	943
Группы алгоритмов	944
Основные свойства алгоритмов	944
Библиотека STL и класс <code>string</code>	946
Сравнение функций и методов контейнеров	947
Использование STL	949
Шаблон <code>initializer_list</code> (C++11)	957
Замечания по программе	960
Резюме	960
Вопросы для самоконтроля	962
Упражнения по программированию	963
<b>Глава 17. Ввод, вывод и файлы</b>	967
Обзор ввода и вывода в C++	968
Потоки и буферы	969
Потоки, буферы и файл <code>iostream</code>	971
Перенаправление	973
Вывод с помощью <code>cout</code>	975
Перегруженная операция <code>&lt;&lt;</code>	975
Другие методы <code>ostream</code>	977
Очистка выходного буфера	980
Форматирование с помощью <code>cout</code>	981
Ввод с помощью <code>cin</code>	996
Восприятие ввода операцией <code>cin &gt;&gt;</code>	998
Состояния потока	1000
Другие методы класса <code>istream</code>	1004
Другие методы класса <code>istream</code>	1011
Файловый ввод и вывод	1015
Простой файловый ввод-вывод	1016
Проверка потока и <code>is_open()</code>	1019

## 16 Содержание

Открытие нескольких файлов	1020
Обработка командной строки	1020
Режимы файла	1022
Произвольный доступ	1032
Внутреннее форматирование	1040
Резюме	1042
Вопросы для самопроверки	1044
Упражнения по программированию	1045
<b>Глава 18. Новый стандарт C++</b>	<b>1049</b>
Обзор уже известных функциональных средств C++11	1050
Новые типы	1050
Унифицированная инициализация	1050
Объявления	1052
<code>nullptr</code>	1054
Интеллектуальные указатели	1054
Изменения в спецификации исключений	1054
Перечисления с областью видимости	1054
Изменения в классах	1055
Изменения в шаблонах и STL	1056
Ссылка <code>rvalue</code>	1058
Семантика переноса и ссылка <code>rvalue</code>	1059
Необходимость в семантике переноса	1059
Пример семантики переноса	1061
Исследование конструктора переноса	1065
Присваивание	1067
Принудительное применение переноса	1067
Новые возможности классов	1071
Специальные функции-члены	1071
Явно заданные по умолчанию и удаленные методы	1072
Делегирование конструкторов	1074
Наследование конструкторов	1074
Управление виртуальными методами: <code>override</code> и <code>final</code>	1076
Лямбда-функции	1077
Как работают указатели на функции, функторы и лямбда	1077
Более подробно о лямбда-функциях	1081
Оболочки	1083
Оболочка <code>function</code> и неэффективность шаблонов	1084
Решение проблемы	1086
Дополнительные возможности	1087
Шаблоны с переменным числом аргументов	1088
Пакеты параметров шаблонов и функций	1089
Распаковка пакетов	1090
Использование рекурсии в шаблонных функциях с переменным числом аргументов	1090
Другие средства C++11	1093
Параллельное программирование	1093
Библиотечные дополнения	1094
Низкоуровневое программирование	1094

Смешанные средства	1095
Языковые изменения	1096
Проект Boost	1096
Проект TR1	1097
Использование Boost	1097
Что дальше?	1098
Резюме	1099
Вопросы для самоконтроля	1100
Упражнения по программированию	1102
<b>Приложение А. Основания систем счисления</b>	1105
Десятичные числа (основание 10)	1105
Восьмеричные целые числа (основание 8)	1106
Шестнадцатеричные числа (основание 16)	1106
Двоичные числа (основание 2)	1107
Двоичная и шестнадцатеричная формы записи	1107
<b>Приложение Б. Резервированные слова C++</b>	1109
Ключевые слова C++	1109
Альтернативные лексемы	1110
Резервированные имена библиотеки C++	1110
Идентификаторы со специальным назначением	1111
<b>Приложение В. Набор символов ASCII</b>	1113
<b>Приложение Г. Приоритеты операций</b>	1119
<b>Приложение Д. Другие операции</b>	1123
Битовые операции	1123
Операции сдвига	1123
Логические битовые операции	1125
Альтернативные представления битовых операций	1127
Примеры использования битовых операций	1128
Операции разыменования членов	1129
alignof (C++11)	1133
noexcept (C++11)	1134
<b>Приложение Е. Шаблонный класс string</b>	1135
Тринадцать типов и константа	1136
Информация о данных, конструкторы и вспомогательные элементы	1137
Конструктор по умолчанию	1139
Конструкторы, использующие строки в стиле C	1140
Конструкторы, использующие часть строки в стиле C	1140
Конструкторы, использующие ссылку lvalue	1140
Конструкторы, использующие ссылку rvalue (C++11)	1142
Конструктор, использующий n копий символа	1142
Конструктор, использующий диапазон	1142
Конструктор, использующий список инициализаторов (C++11)	1143
Различные действия с памятью	1143

## 18 Содержание

Доступ к строке	1143
Базовое присваивание	1145
Поиск в строках	1145
Семейство <code>find()</code>	1145
Семейство <code>rfind()</code>	1146
Семейство <code>find_first_of()</code>	1147
Семейство <code>find_last_of()</code>	1147
Семейство <code>find_first_not_of()</code>	1147
Семейство <code>find_last_not_of()</code>	1148
Методы и функции сравнения	1148
Модификация строк	1149
Методы присоединения и добавления	1150
Дополнительные методы присваивания	1150
Методы вставки	1151
Методы очистки	1152
Методы замены	1152
Другие методы модификации: <code>copy()</code> и <code>swap()</code>	1153
Ввод и вывод	1153
<b>Приложение Ж. Методы и функции стандартной библиотеки шаблонов</b>	<b>1155</b>
Библиотека STL и C++11	1155
Новые контейнеры	1155
Изменения в контейнерах C++98	1156
Члены, общие для всех или большинства контейнеров	1157
Дополнительные члены для контейнеров последовательностей	1160
Дополнительные операции для множеств и карт	1164
Неупорядоченные ассоциативные контейнеры (C++11)	1166
Функции библиотеки STL	1168
Операции, не модифицирующие последовательности	1169
Операции, видоизменяющие последовательности	1174
Операции сортировки и связанные с ними операции	1184
Числовые операции	1197
<b>Приложение З. Рекомендуемая литература и ресурсы в Интернете</b>	<b>1201</b>
Рекомендуемая литература	1201
Ресурсы в Интернете	1203
<b>Приложение И. Переход к стандарту ANSI/ISO C++</b>	<b>1205</b>
Используйте альтернативы для некоторых директив препроцессора	1205
Используйте <code>const</code> вместо <code>#define</code> для определения констант	1205
Используйте <code>inline</code> вместо <code>#define</code> для определения коротких функций	1207
Используйте прототипы функций	1208
Используйте приведения типов	1208
Знакомьтесь с функциональными возможностями C++	1209
Используйте новую организацию заголовочных файлов	1209
Используйте пространства имен	1209
Используйте интеллектуальные указатели	1211
Используйте класс <code>string</code>	1211
Используйте библиотеку STL	1211

<b>Приложение К. Ответы на вопросы для самоконтроля</b>	1213
Глава 2	1213
Глава 3	1214
Глава 4	1215
Глава 5	1217
Глава 6	1218
Глава 7	1219
Глава 8	1221
Глава 9	1223
Глава 10	1225
Глава 11	1227
Глава 12	1228
Глава 13	1230
Глава 14	1232
Глава 15	1233
Глава 16	1234
Глава 17	1236
Глава 18	1237
<b>Предметный указатель</b>	1240