

Содержание

Введение	19
ГЛАВА 1. Первые шаги	23
1.1. Создание простой программы на языке C++	24
1.1.1. Компиляция и запуск программы	25
1.2. Первый взгляд на ввод-вывод	27
1.2.1. Стандартные объекты ввода и вывода	28
1.2.2. Программа, использующая библиотеку ввода-вывода	28
1.3. Несколько слов о комментариях	32
1.4. Средства управления	34
1.4.1. Оператор while	34
1.4.2. Оператор for	36
1.4.3. Оператор if	39
1.4.4. Ввод неизвестного количества данных	41
1.5. Введение в классы	42
1.5.1. Класс Sales_item	43
1.5.2. Первый взгляд на функции-члены	46
1.6. Программа на языке C++	48
Резюме	49
Термины	50
ЧАСТЬ I. Основы	53
ГЛАВА 2. Переменные и базовые типы	55
2.1. Простые встроенные типы	56
2.1.1. Целочисленные типы	57
2.1.2. Типы с плавающей запятой	60
2.2. Литеральные константы	61
2.3. Переменные	65
2.3.1. Что такое переменная?	67
2.3.2. Имя переменной	68
2.3.3. Определение объектов	70
2.3.4. Правила инициализации переменных	73
2.3.5. Объявления и определения	75
2.3.6. Область видимости имен	76
2.3.7. Определение переменных по месту применения	78
2.4. Спецификатор const	78
2.5. Ссылки	81
2.6. Определение имен типов	83
2.7. Перечисления	84

2.8. Типы классов	85
2.9. Создание собственных файлов заголовка	89
2.9.1. Разработка собственных заголовков	91
2.9.2. Кратко о препроцессоре	93
Резюме	95
Термины	96
ГЛАВА 3. Библиотечные типы данных	101
3.1. Пространства имен и объявления using	102
3.2. Библиотечный тип string	104
3.2.1. Определение и инициализация строк	104
3.2.2. Чтение и запись строк	105
3.2.3. Операции со строками	107
3.2.4. Работа с символами строки	112
3.3. Библиотечный тип vector	114
3.3.1. Определение и инициализация векторов	115
3.3.2. Операции с векторами	117
3.4. Знакомство с итераторами	120
3.4.1. Арифметические действия с итераторами	124
3.5. Библиотечный тип bitset	125
3.5.1. Определение и инициализация наборов битов	125
3.5.2. Операции с наборами битов	127
Резюме	130
Термины	130
ГЛАВА 4. Массивы и указатели	133
4.1. Массивы	134
4.1.1. Определение и инициализация массивов	134
4.1.2. Операции с массивами	137
4.2. Знакомство с указателями	138
4.2.1. Что такое указатель?	138
4.2.2. Определение и инициализация указателей	139
4.2.3. Операции с указателями	144
4.2.4. Использование указателей для доступа к элементам массива	146
4.2.5. Указатели и спецификатор const	151
4.3. Символьная строка в стиле C	154
4.3.1. Динамическое создание массивов	159
4.3.2. Взаимодействие со старым кодом	164
4.4. Многомерные массивы	165
4.4.1. Указатели и многомерные массивы	167
Резюме	168
Термины	168
ГЛАВА 5. Выражения	171
5.1. Арифметические операторы	173
5.2. Операторы отношения и логические операторы	176

5.3. Побитовые операторы	179
5.3.1. Использование битовых наборов и целочисленных значений	181
5.3.2. Использование операторов сдвига для организации ввода и вывода	183
5.4. Оператор присвоения	184
5.4.1. Оператор присвоения имеет порядок выполнения справа налево	184
5.4.2. Оператор присвоения имеет низкий приоритет	185
5.4.3. Составные операторы присвоения	186
5.5. Операторы инкремента и декремента	187
5.6. Оператор стрелка (->)	189
5.7. Условный оператор	190
5.8. Функция sizeof()	191
5.9. Оператор запятая (,)	192
5.10. Вычисление составных выражений	193
5.10.1. Приоритет	193
5.10.2. Порядок	194
5.10.3. Порядок вычисления операндов	197
5.11. Операторы new и delete	199
5.12. Преобразование типов	204
5.12.1. Когда происходит неявное преобразование типов	205
5.12.2. Арифметические преобразования	205
5.12.3. Другие неявные преобразования	207
5.12.4. Явное преобразование	209
5.12.5. Когда может пригодиться приведение типов	209
5.12.6. Именованные операторы приведения	210
5.12.7. Приведение типов в старом стиле	212
Резюме	213
Термины	214
ГЛАВА 6. Операторы	217
6.1. Простые операторы	218
6.2. Операторы объявления	219
6.3. Составные операторы (блоки)	219
6.4. Операторная область видимости	220
6.5. Оператор if	221
6.5.1. Оператор if с разделом else	223
6.6. Оператор switch	225
6.6.1. Использование оператора switch	226
6.6.2. Порядок выполнения внутри оператора switch	227
6.6.3. Метка default	228
6.6.4. Выражение switch и метки case	229
6.6.5. Определение переменной внутри оператора switch	229
6.7. Оператор while	231
6.8. Оператор цикла for	233
6.8.1. Цикл for без частей заголовка	234
6.8.2. Несколько определений в заголовке цикла for	235
6.9. Оператор цикла do...while	236

