

Содержание

Благодарности	23
Об авторе	23
Поддержка читателя	24
Ждем ваших отзывов!	24
ВВЕДЕНИЕ	25
Для кого написана эта книга	25
Структура книги	25
Соглашения, принятые в книге	26
Примеры кода	27
ЧАСТЬ I. Основы C++	29
ЗАНЯТИЕ 1. Первые шаги	31
Краткий экскурс в историю языка C++	32
Связь с языком C	32
Преимущества языка C++	32
Развитие стандарта C++	33
Кто использует программы, написанные на C++	33
Создание приложения C++	33
Этапы создания исполнимого файла	34
Анализ и устранение ошибок	34
Интегрированные среды разработки	34
Создание первого приложения на C++	35
Построение и запуск вашего первого приложения C++	36
Понятие ошибок компиляции	38
Что нового в C++	38
Резюме	39
Вопросы и ответы	39
Коллоквиум	40
Контрольные вопросы	40
Упражнения	40
ЗАНЯТИЕ 2. Структура программы на C++	41
Части программы Hello World	42
Директива препроцессора #include	42
Тело программы — функция main ()	43
Возврат значения	44
Концепция пространств имен	44
Комментарии в коде C++	46
Функции в C++	47
Ввод-вывод с использованием потоков std::cin и std::cout	50
Резюме	52
Вопросы и ответы	52
Коллоквиум	52
Контрольные вопросы	53
Упражнения	53

ЗАНЯТИЕ 3. Использование переменных и констант	55
Что такое переменная	56
Коротко о памяти и адресации	56
Объявление переменных для получения доступа и использования памяти	56
Объявление и инициализация нескольких переменных одного типа	58
Понятие области видимости переменной	59
Глобальные переменные	61
Соглашения об именовании	62
Распространенные типы переменных, поддерживаемые компилятором C++	63
Использование типа <code>bool</code> для хранения логических значений	64
Использование типа <code>char</code> для хранения символьных значений	64
Концепция знаковых и беззнаковых целых чисел	65
Знаковые целочисленные типы <code>short</code> , <code>int</code> , <code>long</code> и <code>long long</code>	66
Беззнаковые целочисленные типы <code>unsigned short</code> , <code>unsigned int</code> , <code>unsigned long</code> и <code>unsigned long long</code>	66
Избегайте переполнения, выбирая подходящие типы	67
Типы с плавающей точкой <code>float</code> и <code>double</code>	69
Определение размера переменной с использованием оператора <code>sizeof</code>	69
Запрет сужающего преобразования при использовании инициализации списком	71
Автоматический вывод типа с использованием <code>auto</code>	72
Использование ключевого слова <code>typedef</code> для замены типа	73
Что такое константа	74
Литеральные константы	74
Объявление переменных как констант с использованием ключевого слова <code>const</code>	75
Объявление констант с использованием ключевого слова <code>constexpr</code>	76
Перечисления	78
Определение констант с использованием директивы <code>#define</code>	80
Ключевые слова, недопустимые для использования в качестве имен переменных и констант	80
Резюме	81
Вопросы и ответы	82
Коллоквиум	84
Контрольные вопросы	84
Упражнения	84
ЗАНЯТИЕ 4. Массивы и строки	85
Что такое массив	86
Необходимость в массивах	86
Объявление и инициализация статических массивов	87
Как данные хранятся в массиве	88
Доступ к данным, хранимым в массиве	89
Изменение данных в массиве	90
Многомерные массивы	93
Объявление и инициализация многомерных массивов	93
Доступ к элементам многомерного массива	94
Динамические массивы	95
Строки символов в стиле C	97

