

Содержание

Предисловие	25
Обращение к студентам	28
Обращение к преподавателям	29
Стандарт ISO	30
Поддержка	31
Благодарности	31
Глава 0. Обращение к читателям	33
0.1. Структура книги	34
0.1.1. Общие принципы	35
0.1.2. Упражнения, задачи и т.п.	37
0.1.3. Что дальше	38
0.2. Принципы обучения и изучения	39
0.2.1. Порядок изложения	42
0.2.2. Программирование и языки программирования	44
0.2.3. Переносимость	45
0.3. Программирование и компьютерные науки	46
0.4. Творческое начало и решение задач	46
0.5. Обратная связь	47
0.6. Библиографические ссылки	47
0.7. Биографии	48
Бьярне Страуструп	48
Лоуренс “Пит” Петерсен	49
Ждем ваших отзывов!	51
Глава 1. Компьютеры, люди и программирование	53
1.1. Введение	54
1.2. Программное обеспечение	55
1.3. Люди	58
1.4. Компьютерные науки	62
1.5. Компьютеры повсюду	62
1.5.1. С экранами и без них	63
1.5.2. Кораблестроение	64
1.5.3. Телекоммуникации	65
1.5.4. Медицина	68
1.5.5. Информация	69
1.5.6. Вид сверху	71
1.5.7. И что?	73
1.6. Идеалы программистов	73

Часть I. Основы	81
Глава 2. Hello, World!	83
2.1. Программы	84
2.2. Классическая первая программа	85
2.3. Компиляция	88
2.4. Редактирование связей	91
2.5. Среды программирования	92
Глава 3. Объекты, типы и значения	99
3.1. Ввод	100
3.2. Переменные	102
3.3. Ввод и тип	104
3.4. Операции и операторы	106
3.5. Присваивание и инициализация	109
3.5.1. Пример: выявление повторяющихся слов	111
3.6. Составные операторы присваивания	113
3.6.1. Пример: выявление повторяющихся слов	114
3.7. Имена	115
3.8. Типы и объекты	117
3.9. Безопасность типов	119
3.9.1. Безопасные преобразования	120
3.9.2. Небезопасные преобразования	121
Глава 4. Вычисления	129
4.1. Вычисления	130
4.2. Цели и средства	132
4.3. Выражения	135
4.3.1. Константные выражения	136
4.3.2. Операторы	138
4.3.3. Преобразования	140
4.4. Инструкции	141
4.4.1. Инструкции выбора	143
4.4.2. Итерация	149
4.5. Функции	154
4.5.1. Зачем нужны функции	156
4.5.2. Объявления функций	157
4.6. Вектор	158
4.6.1. Обход вектора	160
4.6.2. Увеличение вектора	160
4.6.3. Числовой пример	161
4.6.4. Текстовый пример	164
4.7. Языковые возможности	166
Глава 5. Ошибки	173
5.1. Введение	174
5.2. Источники ошибок	176

5.3. Ошибки времени компиляции	177
5.3.1. Синтаксические ошибки	178
5.3.2. Ошибки, связанные с типами	179
5.3.3. Не ошибки	180
5.4. Ошибки времени редактирования связей	181
5.5. Ошибки времени выполнения программы	182
5.5.1. Обработка ошибок в вызывающем коде	183
5.5.2. Обработка ошибок в вызываемом коде	185
5.5.3. Сообщения об ошибках	187
5.6. Исключения	188
5.6.1. Неправильные аргументы	189
5.6.2. Ошибки, связанные с диапазоном	190
5.6.3. Неправильный ввод	192
5.6.4. Сужающие преобразования	196
5.7. Логические ошибки	197
5.8. Оценка	200
5.9. Отладка	201
5.9.1. Практические советы по отладке	203
5.10. Пред- и постусловия	207
5.10.1. Постусловия	209
5.11. Тестирование	210
Глава 6. Написание программ	217
6.1. Задача	218
6.2. Размышления над задачей	219
6.2.1. Стадии разработки программы	220
6.2.2. Стратегия	220
6.3. Вернемся к калькулятору	223
6.3.1. Первая попытка	224
6.3.2. Лексемы	226
6.3.3. Реализация лексем	228
6.3.4. Использование лексем	230
6.3.5. Назад к школьной доске!	232
6.4. Грамматики	233
6.4.1. Отступление: грамматика английского языка	238
6.4.2. Написание грамматики	239
6.5. Превращение грамматики в программу	241
6.5.1. Реализация грамматических правил	241
6.5.2. Выражения	242
6.5.3. Термы	246
6.5.4. Первичные выражения	248
6.6. Испытание первой версии	249
6.7. Испытание второй версии	254
6.8. Потоки лексем	255
6.8.1. Реализация класса <code>Token_stream</code>	257

