

Содержание

Предисловие ко второму изданию	19
Благодарности ко второму изданию	20
Предисловие к первому изданию	21
Благодарности к первому изданию	22
Ждем ваших отзывов!	24
Глава 1. О книге	25
1.1. Для чего предназначена эта книга	25
1.2. Что необходимо знать читателю	26
1.3. Стиль и структура книги	26
Содержание книги	27
1.4. Как читать книгу	29
1.5. Последние достижения	29
1.6. Примеры и дополнительная информация	29
1.7. Обратная связь	30
Глава 2. Введение в язык C++ и стандартную библиотеку	31
2.1. История стандартов языка C++	31
2.1.1. Обычные вопросы о стандарте C++11	32
2.1.2. Совместимость стандартов C++98 и C++11	33
2.2. Сложность и O-обозначения	34
Глава 3. Новые средства языка	37
3.1. Новые языковые средства стандарта C++11	37
3.1.1. Небольшие, но важные синтаксические уточнения	37
3.1.2. Автоматическое выведение типа с помощью ключевого слова <code>auto</code>	38
3.1.3. Универсальная инициализация и списки инициализации	39
3.1.4. Диапазонные циклы <code>for</code>	41
3.1.5. Семантика перемещения и <code>rvalue</code> -ссылки	43
3.1.6. Новые строковые литералы	48
3.1.7. Ключевое слово <code>noexcept</code>	49
3.1.8. Ключевое слово <code>constexpr</code>	51
3.1.9. Новые возможности шаблонов	52
3.1.10. Лямбда-выражения и лямбда-функции	53
3.1.12. Новый синтаксис объявления функций	58
3.1.13. Перечисления с ограниченной областью видимости	58
3.1.14. Новые фундаментальные типы данных	59

3.2. Старые “новые” средства языка	59
3.2.1. Неявная инициализация фундаментальных типов	63
3.2.2. Определение функции <code>main()</code>	64
Глава 4. Общие принципы	65
4.1. Пространство имен <code>std</code>	65
4.2. Заголовочные файлы	66
4.3. Обработка ошибок и исключений	67
4.3.1. Стандартные классы исключений	68
4.3.2. Члены классов исключений	71
4.3.3. Передача исключений с помощью класса <code>exception_ptr</code>	80
4.3.4. Генерирование стандартных исключений	80
4.3.5. Наследование классов стандартных исключений	81
4.4. Вызываемые объекты	82
4.5. Параллельное программирование и многопоточность	83
4.6. Распределители памяти	85
Глава 5. Вспомогательные средства	87
5.1. Пары и кортежи	87
5.1.1. Пары	88
5.1.2. Кортежи	96
5.1.3. Ввод-вывод кортежей	102
5.1.4. Преобразования типов <code>tuples</code> и <code>pairs</code>	104
5.2. Интеллектуальные указатели	104
5.2.1. Класс <code>shared_ptr</code>	105
5.2.2. Класс <code>weak_ptr</code>	113
5.2.3. Неправильное использование совместно используемых указателей	118
5.2.4. Подробное описание совместно используемых и слабых указателей	120
5.2.5. Класс <code>unique_ptr</code>	127
5.2.6. Подробное описание класса <code>unique_ptr</code>	139
5.2.7. Класс <code>auto_ptr</code>	142
5.2.8. Заключительные замечания об интеллектуальных указателях	144
5.3. Числовые пределы	145
5.4. Свойства и утилиты типов	152
5.4.1. Предназначение свойств типов	152
5.4.2. Подробное описание свойств типов	155
5.4.3. Обертки для ссылок	163
5.4.4. Обертки функциональных типов	163
5.5. Вспомогательные функции	164
5.5.1. Вычисление минимума и максимума	164
5.5.2. Обмен двух значений	167
5.5.3. Вспомогательные операторы сравнения	169
5.6. Арифметика рациональных чисел на этапе компиляции	170

5.7. Часы и таймеры	174
5.7.1. Обзор библиотеки Chrono	174
5.7.2. Интервалы времени	175
5.7.3. Часы и моменты времени	180
5.7.4. Функции для работы с датами и временем в языке C и стандарте POSIX	189
5.7.5. Блокировка с помощью таймеров	191
5.8. Заголовочные файлы <code><cstdlib></code> , <code><stdlib.h></code> и <code><cstring></code>	192
5.8.1. Определения в заголовочном файле <code><cstdlib></code>	192
5.8.2. Определения в заголовочном файле <code><stdlib.h></code>	193
5.8.3. Определения в заголовочном файле <code><cstring></code>	194