6.10. Оператор break	237
6.11. Оператор continue	239
6.12. Оператор goto	239
6.13. Блок try и обработка исключений	241
6.13.1. Оператор throw	242
6.13.2. Блок try	242
6.13.3. Стандартные исключения	245
6.14. Использование препроцессора для отладки	246
Резюме	248
Термины	249
ГЛАВА 7. Функции	251
7.1. Определение функций	251
7.1.1. Тип возвращаемого значения функции	253
7.1.2. Список параметров функции	254
7.2. Передача аргумента	256
7.2.1. Нессылочные параметры	256
7.2.2. Ссылочные параметры	258
7.2.3. Параметры типа векторов и других контейнеров	264
7.2.4. Параметры в виде массива	264
7.2.5. Манипулирование массивами, переданными в функции	267
7.2.6. Функция main(): обработка параметров командной строки	269
7.2.7. Функции с варьирующимися параметрами	270
7.3. Оператор return	271
7.3.1. Функции без возвращаемого значения	271
7.3.2. Функции, возвращающие значение	272
7.3.3. Рекурсия	276
7.4. Объявление функций	277
7.4.1. Значения параметров по умолчанию	278
7.5. Локальные объекты	280
7.5.1. Автоматические объекты	280
7.5.2. Статические локальные объекты	281
7.6. Встраиваемые функции	282
7.7. Функции-члены класса	284
7.7.1. Определение тела функции-члена	284
7.7.2. Определение функции-члена вне класса	287
7.7.3. Создание конструктора Sales_item()	288
7.7.4. Организация файлов кода классов	290
7.8. Перегруженные функции	291
7.8.1. Перегрузка и область видимости	293
7.8.2. Подбор функций и преобразование аргументов	295
7.8.3. Три этапа подбора перегруженной версии	296
7.8.4. Преобразование типов аргументов	299
7.9. Указатели на функции	302
Резюме	305
Термины	306

ГЛАВА 8. Библиотека ввода-вывода	309
8.1. Объектно-ориентированная библиотека	310
8.2. Значения состояния потока	314
8.3. Управление буфером вывода	317
8.4. Ввод и вывод в файл	319
8.4.1. Использование объектов файловых потоков	320
8.4.2. Режимы файла	323
8.4.3. Программа, открывающая и проверяющая файл	325
8.5. Строковые потоки	326
Резюме	329
Термины	329
ЧАСТЬ II. Контейнеры и алгоритмы	331
ГЛАВА 9. Последовательные контейнеры	333
9.1. Определение последовательного контейнера	334
9.1.1. Инициализация элементов контейнера	335
9.1.2. Ограничения типов элементов, которые может содержать контейнер	337
9.2. Итераторы и диапазоны итераторов	339
9.2.1. Диапазоны итераторов	341
9.2.2. Некоторые операции с контейнерами делают итераторы некорректными	343
9.3. Операции с последовательными контейнерами	343
9.3.1. Вспомогательные типы, определенные в классе контейнера	344
9.3.2. Функции-члены <code>begin()</code> и <code>end()</code>	345
9.3.3. Добавление элементов в последовательный контейнер	345
9.3.4. Операторы сравнения	349
9.3.5. Операции с размером контейнера	351
9.3.6. Доступ к элементам	352
9.3.7. Удаление элементов	353
9.3.8. Присвоение и функция <code>swap()</code>	356
9.4. Как увеличивается размер вектора	358
9.4.1. Функции-члены <code>capacity()</code> и <code>reserve()</code>	359
9.5. Как выбрать тип контейнера	361
9.6. Еще раз о строках	363
9.6.1. Дополнительные способы создания строк	365
9.6.2. Дополнительные способы изменения строк	367
9.6.3. Операции, специфические только для строк	369
9.6.4. Операции поиска строк	371
9.6.5. Сравнение строк	374
9.7. Адаптеры контейнеров	376
9.7.1. Адаптер <code>stack</code>	377
9.7.2. Очередь и приоритетная очередь	379
Резюме	380
Термины	380