6.8.2. Чтение лексем	259
6.8.3. Считывание чисел	260
6.9. Структура программы	261
Глава 7. Завершение программы	267
7.1. Введение	268
7.2. Ввод и вывод	268
7.3. Обработка ошибок	270
7.4. Отрицательные числа	275
7.5. Остаток от деления: %	276
7.6. Приведение кода в порядок	278
7.6.1. Символические константы	278
7.6.2. Использование функций	280
7.6.3. Размещение кода	281
7.6.4. Комментарии	283
7.7. Восстановление после ошибок	285
7.8. Переменные	288
7.8.1. Переменные и определения	288
7.8.2. Использование имен	293
7.8.3. Предопределенные имена	296
7.8.4. Это все?	296
Глава 8. Технические детали: функции и прочее	301
8.1. Технические детали	302
8.2. Объявления и определения	303
8.2.1. Виды объявлений	308
8.2.2. Объявления переменных и констант	308
8.2.3. Инициализация по умолчанию	310
8.3. Заголовочные файлы	310
8.4. Область видимости	313
8.5. Вызов функции и возврат значения	319
8.5.1. Объявление аргументов и тип возвращаемого значения	319
8.5.2. Возврат значения	321
8.5.3. Передача параметров по значению	322
8.5.4. Передача параметров по константной ссылке	323
8.5.5. Передача параметров по ссылке	325
8.5.6. Сравнение механизмов передачи параметров по значению и по ссылке	328
8.5.7. Проверка аргументов и преобразование типов	331
8.5.8. Реализация вызова функции	332
8.5.9. constexpr-функции	337
8.6. Порядок вычислений	338
8.6.1. Вычисление выражения	340
8.6.2. Глобальная инициализация	340
8.7. Пространства имен	342
8.7.1. Объявления using и директивы using	343

Глава 9. Технические детали: классы и прочее	351
9.1. Типы, определенные пользователем	352
9.2. Классы и члены класса	354
9.3. Интерфейс и реализация	354
9.4. Разработка класса	356
9.4.1. Структура и функции	356
9.4.2. Функции-члены и конструкторы	358
9.4.3. Соккрытие деталей	360
9.4.4. Определение функций-членов	362
9.4.5. Ссылка на текущий объект	365
9.4.6. Сообщения об ошибках	365
9.5. Перечисления	367
9.5.1. “Простые” перечисления	369
9.6. Перегрузка операторов	370
9.7. Интерфейсы классов	371
9.7.1. Типы аргументов	372
9.7.2. Копирование	375
9.7.3. Конструкторы по умолчанию	376
9.7.4. Константные функции-члены	379
9.7.5. Члены и вспомогательные функции	381
9.8. Класс Date	383
Часть II. Ввод и вывод	391
Глава 10. Потоки ввода и вывода	393
10.1. Ввод и вывод	394
10.2. Модель потока ввода-вывода	396
10.3. Файлы	398
10.4. Открытие файла	399
10.5. Чтение и запись файла	401
10.6. Обработка ошибок ввода-вывода	403
10.7. Считывание отдельного значения	407
10.7.1. Разделение задачи на управляемые части	409
10.7.2. Отделение диалога от функции	412
10.8. Операторы вывода, определенные пользователем	413
10.9. Операторы ввода, определенные пользователем	414
10.10. Стандартный цикл ввода	415
10.11. Чтение структурированного файла	417
10.11.1. Представление в памяти	418
10.11.2. Чтение структурированных значений	420
10.11.3. Изменение представлений	424
Глава 11. Настройка ввода и вывода	429
11.1. Регулярность и нерегулярность	430
11.2. Форматирование вывода	431
11.2.1. Вывод целых чисел	431