Глава 6. Стандартная библиотека шаблонов 195

6.1. Компоненты библиотеки STL	195
6.2. Контейнеры	197
6.2.1. Последовательные контейнеры	199
6.2.2. Ассоциативные контейнеры	207
6.2.3. Неупорядоченные контейнеры	211
6.2.4. Ассоциативные массивы	216
6.2.5. Другие контейнеры	218
6.2.6. Адаптеры контейнеров	219
6.3. Итераторы	219
6.3.1. Дополнительные примеры использования ассоциативных и неупорядоченных контейнеров	225
6.3.2. Категории итераторов	229
6.4. Алгоритмы	231
6.4.1. Диапазоны	234
6.4.2. Обработка нескольких диапазонов	239
6.5. Адаптеры итераторов	241
6.5.1. Итераторы вставки	241
6.5.2. Поточковые итераторы	244
6.5.3. Обратные итераторы	246
6.5.4. Итераторы перемещения	247
6.6. Пользовательские обобщенные функции	247
6.7. Модифицирующие алгоритмы	248
6.7.1. Удаление элементов	249
6.7.2. Работа с ассоциативными и неупорядоченными контейнерами	252
6.7.3. Алгоритмы и функции-члены	253
6.8. Функции в качестве аргументов алгоритма	254
6.8.1. Использование функций в качестве аргументов алгоритмов	255
6.8.2. Предикаты	257
6.9. Использование лямбда-выражений	259
6.10. Функциональные объекты	263
6.10.1. Определение функциональных объектов	263
6.10.2. Стандартные функциональные объекты	269

6.10.3. Связыватели	271
6.10.4. Функциональные объекты и связыватели против лямбда-функции	274
6.11. Элементы контейнеров	274
6.11.1. Требования к элементам контейнеров	274
6.11.2. Семантика значений и семантика ссылок	275
6.12. Ошибки и исключения в библиотеке STL	276
6.12.1. Обработка ошибок	276
6.12.2. Обработка исключений	278
6.13. Расширение библиотеки STL	281
6.13.1. Интеграция дополнительных типов	281
6.13.2. Наследование типов библиотеки STL	282

Глава 7. Контейнеры STL 283

7.1. Общие возможности и операции над контейнерами	283
7.1.1. Возможности контейнеров	283
7.1.2. Операции над контейнерами	284
7.1.3. Типы контейнеров	291
7.2. Массивы	291
7.2.1. Возможности массивов	292
7.2.2. Операции над массивами	294
7.2.3. Использование объектов <code>array<></code> как массивов в стиле языка C	298
7.2.4. Обработка исключений	299
7.2.5. Интерфейс кортежа	299
7.2.6. Примеры использования массивов	299
7.3. Векторы	300
7.3.1. Возможности векторов	301
7.3.2. Операции над векторами	303
7.3.3. Использование векторов в качестве массивов языка C	309
7.3.4. Обработка исключений	310
7.3.5. Примеры использования векторов	311
7.3.6. Класс <code>vector<bool></code>	313
7.4. Деки	314
7.4.1. Возможности деков	315
7.4.2. Операции над деком	316
7.4.3. Обработка исключений	320
7.4.4. Примеры использования деков	320
7.5. Списки	321
7.5.1. Возможности списков	322
7.5.2. Операции над списками	323
7.5.3. Обработка исключений	329
7.5.4. Примеры использования списков	330
7.6. Последовательные списки	332
7.6.1. Возможности последовательных списков	332
7.6.2. Операции над последовательными списками	334