Строки C++: использование <code>std::string</code>	100
Резюме	101
Вопросы и ответы	102
Коллоквиум	103
Контрольные вопросы	103
Упражнения	103
ЗАНЯТИЕ 5. Выражения, инструкции и операторы	105
Выражения	106
Составные инструкции, или блоки	107
Использование операторов	107
Оператор присваивания (=)	107
Понятие l- и r-значений	107
Операторы сложения (+), вычитания (-), умножения (*), деления (/) и деления по модулю (%)	108
Операторы инкремента (++) и декремента (--)	109
Что значит “постфиксный” и “префиксный”	109
Операторы равенства (==) и неравенства (!=)	111
Операторы сравнения	111
Логические операции НЕ, И, ИЛИ и ИСКЛЮЧАЮЩЕЕ ИЛИ	114
Использование логических операторов C++ !, && и	115
Побитовые операторы ~, &, и ^	119
Побитовые операторы сдвига вправо (>>) и влево (<<)	121
Составные операторы присваивания	122
Использование оператора <code>sizeof</code> для определения объема памяти, занимаемого переменной	124
Приоритет операторов	125
Резюме	127
Вопросы и ответы	127
Коллоквиум	128
Контрольные вопросы	128
Упражнения	128
ЗАНЯТИЕ 6. Управление потоком выполнения программы	129
Условное программирование с использованием конструкции <code>if...else</code>	131
Условное выполнение нескольких инструкций	133
Вложенные инструкции <code>if</code>	134
Условная обработка с использованием конструкции <code>switch-case</code>	138
Тернарный условный оператор (?:)	141
Выполнение кода в циклах	142
Рудиментарный цикл с использованием инструкции <code>goto</code>	143
Цикл <code>while</code>	145
Цикл <code>do...while</code>	146
Цикл <code>for</code>	148
Цикл <code>for</code> для диапазона	151
Изменение поведения цикла с использованием операторов <code>continue</code> и <code>break</code>	153
Бесконечные циклы, которые никогда не заканчиваются	154
Управление бесконечными циклами	154

Программирование вложенных циклов	157
Использование вложенных циклов для перебора многомерного массива	158
Использование вложенных циклов для вычисления чисел Фибоначчи	160
Резюме	161
Вопросы и ответы	162
Коллоквиум	162
Контрольные вопросы	163
Упражнения	163
ЗАНЯТИЕ 7. Организация кода с помощью функций	165
Потребность в функциях	166
Что такое прототип функции	167
Что такое определение функции	168
Что такое вызов функции и аргументы	168
Создание функций с несколькими параметрами	169
Создание функций без параметров и возвращаемых значений	170
Параметры функций со значениями по умолчанию	171
Рекурсия — функция, вызывающая сама себя	173
Функции с несколькими операторами <code>return</code>	175
Использование функций для работы с данными различных видов	176
Перегрузка функций	177
Передача в функцию массива значений	178
Передача аргументов по ссылке	180
Как процессор обрабатывает вызовы функций	182
Встраиваемые функции	183
Автоматический вывод возвращаемого типа	184
Лямбда-функции	186
Резюме	187
Вопросы и ответы	188
Коллоквиум	188
Контрольные вопросы	188
Упражнения	189
ЗАНЯТИЕ 8. Указатели и ссылки	191
Что такое указатель	192
Объявление указателя	192
Определение адреса переменной с использованием оператора получения адреса <code>&</code>	193
Использование указателей для хранения адресов	194
Доступ к данным с использованием оператора разыменования <code>*</code>	197
Значение <code>sizeof()</code> для указателя	199
Динамическое распределение памяти	201
Использование <code>new</code> и <code>delete</code> для выделения и освобождения памяти	201
Указатели и операции инкремента и декремента	204
Использование ключевого слова <code>const</code> с указателями	207
Передача указателей в функции	208
Сходство между массивами и указателями	209
Наиболее распространенные ошибки при использовании указателей	212
Утечки памяти	212