11.2.2. Ввод целых чисел	434
11.2.3. Вывод чисел с плавающей точкой	435
11.2.4. Точность	436
11.2.5. Поля	437
11.3. Открытие файла и позиционирование	438
11.3.1. Режимы открытия файлов	439
11.3.2. Бинарные файлы	440
11.3.3. Позиционирование в файлах	443
11.4. Строковые потоки	444
11.5. Ввод, ориентированный на строки	446
11.6. Классификация символов	447
11.7. Использование нестандартных разделителей	449
11.8. И еще много чего	456
Глава 12. Модель вывода на экран	463
12.1. Почему графика?	464
12.2. Модель вывода на дисплей	465
12.3. Первый пример	467
12.4. Использование библиотеки графического пользовательского интерфейса	471
12.5. Координаты	472
12.6. Класс Shape	473
12.7. Использование примитивов Shape	474
12.7.1. Заголовочные файлы и функция main	474
12.7.2. Почти пустое окно	475
12.7.3. Оси координат	477
12.7.4. График функции	479
12.7.5. Многоугольники	480
12.7.6. Прямоугольник	481
12.7.7. Заполнение	484
12.7.8. Текст	484
12.7.9. Изображения	486
12.7.10. И многое другое	487
12.8. Запуск программы	488
12.8.1. Исходные файлы	490
Глава 13. Графические классы	495
13.1. Обзор графических классов	496
13.2. Классы Point и Line	498
13.3. Класс Lines	501
13.4. Класс Color	504
13.5. Класс Line_style	507
13.6. Класс Open_polyline	509
13.7. Класс Closed_polyline	510
13.8. Класс Polygon	512
13.9. Класс Rectangle	514

13.10. Управление неименованными объектами	519
13.11. Класс Text	521
13.12. Класс Circle	523
13.13. Класс Ellipse	525
13.14. Класс Marked_polyline	527
13.15. Класс Marks	529
13.16. Класс Mark	530
13.17. Класс Image	532
Глава 14. Проектирование графических классов	539
14.1. Принципы проектирования	540
14.1.1. Типы	540
14.1.2. Операции	542
14.1.3. Именованное	543
14.1.4. Изменяемость	545
14.2. Класс Shape	546
14.2.1. Абстрактный класс	547
14.2.2. Управление доступом	549
14.2.3. Рисование фигур	552
14.2.4. Копирование и изменчивость	556
14.3. Базовые и производные классы	557
14.3.1. Схема объекта	559
14.3.2. Порождение классов и определение виртуальных функций	561
14.3.3. Перекрытие	562
14.3.4. Доступ	564
14.3.5. Чисто виртуальные функции	565
14.4. Преимущества объектно-ориентированного программирования	567
Глава 15. Графическое представление функций и данных	575
15.1. Введение	576
15.2. Графики простых функций	576
15.3. Класс Function	580
15.3.1. Аргументы по умолчанию	582
15.3.2. Другие примеры	583
15.3.3. Лямбда-выражения	585
15.4. Оси координат	586
15.5. Аппроксимация	588
15.6. Графическое представление данных	594
15.6.1. Чтение файла	596
15.6.2. Общая схема	598
15.6.3. Масштабирование данных	598
15.6.4. Построение графика	600