7.6.3. Обработка исключений	345
7.6.4. Примеры использования последовательных списков	345
7.7. Множества и мультимножества	347
7.7.1. Возможности множеств и мультимножеств	348
7.7.2. Операции над множествами и мультимножествами	349
7.7.3. Обработка исключений	359
7.7.4. Примеры использования множеств и мультимножеств	359
7.7.5. Пример задания критерия сортировки во время выполнения программы	362
7.8. Отображения и мультиотображения	364
7.8.1. Возможности отображений и мультиотображений	365
7.8.2. Операции над отображениями и мультиотображениями	366
7.8.3. Использование отображений как ассоциативных массивов	377
7.8.4. Обработка исключений	379
7.8.5. Примеры использования отображений и мультиотображений	379
7.8.6. Пример с отображениями, строками и критериями сортировки, задаваемыми во время выполнения программы	384
7.9. Неупорядоченные контейнеры	387
7.9.1. Возможности неупорядоченных контейнеров	389
7.9.2. Создание неупорядоченных контейнеров и управление ими	393
7.9.3. Другие операции над неупорядоченными контейнерами	400
7.9.4. Интерфейс сегментов	407
7.9.5. Использование неупорядоченных отображений в качестве ассоциативных массивов	408
7.9.6. Обработка исключений	409
7.9.7. Примеры использования неупорядоченных контейнеров	409
7.10. Другие контейнеры STL	418
7.10.1. Строки как контейнеры STL	419
7.10.2. Обычные массивы в стиле языка C как контейнеры STL	419
7.11. Реализация семантики ссылок Использование разделяемых указателей	421
7.12. Когда и какой контейнер использовать	425
Глава 8. Детальное описание контейнеров STL	429
8.1. Определения типов	429
8.2. Операции создания, копирования и удаления	432
8.3. Немодифицирующие операции	435
8.3.1. Операции над размером	435
8.3.2. Операции сравнения	436
8.3.3. Немодифицирующие операции над ассоциативными и неупорядоченными контейнерами	437
8.4. Присваивание	439
8.5. Прямой доступ к элементам	441
8.6. Операции генерации итераторов	443

8.7. Вставка и удаление элементов	445
8.7.1. Вставка отдельных элементов	445
8.7.2. Вставка нескольких элементов	450
8.7.3. Удаление элементов	452
8.7.4. Изменение размера	455
8.8. Специальные функции-члены для списков и последовательных списков	455
8.8.1. Специальные функции-члены для списков (и последовательных списков)	455
8.8.2. Специальные функции-члены, предназначенные только для последовательных списков	459
8.9. Интерфейсы стратегий	463
8.9.1. Немодифицирующие вспомогательные функции	463
8.9.2. Модифицирующие вспомогательные функции	464
8.9.3. Сегментный интерфейс для неупорядоченных контейнеров	465
8.10. Функции для выделения памяти	466
8.10.1. Основные члены распределителя памяти	467
8.10.2. Конструкторы для необязательных параметров распределителя памяти	467
Глава 9. Итераторы STL	471
9.1. Заголовочные файлы для итераторов	471
9.2. Категории итераторов	471
9.2.1. Итераторы вывода	471
9.2.2. Итераторы ввода	473
9.2.3. Однонаправленные итераторы	474
9.2.4. Двухнаправленные итераторы	475
9.2.5. Итераторы произвольного доступа	475
9.2.6. Проблема инкремента и декремента итераторов вектора	478
9.3. Вспомогательные функции для работы с итераторами	479
9.3.1. Функция <code>advance()</code>	479
9.3.2. Функции <code>next()</code> и <code>prev()</code>	481
9.3.3. Функция <code>distance()</code>	483
9.3.4. Функция <code>iter_swap()</code>	484
9.4. Адаптеры итераторов	485
9.4.1. Обратные итераторы	486
9.4.2. Итераторы вставки	491
9.4.3. Итераторы потоков	497
9.4.4. Итераторы перемещения	502
9.5. Свойства итераторов	502
9.5.1. Запись обобщенных функций для итераторов	504
9.6. Создание пользовательских итераторов	506
Глава 10. Функциональные объекты STL и лямбда-функции	511
10.1. Концепция функциональных объектов	511
10.1.1. Функциональные объекты как критерий сортировки	512

10.1.2. Функциональные объекты, имеющие внутреннее состояние	513
10.1.3. Значение, возвращаемое алгоритмом <code>for_each()</code>	517
10.1.4. Предикаты и функциональные объекты	518
10.2. Стандартные функциональные объекты и привязки	521
10.2.1. Стандартные функциональные объекты	521
10.2.2. Функциональные адаптеры и привязки	522
10.2.3. Пользовательские функциональные объекты для функциональных адаптеров	530
10.2.4. Устаревшие функциональные адаптеры	532
10.3. Использование лямбда-функций	534
10.3.1. Лямбда-функции и адаптеры	534
10.3.2. Лямбда-функции и функциональные объекты, имеющие состояние	535
10.3.3. Лямбда-функции, вызывающие глобальные функции и функции-члены	537
10.3.4. Лямбда-функции как функции-хеширования, критерий сортировки и критерий эквивалентности	539