Когда указатели указывают на недопустимые области памяти	213
Висячие (беспризорные, дикие) указатели	214
Проверка успешности запроса с использованием оператора <code>new</code>	215
Полезные советы по применению указателей	218
Что такое ссылка	218
Зачем нужны ссылки	220
Использование ключевого слова <code>const</code> со ссылками	221
Передача аргументов в функции по ссылке	221
Резюме	223
Вопросы и ответы	223
Коллоквиум	225
Контрольные вопросы	225
Упражнения	225
часть II. Объектно-ориентированное программирование на C++	227
ЗАНЯТИЕ 9. Классы и объекты	229
Концепция классов и объектов	230
Объявление класса	230
Объект как экземпляр класса	231
Доступ к членам класса с использованием оператора точки (<code>.</code>)	232
Обращение к членам класса с использованием оператора указателя (<code>-></code>)	232
Ключевые слова <code>public</code> и <code>private</code>	234
Абстракция данных с помощью ключевого слова <code>private</code>	236
Конструкторы	237
Объявление и реализация конструктора	237
Когда и как использовать конструкторы	238
Перегрузка конструкторов	240
Класс без конструктора по умолчанию	242
Параметры конструктора со значениями по умолчанию	243
Конструкторы со списками инициализации	244
Деструктор	246
Объявление и реализация деструктора	246
Когда и как использовать деструкторы	247
Копирующий конструктор	250
Поверхностное копирование и связанные с ним проблемы	250
Глубокое копирование с использованием копирующего конструктора	252
Перемещающий конструктор улучшает производительность	257
Способы использования конструкторов и деструктора	258
Класс, который не разрешает себя копировать	258
Класс-синглтон, обеспечивающий наличие только одного экземпляра	259
Класс, запрещающий создание экземпляра в стеке	262
Применение конструкторов для преобразования типов	264
Указатель <code>this</code>	266
Размер класса	267
Чем структура отличается от класса	269
Объявление друзей класса	270

Специальный механизм хранения данных — <code>union</code>	272
Объявление объединения	272
Где используется объединение	273
Агрегатная инициализация классов и структур	275
<code>constexpr</code> с классами и объектами	278
Резюме	279
Вопросы и ответы	279
Коллоквиум	280
Контрольные вопросы	280
Упражнения	281
ЗАНЯТИЕ 10. Реализация наследования	283
Основы наследования	284
Наследование и порождение	284
Синтаксис наследования C++	286
Модификатор доступа <code>protected</code>	288
Инициализация базового класса — передача параметров базовому классу	290
Перекрытие методов базового класса в производном	293
Вызов перекрытых методов базового класса	295
Вызов методов базового класса в производном классе	296
Производный класс, скрывающий методы базового класса	298
Порядок конструирования	300
Порядок деструкции	300
Закрытое наследование	303
Защищенное наследование	305
Проблема срезки	308
Множественное наследование	309
Запрет наследования с помощью ключевого слова <code>final</code>	311
Резюме	313
Вопросы и ответы	313
Коллоквиум	313
Контрольные вопросы	314
Упражнения	314
ЗАНЯТИЕ 11. Полиморфизм	315
Основы полиморфизма	316
Потребность в полиморфном поведении	316
Полиморфное поведение, реализованное с помощью виртуальных функций	318
Необходимость виртуальных деструкторов	320
Как работают виртуальные функции. Понятие таблицы виртуальных функций	324
Абстрактные классы и чисто виртуальные функции	328
Использование виртуального наследования для решения проблемы ромба	330
Ключевое слово <code>override</code> для указания преднамеренного перекрытия	335
Использование ключевого слова <code>final</code> для предотвращения перекрытия функции	336
Виртуальные копирующие конструкторы?	336
Резюме	340
Вопросы и ответы	340