Глава 16. Графические пользовательские интерфейсы	607
16.1. Альтернативы пользовательского интерфейса	608
16.2. Кнопка Next	609
16.3. Простое окно	611
16.3.1. Функции обратного вызова	613
16.3.2. Цикл ожидания	616
16.3.3. Лямбда-выражения в качестве функций обратного вызова	617
16.4. Класс Button и другие разновидности Widget	618
16.4.1. Класс Widget	618
16.4.2. Класс Button	620
16.4.3. Классы In_box и Out_box	620
16.4.4. Класс Menu	621
16.5. Пример	622
16.6. Инверсия управления	626
16.7. Добавление меню	627
16.8. Отладка GUI-программы	632
Часть III. Данные и алгоритмы	639
Глава 17. Векторы и динамически выделяемая память	641
17.1. Введение	642
17.2. Основы	644
17.3. Память, адреса и указатели	646
17.3.1. Оператор sizeof	649
17.4. Динамически распределяемая память и указатели	650
17.4.1. Размещение в динамической памяти	651
17.4.2. Доступ с помощью указателей	653
17.4.3. Диапазоны	654
17.4.4. Инициализация	656
17.4.5. Нулевой указатель	657
17.4.6. Освобождение памяти	658
17.5. Деструкторы	661
17.5.1. Генерируемые деструкторы	663
17.5.2. Деструкторы и динамическая память	664
17.6. Доступ к элементам	665
17.7. Указатели на объекты класса	666
17.8. Путаница с типами: void* и операторы приведения типов	668
17.9. Указатели и ссылки	671
17.9.1. Указатели и ссылки как параметры функций	672
17.9.2. Указатели, ссылки и наследование	673
17.9.3. Пример: списки	674
17.9.4. Операции над списками	676
17.9.5. Использование списков	677
17.10. Указатель this	679
17.10.1. Еще об использовании списков	681

Глава 18. Векторы и массивы	687
18.1. Введение	688
18.2. Инициализация	689
18.3. Копирование	691
18.3.1. Копирующие конструкторы	693
18.3.2. Копирующее присваивание	694
18.3.3. Терминология, связанная с копированием	696
18.3.4. Перемещение	697
18.4. Основные операции	700
18.4.1. Явные конструкторы	702
18.4.2. Отладка конструкторов и деструкторов	703
18.5. Доступ к элементам вектора	706
18.5.1. Константная перегрузка	707
18.6. Массивы	708
18.6.1. Указатели на элементы массива	710
18.6.2. Указатели и массивы	712
18.6.3. Инициализация массива	714
18.6.4. Проблемы с указателями	716
18.7. Примеры: палиндром	719
18.7.1. Палиндромы с использованием <code>string</code>	719
18.7.2. Палиндромы с использованием массивов	720
18.7.3. Палиндромы с использованием указателей	721
Глава 19. Векторы, шаблоны и исключения	729
19.1. Проблемы	730
19.2. Изменение размера	734
19.2.1. Представление	734
19.2.2. Функции <code>reserve</code> и <code>capacity</code>	736
19.2.3. Функция <code>resize</code>	736
19.2.4. Функция <code>push_back</code>	737
19.2.5. Присваивание	738
19.2.6. Текущее состояние дел	740
19.3. Шаблоны	741
19.3.1. Типы как шаблонные параметры	741
19.3.2. Обобщенное программирование	744
19.3.3. Концепции	747
19.3.4. Контейнеры и наследование	750
19.3.5. Целые числа как параметры шаблонов	751
19.3.6. Вывод аргументов шаблона	753
19.3.7. Обобщение класса <code>vector</code>	753
19.4. Проверка диапазона и исключения	757
19.4.1. Отступление от темы: вопросы проектирования	758
19.4.2. Признание в использовании макроса	760
19.5. Ресурсы и исключения	762
19.5.1. Потенциальные проблемы управления ресурсами	763
19.5.2. Захват ресурса — это инициализация	765