Глава 11. Алгоритмы STL 541

11.1. Заголовочные файлы для алгоритмов	541
11.2. Обзор алгоритмов	541
11.2.1. Краткое введение	542
11.2.2. Классификация алгоритмов	542
11.3. Вспомогательные функции	554
11.4. Алгоритм <code>for_each()</code>	556
11.5. Немодифицирующие алгоритмы	560
11.5.1. Подсчет элементов	560
11.5.2. Минимум и максимум	562
11.5.3. Поиск элементов	564
11.5.4. Сравнение диапазонов	576
11.5.5. Предикаты для диапазонов	583
11.6. Модифицирующие алгоритмы	589
11.6.1. Копирование элементов	590
11.6.2. Перемещение элементов	593
11.6.3. Преобразование и объединение элементов	595
11.6.5. Присвоение новых значений	600
11.6.6. Замена элементов	603
11.7. Алгоритмы удаления	606
11.7.1. Удаление определенных значений	606
11.7.2. Удаление дубликатов	609
11.8. Перестановочные алгоритмы	613
11.8.1. Перестановка элементов в обратном порядке	613
11.8.2. Циклическая перестановка элементов	614
11.8.3. Перестановка элементов	617
11.8.4. Перетасовка элементов	619

11.8.5. Перемещение элементов в начало	621
11.8.6. Разделение на два подынтервала	623
11.9. Алгоритмы сортировки	624
11.9.1. Сортировка всех элементов	624
11.9.2. Частичная сортировка	627
11.9.3. Сортировка по <i>n</i> -му элементу	630
11.9.4. Алгоритмы для работы с пирамидой	632
11.10. Алгоритмы для упорядоченных диапазонов	635
11.10.1. Поиск элементов	636
11.10.2. Слияние диапазонов	641
11.11. Численные алгоритмы	649
11.11.1. Вычисления	649
11.11.2. Преобразования относительных и абсолютных значений	653
Глава 12. Специальные контейнеры	657
Недавние изменения, связанные со стандартом C++11	657
12.1. Стеки	657
12.1.1. Основной интерфейс	658
12.1.2. Пример использования стеков	659
12.1.3. Пользовательский класс стека	660
12.1.4. Подробное описание класса <code>stack<></code>	663
12.2. Очереди	663
12.2.1. Основной интерфейс	665
12.2.2. Пример использования очереди	665
12.2.3. Пользовательский класс очереди	666
12.2.4. Подробное описание класса <code>queue <></code>	666
12.3. Очереди с приоритетами	666
12.3.1. Основной интерфейс	668
12.3.2. Пример использования очереди с приоритетами	668
12.3.3. Подробное описание класса <code>priority_queue<></code>	669
12.4. Подробное описание контейнерных адаптеров	670
12.4.1. Определения типов	670
12.4.2. Конструкторы	670
12.4.3. Вспомогательные конструкторы для очередей с приоритетами	671
12.4.4. Операции	672
12.5. Битовые множества	674
12.5.1. Примеры использования битовых множеств	675
12.5.2. Подробное описание класса <code>bitset</code>	677
Глава 13. Строки	679
Изменения, внесенные в стандарте C++11	679
13.1. Предназначение строковых классов	680
13.1.1. Первый пример: извлечение имени временного файла	680
13.1.2. Второй пример: извлечение слов и вывод их в обратном порядке	684

13.2. Описание строковых классов	687
13.2.1. Строковые типы	687
13.2.2. Обзор операций	690
13.2.3. Конструкторы и деструкторы	692
13.2.4. Строки и С-строки	693
13.2.5. Размер и емкость	694
13.2.6. Доступ к элементам	695
13.2.7. Сравнения	697
13.2.8. Модифицирующие операции	698
13.2.9. Конкатенация подстрок и строк	701
13.2.10. Операторы ввода-вывода	701
13.2.11. Поиск	703
13.2.12. Значение <code>npos</code>	705
13.2.13. Числовые преобразования	706
13.2.14. Поддержка итераторов для строк	708
13.2.15. Интернационализация	713
13.2.16. Производительность	715
13.2.17. Строки и векторы	716
13.3. Подробное описание класса <code>string</code>	716
13.3.1. Определения типов и статические значения	716
13.3.2. Операции создания, копирования и уничтожения	718
13.3.3. Операции над размерами и емкостью	719
13.3.4. Сравнения	720
13.3.5. Доступ к символам	722
13.3.6. Создание С-строк и массивов символов	723
13.3.7. Модифицирующие операции	724
13.3.8. Поиск	732
13.3.9. Подстроки и конкатенация строк	735
13.3.10. Функции ввода-вывода	736
13.3.11. Числовые преобразования	737
13.3.12. Генерация итераторов	738
13.3.13. Поддержка механизмов распределения памяти	739
Глава 14. Регулярные выражения	741
14.1. Интерфейс сравнения и поиска регулярных выражений	741
14.2. Работа с подвыражениями	744
14.3. Итераторы регулярных выражений	750
14.4. Итераторы токенов регулярных выражений	751
14.5. Замена регулярных выражений	754
14.6. Флаги регулярных выражений	755
14.7. Исключения, связанные с регулярными выражениями	759
14.8. Грамматика ECMAScript	761
14.9. Другие грамматики	763
14.10. Подробное описание основных сигнатур регулярных выражений	764