ГЛАВА 10. Ассоциативные контейнеры	383
10.1. Предварительные сведения: тип <code>pair</code>	384
10.2. Ассоциативные контейнеры	386
10.3. Тип <code>map</code>	387
10.3.1. Определение карты	387
10.3.2. Типы, определенные в шаблоне <code>map</code>	389
10.3.3. Добавление элементов карты	390
10.3.4. Индексация карты	390
10.3.5. Применение функции <code>map::insert()</code>	392
10.3.6. Поиск и возвращение элементов карты	394
10.3.7. Удаление элементов карты	396
10.3.8. Перебор элементов карты	396
10.3.9. Карта преобразования слов	397
10.4. Тип <code>set</code>	399
10.4.1. Определение и применение наборов	400
10.4.2. Создание набора, исключающего слова	401
10.5. Типы <code>multimap</code> и <code>multiset</code>	403
10.5.1. Добавление и удаление элементов	403
10.5.2. Поиск элементов в контейнерах <code>multimap</code> и <code>multiset</code>	404
10.6. Применение контейнеров: программа <code>TextQuery</code>	407
10.6.1. Проект программы	408
10.6.2. Класс <code>TextQuery</code>	409
10.6.3. Применение класса <code>TextQuery</code>	411
10.6.4. Создание функций-членов	413
Резюме	415
Термины	416
ГЛАВА 11. Общие алгоритмы	417
11.1. Краткий обзор	418
11.2. Первый взгляд на алгоритмы	421
11.2.1. Алгоритмы, только читающие элементы контейнера	421
11.2.2. Алгоритмы, записывающие элементы контейнера	424
11.2.3. Алгоритмы, переупорядочивающие элементы контейнера	427
11.3. Возвращаясь к итераторам	432
11.3.1. Итераторы вставки	432
11.3.2. Итераторы ввода-вывода	434
11.3.3. Реверсивные итераторы	439
11.3.4. Константные итераторы	442
11.3.5. Пять категорий итераторов	443
11.4. Структура общих алгоритмов	446
11.4.1. Параметрическая схема алгоритмов	446
11.4.2. Соглашения об именовании алгоритмов	448
11.5. Алгоритмы, специфические для контейнеров	449
Резюме	451
Термины	452

ЧАСТЬ III. Абстракция, классы и данные	455
ГЛАВА 12. Классы	457
12.1. Определение и объявление классов	458
12.1.1. Определение класса	458
12.1.2. Абстракция данных и инкапсуляция	460
12.1.3. Подробнее об определении классов	463
12.1.4. Объявление и определение класса	466
12.1.5. Объекты класса	467
12.2. Неявный указатель this	468
12.3. Область видимости класса	473
12.3.1. Поиск имен в области видимости класса	475
12.4. Конструкторы	480
12.4.1. Список инициализирующих значений конструктора	482
12.4.2. Аргументы по умолчанию и конструкторы	486
12.4.3. Стандартный конструктор	487
12.4.4. Неявное преобразование	490
12.4.5. Явная инициализация переменных-членов класса	492
12.5. Дружественные отношения	493
12.6. Статические члены класса	496
12.6.1. Статические функции-члены	498
12.6.2. Статические переменные-члены	498
Резюме	501
Термины	501
ГЛАВА 13. Управление копированием	505
13.1. Конструктор копий	506
13.1.1. Синтезируемый конструктор копий	509
13.1.2. Определение собственного конструктора копий	510
13.1.3. Предотвращение копирования	511
13.2. Оператор присвоения	512
13.3. Деструктор	514
13.4. Пример обработки сообщения	517
13.5. Работа с указателями	522
13.5.1. Определение классов интеллектуальных указателей	525
13.5.2. Определение классов подобных значению	530
Резюме	532
Термины	533
ГЛАВА 14. Перегрузка операторов и преобразования	535
14.1. Определение перегруженного оператора	536
14.1.1. Проект перегруженного оператора	540
14.2. Операторы ввода и вывода	543
14.2.1. Перегрузка оператора вывода <<	543
14.2.2. Перегрузка оператора ввода >>	545
14.3. Арифметические операторы и операторы отношения	548