19.5.3. Гарантии	766
19.5.4. Класс <code>unique_ptr</code>	767
19.5.5. Возврат посредством перемещения	769
19.5.6. RAII для класса <code>vector</code>	770
Глава 20. Контейнеры и итераторы	777
20.1. Хранение и обработка данных	778
20.1.1. Работа с данными	779
20.1.2. Обобщение кода	780
20.2. Принципы библиотеки STL	783
20.3. Последовательности и итераторы	787
20.3.1. Вернемся к примерам	790
20.4. Связанные списки	791
20.4.1. Операции над списками	793
20.4.2. Итерация	794
20.5. Очередное обобщение класса <code>vector</code>	797
20.5.1. Обход контейнера	799
20.5.2 <code>auto</code>	799
20.6. Пример: простой текстовый редактор	801
20.6.1. Строки	803
20.6.2. Итерация	804
20.7. Классы <code>vector</code> , <code>list</code> и <code>string</code>	808
20.7.1. Операции <code>insert</code> и <code>erase</code>	810
20.8. Адаптация нашего класса <code>vector</code> к STL	813
20.9. Адаптация встроенных массивов к библиотеке STL	815
20.10. Обзор контейнеров	817
20.10.1. Категории итераторов	820
Глава 21. Алгоритмы и ассоциативные массивы	827
21.1. Алгоритмы стандартной библиотеки	828
21.2. Простейший алгоритм: <code>find()</code>	829
21.2.1. Примеры использования обобщенных алгоритмов	832
21.3. Универсальный поиск: <code>find_if()</code>	833
21.4. Функциональные объекты	835
21.4.1. Абстрактная точка зрения на функциональные объекты	836
21.4.2. Предикаты на членах класса	838
21.4.3. Лямбда-выражения	839
21.5. Численные алгоритмы	840
21.5.1. Алгоритм <code>accumulate()</code>	841
21.5.2. Обобщение алгоритма <code>accumulate()</code>	842
21.5.3. Алгоритм <code>inner_product()</code>	844
21.5.4. Обобщение алгоритма <code>inner_product()</code>	845
21.6. Ассоциативные контейнеры	846
21.6.1. Контейнер <code>map</code>	847
21.6.2. Обзор контейнера <code>map</code>	849
21.6.3. Еще один пример использования <code>map</code>	852

21.6.4. Контейнер <code>unordered_map</code>	854
21.6.5. Контейнер <code>set</code>	857
21.7. Копирование	859
21.7.1. Алгоритм <code>copy</code>	859
21.7.2. Итераторы потоков	860
21.7.3. Использование <code>set</code> для поддержки упорядоченности	863
21.7.4. Алгоритм <code>copy_if</code>	863
21.8. Сортировка и поиск	864
21.9. Алгоритмы контейнеров	866

Часть IV. Дополнительные темы 873

Глава 22. Идеалы и история 875

22.1. История, идеалы и профессионализм	876
22.1.1. Цели и философия языка программирования	877
22.1.2. Идеалы программирования	878
22.1.3. Стили и парадигмы	887
22.2. Обзор истории языков программирования	891
22.2.1. Первые языки программирования	892
22.2.2. Корни современных языков программирования	894
22.2.3. Семейство языков Algol	901
22.2.4. Язык программирования Simula	909
22.2.5. Язык программирования C	911
22.2.6. Язык программирования C++	915
22.2.7. Современное состояние дел	919
22.2.8. Источники информации	920

Глава 23. Работа с текстом 925

23.1. Текст	926
23.2. Строки	926
23.3. Потоки ввода-вывода	930
23.4. Ассоциативные контейнеры	931
23.4.1. Детали реализации	937
23.5. Проблема	939
23.6. Идея регулярных выражений	941
23.6.1. Необработанные строковые литералы	944
23.7. Поиск с помощью регулярных выражений	945
23.8. Синтаксис регулярных выражений	947
23.8.1. Символы и специальные символы	948
23.8.2. Классы символов	949
23.8.3. Повторения	950
23.8.4. Группировка	951
23.8.5. Альтернативы	951
23.8.6. Наборы символов и диапазоны	952
23.8.7. Ошибки в регулярных выражениях	954