Коллоквиум	341
Контрольные вопросы	341
Упражнения	342
ЗАНЯТИЕ 12. Типы операторов и их перегрузка	343
Что такое операторы C++	344
Унарные операторы	345
Типы унарных операторов	345
Программирование унарного оператора инкремента или декремента	345
Создание операторов преобразования	348
Создание оператора разыменования (*) и оператора выбора члена (->)	351
Бинарные операторы	353
Типы бинарных операторов	353
Создание бинарных операторов сложения (a+b) и вычитания (a-b)	354
Реализация операторов сложения с присваиванием (+=)	
и вычитания с присваиванием (-=)	357
Перегрузка операторов равенства (==) и неравенства (!=)	359
Перегрузка операторов <, >, <= и >=	361
Перегрузка оператора копирующего присваивания (=)	363
Оператор индексации ([])	366
Оператор функции ()	369
Перемещающий конструктор и оператор перемещающего присваивания	370
Проблема излишнего копирования	370
Объявление перемещающих конструктора и оператора присваивания	371
Пользовательские литералы	376
Операторы, которые не могут быть перегружены	378
Резюме	379
Вопросы и ответы	379
Коллоквиум	380
Контрольные вопросы	380
Упражнения	380
ЗАНЯТИЕ 13. Операторы приведения	381
Потребность в приведении типов	382
Почему приведения в стиле C не нравятся некоторым программистам C++	383
Операторы приведения C++	383
Использование оператора <code>static_cast</code>	384
Использование оператора <code>dynamic_cast</code> и идентификация типа	
времени выполнения	385
Использование оператора <code>reinterpret_cast</code>	388
Использование оператора <code>const_cast</code>	389
Проблемы с операторами приведения C++	390
Резюме	392
Вопросы и ответы	392
Коллоквиум	392
Контрольные вопросы	393
Упражнения	393

ЗАНЯТИЕ 14. Введение в макросы и шаблоны	395
Препроцессор и компилятор	396
Использование <code>#define</code> для определения констант	396
Использование макроса для защиты от множественного включения	399
Использование директивы <code>#define</code> для написания макрофункции	400
Зачем все эти скобки?	402
Использование макроса <code>assert</code> для проверки выражений	402
Преимущества и недостатки использования макрофункций	404
Введение в шаблоны	405
Синтаксис объявления шаблона	406
Типы объявлений шаблонов	406
Шаблонные функции	407
Шаблоны и безопасность типов	409
Шаблонные классы	409
Объявление шаблонов с несколькими параметрами	410
Объявление шаблонов параметрами по умолчанию	411
Простой шаблон класса <code>holdsPair</code>	411
Инстанцирование и специализация шаблона	413
Шаблонные классы и статические члены	415
Шаблоны с переменным количеством параметров (вариадические шаблоны)	416
Использование <code>static_assert</code> для выполнения проверок времени компиляции	420
Использование шаблонов в практическом программировании на C++	421
Резюме	422
Вопросы и ответы	422
Коллоквиум	423
Контрольные вопросы	423
Упражнения	423
Часть III. Стандартная библиотека шаблонов	425
ЗАНЯТИЕ 15. Введение в стандартную библиотеку шаблонов	427
Контейнеры STL	428
Последовательные контейнеры	428
Ассоциативные контейнеры	429
Адаптеры контейнеров	431
Итераторы STL	431
Алгоритмы STL	432
Взаимодействие контейнеров и алгоритмов с использованием итераторов	432
Использование ключевого слова <code>auto</code> для определения типа	434
Выбор правильного контейнера	435
Классы строк библиотеки STL	437
Резюме	437
Вопросы и ответы	437
Коллоквиум	438
Контрольные вопросы	438

ЗАНЯТИЕ 16. Класс строки библиотеки STL	439
Потребность в классах обработки строк	440
Работа с классом строки STL	441
Создание экземпляров и копий строк STL	441
Доступ к символу в строке <code>std::string</code>	443
Конкатенация строк	445
Поиск символа или подстроки в строке	446
Усечение строк STL	448
Обращение строки	450
Смена регистра символов	451
Реализация строки на базе шаблона STL	453
Оператор <code>""s</code> в <code>std::string</code> в C++14	453
Резюме	454
Вопросы и ответы	455
Коллоквиум	455
Контрольные вопросы	455
Упражнения	455
ЗАНЯТИЕ 17. Классы динамических массивов библиотеки STL	457
Характеристики класса <code>std::vector</code>	458
Типичные операции с вектором	458
Создание экземпляра вектора	458
Вставка элементов в конец вектора с помощью <code>push_back()</code>	460
Инициализация списком	461
Вставка элементов в определенную позицию с помощью <code>insert()</code>	461
Доступ к элементам вектора с использованием семантики массива	464
Доступ к элементам вектора с использованием семантики указателя	465
Удаление элементов из вектора	466
Концепции размера и емкости	468
Класс <code>deque</code> библиотеки STL	470
Резюме	473
Вопросы и ответы	473
Коллоквиум	474
Контрольные вопросы	474
Упражнения	474
ЗАНЯТИЕ 18. Классы <code>list</code> и <code>forward_list</code>	475
Характеристики класса <code>std::list</code>	476
Основные операции со списком	476
Инстанцирование класса <code>std::list</code>	476
Вставка элементов в начало и в конец списка	478
Вставка в середину списка	479
Удаление элементов из списка	482
Обращение списка и сортировка его элементов	483
Обращение элементов списка с помощью <code>list::reverse()</code>	484
Сортировка элементов	485
Сортировка и удаление элементов из списка, который содержит объекты класса	487
Шаблон класса <code>std::forward_list</code>	490