Глава 15. Классы потоков ввода-вывода	767
15.1. Основы потоков ввода-вывода	768
15.1.1. Потокные объекты	768
15.1.2. Классы потоков	768
15.1.3. Глобальные потокные объекты	769
15.1.4. Потокные операторы	769
15.1.5. Манипуляторы	769
15.1.6. Простой пример	770
15.2. Основные потокные классы и объекты	771
15.2.1. Иерархия классов	771
15.2.2. Глобальные потокные объекты	775
15.2.3. Заголовочные файлы	775
15.3. Стандартные потокные операторы << и >>	776
15.3.1. Оператор вывода <<	776
15.3.2. Оператор ввода >>	778
15.3.3. Ввод и вывод специальных типов	778
15.4. Состояние потоков	781
15.4.1. Константы состояния потоков	781
15.4.2. Функции-члены для доступа к состоянию потоков	783
15.4.3. Состояние потока и булевы условия	784
15.4.4. Состояние потока и исключения	786
15.5. Стандартные функции ввода-вывода	791
15.5.1. Функции-члены для ввода	791
15.5.2. Функции-члены для вывода	795
15.5.3. Примеры использования	796
15.5.4. Объекты класса <code>sentry</code>	797
15.6. Манипуляторы	797
15.6.1. Обзор манипуляторов	798
15.6.2. Как работают манипуляторы	800
15.6.3. Пользовательские манипуляторы	801
15.7. Форматирование	802
15.7.1. Флаги форматирования	802
15.7.2. Формат ввода-вывода булевых значений	804
15.7.3. Ширина поля, символ-заполнитель и выравнивание	805
15.7.4. Положительный знак и верхний регистр	807
15.7.5. Основание системы счисления	808
15.7.6. Вывод чисел с плавающей точкой	810
15.7.7. Общие определения формата	812
15.8. Интернационализация	813
15.9. Доступ к файлам	814
15.9.1. Классы файловых потоков	814
15.9.2. <code>Rvalue</code> и семантика перемещения для файловых потоков	818
15.9.3. Флаги файлов	819
15.9.4. Произвольный доступ	822
15.9.5. Использование дескрипторов файлов	824

15.10. Поточковые классы для работы со строками	825
15.10.1. Строковые потоковые классы	825
15.10.2. Семантика перемещения для строковых потоков	829
15.10.3. Поточковые классы <code>char*</code>	830
15.11. Операции ввода-вывода для пользовательских типов	832
15.11.1. Реализация операций вывода	832
15.11.2. Реализация операций ввода	835
15.11.3. Ввод и вывод с помощью вспомогательных функций	837
15.11.4. Пользовательские флаги форматов	838
15.11.5. Соглашения создания пользовательских операций ввода-вывода	840
15.12. Связывание потоков ввода и вывода	841
15.12.1. Нежесткое связывание с помощью функции <code>tie()</code>	841
15.12.2. Жесткое связывание с помощью потоковых буферов	842
15.12.3. Перенаправление стандартных потоков	844
15.12.4. Потоки для чтения и записи	846
15.13. Классы потоковых буферов	847
15.13.1. Интерфейсы потоковых буферов	848
15.13.2. Итераторы потоковых буферов	850
15.13.3. Пользовательские потоковые буфера	853
15.14. Проблемы эффективности	865
15.14.1. Синхронизация со стандартными потоками языка C	865
15.14.2. Буферизация в потоковых буферах	866
15.14.3. Непосредственное использование потоковых буферов	867