23.9. Сопоставление регулярных выражений	955
23.10. Ссылки	960
Глава 24. Числа	965
24.1. Введение	966
24.2. Размер, точность и переполнение	967
24.2.1. Пределы числовых диапазонов	970
24.3. Массивы	971
24.4. Многомерные массивы в стиле языка C	972
24.5. Библиотека <code>Matrix</code>	974
24.5.1. Размерности и доступ	975
24.5.2. Одномерная матрица	978
24.5.3. Двумерные матрицы	981
24.5.4. Ввод-вывод матриц	984
24.5.5. Трехмерные матрицы	984
24.6. Пример: решение систем линейных уравнений	985
24.6.1. Классическое исключение Гаусса	987
24.6.2. Выбор опорного элемента	988
24.6.3. Тестирование	989
24.7. Случайные числа	991
24.8. Стандартные математические функции	994
24.9. Комплексные числа	995
24.10. Ссылки	997
Глава 25. Программирование встроенных систем	1003
25.1. Встроенные системы	1004
25.2. Основные концепции	1008
25.2.1. Предсказуемость	1011
25.2.2. Идеалы	1012
25.2.3. Сохранение работоспособности после сбоя	1013
25.3. Управление памятью	1015
25.3.1. Проблемы с динамической памятью	1017
25.3.2. Альтернативы динамической памяти	1020
25.3.3. Пример пула	1021
25.3.4. Пример стека	1023
25.4. Адреса, указатели и массивы	1024
25.4.1. Непроверяемые преобразования	1024
25.4.2. Проблема: дисфункциональный интерфейс	1025
25.4.3. Решение: интерфейсный класс	1029
25.4.4. Наследование и контейнеры	1032
25.5. Биты, байты и слова	1036
25.5.1. Операции с битами и байтами	1036
25.5.2. Класс <code>bitset</code>	1041
25.5.3. Целые числа со знаком и без знака	1042
25.5.4. Работа с битами	1047
25.5.5. Битовые поля	1049

25.5.6. Пример: простое шифрование	1051
25.6. Стандарты кодирования	1056
25.6.1. Каким должен быть стандарт кодирования?	1058
25.6.2. Примеры правил	1059
25.6.3. Реальные стандарты кодирования	1065
Глава 26. Тестирование	1073
26.1. Чего мы хотим	1074
26.1.1. Предостережение	1076
26.2. Доказательства	1076
26.3. Тестирование	1076
26.3.1. Регрессивные тесты	1078
26.3.2. Модульные тесты	1078
26.3.3. Алгоритмы и не алгоритмы	1086
26.3.4. Системные тесты	1094
26.3.5. Поиск предположений, которые не выполняются	1095
26.4. Проектирование с учетом тестирования	1097
26.5. Отладка	1098
26.6. Производительность	1099
26.6.1. Измерение времени	1101
26.7. Ссылки	1103
Глава 27. Язык программирования C	1107
27.1. Языки C и C++: братья	1108
27.1.1. Совместимость языков C и C++	1111
Ссылки	1112
27.1.2. Возможности C++, отсутствующие в C	1112
27.1.3. Стандартная библиотека языка C	1114
27.2. Функции	1115
27.2.1. Отсутствие перегрузки имен функций	1116
27.2.2. Проверка типов аргументов функций	1116
27.2.3. Определения функций	1118
27.2.4. Вызов C-функций из C++-программы и наоборот	1120
27.2.5. Указатели на функции	1122
27.3. Второстепенные языковые различия	1123
27.3.1. Дескриптор пространства имен struct	1124
27.3.2. Ключевые слова	1125
27.3.3. Определения	1125
27.3.4. Приведение типов в стиле языка C	1127
27.3.5. Преобразование указателей типа void*	1128
27.3.6. Перечисление	1129
27.3.7. Пространства имен	1130
27.4. Динамическая память	1130
27.5. Строки в стиле C	1132
27.5.1. Строки в стиле C и ключевое слово const	1135
27.5.2. Операции над байтами	1136