Резюме	492
Вопросы и ответы	492
Коллоквиум	493
Контрольные вопросы	493
Упражнения	493
ЗАНЯТИЕ 19. Классы множеств STL	495
Введение в классы множеств STL	496
Фундаментальные операции с классами <code>set</code> и <code>multiset</code>	496
Инстанцирование объекта <code>std::set</code>	497
Вставка элементов в множество и мультимножество	499
Поиск элементов в множестве и мультимножестве	500
Удаление элементов из множества и мультимножества	502
Преимущества и недостатки использования множеств и мультимножеств	507
Реализация хеш-множеств <code>std::unordered_set</code> и <code>std::unordered_multiset</code>	507
Резюме	511
Вопросы и ответы	511
Коллоквиум	511
Контрольные вопросы	512
Упражнения	512
ЗАНЯТИЕ 20. Классы отображений библиотеки STL	513
Введение в классы отображений библиотеки STL	514
Фундаментальные операции с классами <code>std::map</code> и <code>std::multimap</code>	515
Инстанцирование классов <code>std::map</code> и <code>std::multimap</code>	515
Вставка элементов в <code>map</code> и <code>multimap</code>	517
Поиск элементов в отображении	519
Поиск элементов в мультиотображении STL	522
Удаление элементов из <code>map</code> и <code>multimap</code>	522
Применение пользовательского предиката	524
Контейнер для пар “ключ–значение” на базе хеш-таблиц	528
Как работают хеш-таблицы	528
Использование <code>unordered_map</code> и <code>unordered_multimap</code>	529
Резюме	533
Вопросы и ответы	533
Коллоквиум	534
Контрольные вопросы	534
Упражнения	534
ЧАСТЬ IV. Углубляемся в STL	535
ЗАНЯТИЕ 21. Понятие о функциональных объектах	537
Концепция функциональных объектов и предикатов	538
Типичные приложения функциональных объектов	538
Унарные функции	538
Унарный предикат	543
Бинарные функции	545
Бинарный предикат	547

Резюме	550
Вопросы и ответы	550
Коллоквиум	550
Контрольные вопросы	550
Упражнения	551
ЗАНЯТИЕ 22. Лямбда-выражения языка C++11	553
Что такое лямбда-выражение	554
Как определить лямбда-выражение	555
Лямбда-выражение для унарной функции	555
Лямбда-выражение для унарного предиката	557
Лямбда-выражения с состоянием и списки захвата [. . .]	558
Обобщенный синтаксис лямбда-выражений	560
Лямбда-выражение для бинарной функции	561
Лямбда-выражение для бинарного предиката	563
Резюме	565
Вопросы и ответы	565
Коллоквиум	566
Контрольные вопросы	566
Упражнения	566
ЗАНЯТИЕ 23. Алгоритмы библиотеки STL	567
Что такое алгоритмы STL	568
Классификация алгоритмов STL	568
Не изменяющие алгоритмы	568
Изменяющие алгоритмы	569
Использование алгоритмов STL	571
Поиск элементов по заданному значению или условию	571
Подсчет элементов с использованием значения или условия	573
Поиск элемента или диапазона в коллекции	575
Инициализация элементов в контейнере заданным значением	577
Использование алгоритма <code>std::generate()</code>	
для инициализации значениями, генерируемыми во время выполнения	579
Обработка элементов диапазона с использованием алгоритма <code>for_each()</code>	581
Выполнение преобразований с помощью алгоритма <code>std::transform()</code>	583
Операции копирования и удаления	585
Замена значений и элементов с использованием условия	588
Сортировка, поиск в отсортированной коллекции и удаление дубликатов	590
Разделение диапазона	592
Вставка элементов в отсортированную коллекцию	594
Резюме	597
Вопросы и ответы	597
Коллоквиум	598
Контрольные вопросы	598
Упражнения	598
ЗАНЯТИЕ 24. Адаптивные контейнеры: стек и очередь	599
Поведенческие характеристики стеков и очередей	600
Стеки	600
Очереди	600