Глава 16. Интернационализация 869

16.1. Кодирование и наборы символов	870
16.1.1. Многобайтовый текст и текст из широких символов	870
16.1.2. Разные кодировки символов	871
16.1.3. Работа с кодировками в языке C++	872
16.1.4. Свойства символов	873
16.1.5. Интернационализация специальных символов	877
16.2. Концепция локального контекста	878
16.2.1. Использование локальных контекстов	880
16.2.2. Фацеты	885
16.3. Подробное описание объекта локального контекста	888
16.4. Подробное описание фацетов	890
16.4.1. Форматирование чисел	891
16.4.2. Форматирование денежных величин	896
16.4.3. Форматирование времени и даты	905
16.4.4. Классификация и преобразование символов	912
16.4.5. Сравнение строк	926
16.4.6. Интернационализация сообщений	927

Глава 17. Работа с числами	929
17.1. Случайные числа и распределения	929
17.1.1. Первый пример	930
17.1.2. Генераторы	934
17.1.3. Подробное описание генераторов	937
17.1.4. Распределения	939
17.1.5. Подробное описание распределений	943
17.2. Комплексные числа	947
17.2.1. Общее описание класса <code>complex<></code>	947
17.2.2. Примеры использования класса <code>complex<></code>	948
17.2.3. Операции над комплексными числами	950
17.2.4. Подробное описание класса <code>complex<></code>	957
17.3. Глобальные числовые функции	962
17.4. Массивы значений	964
Глава 18. Параллельное программирование	967
18.1. Высокоуровневый интерфейс: <code>async()</code> и <code>future<></code>	968
18.1.1. Первый пример использования функции <code>async()</code> и класса <code>future<></code>	968
18.1.2. Пример ожидания двух задач	978
18.1.3. Разделяемые фьючерсы	983
18.2. Низкоуровневый интерфейс: потоки и обещания	986
18.2.1. Класс <code>std::thread</code>	986
18.2.2. Обещания	992
18.2.3. Класс <code>packaged_task<></code>	994
18.3. Подробное описание потоков	995
18.3.1. Подробное описание функции <code>async()</code>	996
18.3.2. Подробное описание фьючерсов	998
18.3.3. Подробное описание разделяемых фьючерсов	999
18.3.4. Подробное описание класса <code>std::promise</code>	1000
18.3.5. Подробное описание класса <code>std::packaged_task</code>	1001
18.3.6. Подробное описание класса <code>std::thread</code>	1003
18.3.7. Пространство имен <code>this_thread</code>	1004
18.4. Синхронизация потоков, или проблема конкурентности	1005
18.4.1. Осторожно, конкурентность!	1005
18.4.2. Причина проблем при состязании за данные	1006
18.4.3. Что именно создает опасность (расширение проблемы)	1007
18.4.4. Способы решения проблем	1010
18.5. Мьютексы и блокировки	1012
18.5.1. Использование мьютексов и блокировок	1012
18.5.2. Подробное описание мьютексов и блокировок	1022
18.5.3. Одновременный вызов нескольких потоков	1025
18.6. Условные переменные	1027
18.6.1. Предназначение условных переменных	1027

18.6.2. Первый законченный пример использования условных переменных	1028
18.6.3. Использование условных переменных для реализации очереди для нескольких потоков	1030
18.6.4. Подробное описание условных переменных	1033
18.7. Атомарные операции	1035
18.7.1. Пример использования атомарных операций	1036
18.7.2. Подробное описание атомарных типов и их низкоуровневого интерфейса	1040
18.7.3. Интерфейс атомарных типов в стиле языка C	1042
18.7.4. Низкоуровневый интерфейс атомарных типов	1044
Глава 19. Распределители памяти	1047
19.1. Использование распределителей памяти с точки зрения прикладного программиста	1047
19.2. Пользовательский распределитель памяти	1048
19.3. Использование распределителей памяти с точки зрения разработчика библиотеки	1050
Приложение	1055
S.1. Битовые множества	1055
S.1.1. Примеры использования битовых множеств	1056
S.1.2. Подробное описание класса <code>bitset<></code>	1058
S.2. Массивы значений	1065
S.2.1. Описание массивов значений	1066
S.2.2. Подмножества массивов значений	1072
S.2.3. Подробное описание класса <code>valarray</code>	1086
S.2.4. Подробное описание классов подмножеств массивов значений	1093
S.3. Подробное описание распределителей памяти и функций для работы с памятью	1097
S.3.1. Распределители памяти с ограниченной областью видимости	1098
S.3.2. Пользовательские распределители памяти в стандарте C++	1099
S.3.3. Распределитель памяти по умолчанию	1102
S.3.4. Подробное описание распределителей	1104
S.3.5. Подробное описание утилит для работы с неинициализированной памятью	1108
Библиография	1111
Новостные группы и форумы	1111
Книги и веб-сайты	1111
Предметный указатель	1117