27.5.3. Пример: функция <code>strcpy()</code>	1136
27.5.4. Вопросы стиля	1137
27.6. Ввод-вывод: заголовочный файл <code>stdio.h</code>	1138
27.6.1. Вывод	1138
27.6.2. Ввод	1139
27.6.3. Файлы	1141
27.7. Константы и макросы	1142
27.8. Макросы	1143
27.8.1. Макросы, похожие на функции	1144
27.8.2. Синтаксические макросы	1145
27.8.3. Условная компиляция	1146
27.9. Пример: интрузивные контейнеры	1147

Часть V. Приложения 1159

Приложение А. Обзор языка	1161
А.1. Общие сведения	1162
А.1.1. Терминология	1163
А.1.2. Запуск и завершение программы	1164
А.1.3. Комментарии	1164
А.2. Литералы	1165
А.2.1. Целочисленные литералы	1165
А.2.2. Литералы с плавающей точкой	1167
А.2.3. Булевы литералы	1168
А.2.4. Символьные литералы	1168
А.2.5. Строковые литералы	1169
А.2.6. Указательные литералы	1169
А.3. Идентификаторы	1169
А.3.1. Ключевые слова	1170
А.4. Область видимости, класс памяти и время жизни	1170
А.4.1. Область видимости	1171
А.4.2. Класс памяти	1172
А.4.3. Время жизни	1173
А.5. Выражения	1174
А.5.1. Операторы, определенные пользователем	1178
А.5.2. Неявное преобразование типа	1179
А.5.3. Константные выражения	1181
А.5.4. Оператор <code>sizeof</code>	1182
А.5.5. Логические выражения	1182
А.5.6. Операторы <code>new</code> и <code>delete</code>	1182
А.5.7. Операторы приведения	1183
А.6. Инструкции	1184
А.7. Объявления	1186
А.7.1. Определения	1187
А.8. Встроенные типы	1187
А.8.1. Указатели	1188

А.8.2. Массивы	1190
А.8.3. Ссылки	1191
А.9. Функции	1191
А.9.1. Разрешение перегрузки	1192
А.9.2. Аргументы по умолчанию	1193
А.9.3. Неопределенные аргументы	1194
А.9.4. Спецификации связей	1194
А.10. Типы, определенные пользователем	1195
А.10.1. Перегрузка операций	1195
А.11. Перечисления	1196
А.12. Классы	1197
А.12.1. Доступ к членам класса	1197
А.12.2. Определения членов класса	1200
А.12.3. Создание, уничтожение и копирование	1201
А.12.4. Производные классы	1204
А.12.5. Битовые поля	1208
А.12.6. Объединения	1209
А.13. Шаблоны	1209
А.13.1. Аргументы шаблонов	1210
А.13.2. Инстанцирование шаблонов	1211
А.13.3. Шаблонные типы членов-классов	1212
А.14. Исключения	1213
А.15. Пространства имен	1215
А.16. Псевдонимы	1216
А.17. Директивы препроцессора	1216
А.17.1. Директива <code>#include</code>	1217
А.17.2. Директива <code>#define</code>	1217
Приложение Б. Обзор стандартной библиотеки	1219
Б.1. Обзор	1220
Б.1.1. Заголовочные файлы	1221
Б.1.2. Пространство имен <code>std</code>	1224
Б.1.3. Стиль описания	1224
Б.2. Обработка ошибок	1225
Б.2.1. Исключения	1225
Б.3. Итераторы	1227
Б.3.1. Модель итераторов	1227
Б.3.2. Категории итераторов	1229
Б.4. Контейнеры	1231
Б.4.1. Обзор	1233
Б.4.2. Члены-типы	1234
Б.4.3. Конструкторы, деструкторы и присваивания	1234
Б.4.4. Итераторы	1235
Б.4.5. Доступ к элементам	1235
Б.4.6. Операции над стеком и очередью	1236
Б.4.7. Операции со списком	1237