14.3.1. Операторы равенства	549
14.3.2. Операторы отношения	550
14.4. Операторы присвоения	550
14.5. Оператор индексирования	551
14.6. Операторы доступа к членам класса	553
14.7. Операторы инкремента и декремента	556
14.8. Оператор вызова функции объекта	560
14.8.1. Использование объектов функции с библиотечными алгоритмами	561
14.8.2. Библиотечные объекты функций	563
14.8.3. Адаптеры функций для объектов функций	565
14.9. Преобразования и типы классов	566
14.9.1. Зачем нужны функции преобразования	566
14.9.2. Операторы преобразования	567
14.9.3. Соответствие аргументов и преобразования	571
14.9.4. Поиск перегруженной функции и аргументы класса	575
14.9.5. Перегрузка, преобразования и операторы	579
Резюме	582
Термины	583
ЧАСТЬ IV. Объектно-ориентированное и общее программирование	585
ГЛАВА 15. Объектно-ориентированное программирование	587
15.1. Краткий обзор ООР	588
15.2. Определение базовых и производных классов	590
15.2.1. Определение базового класса	590
15.2.2. Защищенные члены	592
15.2.3. Производные классы	594
15.2.4. Виртуальные и другие функции-члены	597
15.2.5. Открытое, закрытое и защищенное наследование	602
15.2.6. Наследование и дружественные отношения	606
15.2.7. Наследование и статические члены	606
15.3. Преобразования и наследование	607
15.3.1. Преобразование производного класса в базовый	608
15.3.2. Преобразование из базового класса в производный	610
15.4. Конструкторы и функции управления копированием	611
15.4.1. Конструкторы и функции управления копированием базового класса	611
15.4.2. Конструкторы производного класса	612
15.4.3. Управление копированием и наследование	616
15.4.4. Виртуальные деструкторы	618
15.4.5. Виртуальность конструкторов и деструкторов	620
15.5. Область видимости класса при наследовании	621
15.5.1. Поиск имен осуществляется во время компиляции	622
15.5.2. Конфликт имен и наследование	622

15.5.3. Область видимости и функции-члены	624
15.5.4. Виртуальные функции и область видимости	625
15.6. Чистые виртуальные функции	627
15.7. Контейнеры и наследование	628
15.8. Управляющие классы и наследование	629
15.8.1. Управляющий класс, подобный указателю	630
15.8.2. Клонирование неизвестного типа	633
15.8.3. Использование управляющего класса	634
15.9. Продолжение приложения TextQuery	639
15.9.1. Объектно-ориентированное решение	640
15.9.2. Управляющий класс, подобный значению	641
15.9.3. Класс Query_base	644
15.9.4. Управляющий класс Query	644
15.9.5. Производные классы	647
15.9.6. Виртуальные функции eval()	650
Резюме	652
Термины	653

ГЛАВА 16. Шаблоны и общее программирование 655

16.1. Определение шаблона	656
16.1.1. Определение шаблона функции	657
16.1.2. Определение шаблона класса	659
16.1.3. Параметры шаблона	660
16.1.4. Параметры типа шаблона	662
16.1.5. Параметры значения шаблона	664
16.1.6. Создание общих программ	665
16.2. Создание экземпляра	667
16.2.1. Дедукция аргумента шаблона	669
16.2.2. Явные аргументы шаблона функции	673
16.3. Модели компиляции шаблона	676
16.4. Члены шаблона класса	679
16.4.1. Функции-члены шаблона класса	682
16.4.2. Аргументы шаблона для параметров значения	686
16.4.3. Дружественные отношения в шаблонах класса	688
16.4.4. Объявление дружественных отношений между шаблонами Queue и QueueItem	690
16.4.5. Шаблоны-члены	692
16.4.6. Законченный класс Queue	696
16.4.7. Статические члены шаблонов класса	697
16.5. Общий управляющий класс	698
16.5.1. Определение управляющего класса	699
16.5.2. Применение управляющего класса	701
16.6. Специализация шаблона	703
16.6.1. Специализация шаблона функции	704
16.6.2. Специализация шаблона класса	706
16.6.3. Специализация членов, но не класса	709
16.6.4. Частичная специализация шаблона класса	710