Б.4.8. Размер и емкость	1237
Б.4.9. Другие операции	1238
Б.4.10. Операции над ассоциативными контейнерами	1238
Б.5. Алгоритмы	1239
Б.5.1. Немодифицирующие алгоритмы для последовательностей	1240
Б.5.2. Алгоритмы, модифицирующие последовательности	1241
Б.5.3. Вспомогательные алгоритмы	1244
Б.5.4. Сортировка и поиск	1244
Б.5.5. Алгоритмы для множеств	1246
Б.5.6. Пирамиды	1247
Б.5.7. Перестановки	1248
Б.5.8. Функции <code>min</code> и <code>max</code>	1249
Б.6. Утилиты библиотеки STL	1250
Б.6.1. Итераторы вставки	1250
Б.6.2. Функциональные объекты	1250
Б.6.3. Классы <code>pair</code> и <code>tuple</code>	1252
Б.6.4. Список инициализации	1253
Б.6.5. Указатели управления ресурсами	1254
Б.7. Потоки ввода-вывода	1255
Б.7.1. Иерархия потоков ввода-вывода	1256
Б.7.2. Обработка ошибок	1258
Б.7.3. Операции ввода	1258
Б.7.4. Операции вывода	1259
Б.7.5. Форматирование	1260
Б.7.6. Стандартные манипуляторы	1260
Б.8. Работа со строками	1261
Б.8.1. Классификация символов	1262
Б.8.2. Строки	1262
Б.8.3. Регулярные выражения	1264
Б.9. Работа с числами	1266
Б.9.1. Предельные значения	1266
Б.9.2. Стандартные математические функции	1267
Б.9.3. Комплексные числа	1268
Б.9.4. Класс <code>valarray</code>	1269
Б.9.5. Обобщенные численные алгоритмы	1269
Б.9.6. Случайные числа	1270
Б.10. Работа со временем	1270
Б.11. Функции стандартной библиотеки языка C	1271
Б.11.1. Файлы	1271
Б.11.2. Семейство функций <code>printf()</code>	1272
Б.11.3. C-строки	1277
Б.11.4. Память	1278
Б.11.5. Дата и время	1279
Б.11.6. Другие функции	1280
Б.12. Другие библиотеки	1281

Приложение В. Начала работы с Visual Studio	1283
В.1. Запуск программы	1284
В.2. Инсталляция Visual Studio	1284
В.3. Создание и запуск программ	1285
В.3.1. Создание нового проекта	1285
В.3.2. Использование заголовочного файла <code>std_lib_facilities.h</code>	1285
В.3.3. Добавление в проект исходного файла на языке C++	1286
В.3.4. Ввод исходного кода	1286
В.3.5. Создание выполнимой программы	1286
В.3.6. Выполнение программы	1287
В.3.7. Сохранение программы	1287
В.4. Что дальше	1287
Приложение Г. Установка FLTK	1289
Г.1. Введение	1290
Г.2. Загрузка библиотеки FLTK	1290
Г.3. Установка библиотеки FLTK	1291
Г.4. Использование библиотеки FLTK в среде Visual Studio	1292
Г.5. Тестирование, все ли работает	1292
Приложение Д. Реализация графического пользовательского интерфейса	1295
Д.1. Реализация обратных вызовов	1296
Д.2. Реализация класса <code>Widget</code>	1297
Д.3. Реализация класса <code>Window</code>	1298
Д.4. Реализация класса <code>Vector_ref</code>	1300
Д.5. Пример: работа с объектами <code>Widget</code>	1301
Глоссарий	1305
Библиография	1313
Предметный указатель	1316
Фотографии	1327