16.7. Перегрузка и шаблоны функций	711
Резюме	714
Термины	715
ЧАСТЬ V. Дополнительные темы	717
ГЛАВА 17. Инструменты для крупномасштабных программ	719
17.1. Обработка исключений	720
17.1.1. Передача исключения типа класса	721
17.1.2. Прокрутка стека	723
17.1.3. Обработка исключения	724
17.1.4. Повторная передача исключения	727
17.1.5. Обработчик для всех исключений	728
17.1.6. Блок try функции и конструкторы	729
17.1.7. Иерархия класса исключения	729
17.1.8. Автоматическое освобождение ресурсов	731
17.1.9. Класс auto_ptr	733
17.1.10. Спецификация исключений	739
17.1.11. Спецификация исключений указателя на функцию	743
17.2. Пространства имен	744
17.2.1. Определение пространств имен	744
17.2.2. Вложенные пространства имен	749
17.2.3. Неименованные пространства имен	750
17.2.4. Использование членов пространства имен	752
17.2.5. Классы, пространства имен и области видимости	756
17.2.6. Перегрузка и пространства имен	759
17.2.7. Пространства имен и шаблоны	762
17.3. Множественное и виртуальное наследование	763
17.3.1. Множественное наследование	763
17.3.2. Преобразования и несколько базовых классов	766
17.3.3. Управление копированием при множественном наследовании	768
17.3.4. Область видимости класса при множественном наследовании	769
17.3.5. Виртуальное наследование	772
17.3.6. Объявление виртуального базового класса	774
17.3.7. Семантика специальной инициализации	776
Резюме	779
Термины	780
ГЛАВА 18. Специализированные инструменты и технологии	783
18.1. Оптимизация распределения памяти	783
18.1.1. Резервирование памяти в языке C++	784
18.1.2. Класс allocator	785
18.1.3. Функции operator new() и operator delete()	789
18.1.4. Размещающий оператор new	791
18.1.5. Явный вызов деструктора	793
18.1.6. Операторы new и delete, специфические для класса	794

18.1.7. Базовый класс системы резервирования памяти	796
18.2. Идентификация типов времени выполнения	803
18.2.1. Оператор <code>dynamic_cast</code>	803
18.2.2. Оператор <code>typeid</code>	806
18.2.3. Применение RTTI	808
18.2.4. Класс <code>type_info</code>	810
18.3. Указатель на член класса	811
18.3.1. Объявление указателя на член класса	811
18.3.2. Применение указателя на член класса	814
18.4. Вложенные классы	817
18.4.1. Реализация вложенного класса	818
18.4.2. Поиск имен в области видимости вложенного класса	821
18.5. Объединение: экономный класс	823
18.6. Локальные классы	826
18.7. Возможности, снижающие переносимость	828
18.7.1. Битовые поля	828
18.7.2. Спецификатор <code>volatile</code>	830
18.7.3. Директивы компоновки: <code>extern "C"</code>	832
Резюме	835
Термины	836
ПРИЛОЖЕНИЕ А. Библиотека	839
A.1. Имена и заголовки стандартной библиотеки	839
A.2. Краткий обзор алгоритмов	841
A.2.1. Алгоритмы поиска объекта	841
A.2.2. Другие алгоритмы, осуществляющие только чтение	843
A.2.3. Алгоритмы бинарного поиска	844
A.2.4. Алгоритмы записи в элементы контейнера	845
A.2.5. Алгоритмы сортировки и разделения	847
A.2.6. Общие функции изменения порядка	849
A.2.7. Алгоритмы перестановки	852
A.2.8. Алгоритмы набора для отсортированных последовательностей	853
A.2.9. Минимальные и максимальные значения	854
A.2.10. Числовые алгоритмы	854
A.3. Возвращаясь к библиотеке ввода-вывода	856
A.3.1. Флаги формата	856
A.3.2. Большинство манипуляторов изменяют флаг формата	857
A.3.3. Управление форматом вывода	858
A.3.4. Управление форматом ввода	865
A.3.5. Бесформатные операции ввода-вывода	866
A.3.6. Однобайтовые операторы	866
A.3.7. Многобайтовые операторы	868
A.3.8. Произвольный доступ к потоку	869
A.3.9. Чтение и запись в тот же файл	872
Предметный указатель	875