Использование класса STL <code>stack</code>	601
Создание экземпляра стека	601
Функции-члены класса <code>stack</code>	602
Вставка и извлечение с помощью методов <code>push()</code> и <code>pop()</code>	603
Использование класса STL <code>queue</code>	605
Создание экземпляра очереди	605
Функции-члены класса <code>queue</code>	606
Вставка в конец и извлечение из начала очереди с использованием методов <code>push()</code> и <code>pop()</code>	607
Использование класса STL <code>priority_queue</code>	608
Создание экземпляра очереди с приоритетами	608
Функции-члены класса <code>priority_queue</code>	610
Вставка в конец и извлечение из начала очереди с приоритетами с использованием методов <code>push()</code> и <code>pop()</code>	611
Резюме	613
Вопросы и ответы	613
Коллоквиум	613
Контрольные вопросы	614
Упражнения	614
ЗАНЯТИЕ 25. Работа с битовыми флагами при использовании библиотеки STL	615
Класс <code>bitset</code>	616
Инстанцирование класса <code>std::bitset</code>	616
Использование класса <code>std::bitset</code> и его членов	617
Полезные операторы, предоставляемые классом <code>std::bitset</code>	618
Методы класса <code>std::bitset</code>	618
Класс <code>vector<bool></code>	621
Создание экземпляра класса <code>vector<bool></code>	621
Функции и операторы класса <code>vector<bool></code>	622
Резюме	623
Вопросы и ответы	623
Коллоквиум	624
Контрольные вопросы	624
Упражнения	624
ЧАСТЬ V. Сложные концепции C++	625
ЗАНЯТИЕ 26. Понятие интеллектуальных указателей	627
Что такое интеллектуальный указатель	628
Проблемы обычных указателей	628
Чем могут помочь интеллектуальные указатели	628
Как реализованы интеллектуальные указатели	629
Типы интеллектуальных указателей	630
Глубокое копирование	631
Механизм копирования при записи	633
Интеллектуальные указатели со счетчиком ссылок	633
Интеллектуальный указатель со списком ссылок	634

Деструктивное копирование	634
Использование интеллектуального указателя <code>std::unique_ptr</code>	637
Популярные библиотеки интеллектуальных указателей	639
Резюме	639
Вопросы и ответы	639
Коллоквиум	640
Контрольные вопросы	640
Упражнения	640
ЗАНЯТИЕ 27. Применение потоков для ввода и вывода	641
Концепция потоков	642
Важнейшие классы и объекты потоков C++	643
Использование <code>std::cout</code> для вывода форматированных данных на консоль	644
Изменение формата представления чисел	645
Выравнивание текста и установка ширины поля	647
Использование <code>std::cin</code> для ввода	648
Использование <code>std::cin</code> для ввода простых старых типов данных	648
Использование метода <code>std::cin::get()</code> для ввода в буфер <code>char*</code>	649
Использование <code>std::cin</code> для ввода в переменную типа <code>std::string</code>	650
Использование потока <code>std::fstream</code> для работы с файлом	652
Открытие и закрытие файла с помощью методов <code>open()</code> и <code>close()</code>	652
Создание и запись текстового файла с использованием метода <code>open()</code> и оператора <code><<</code>	653
Чтение текстового файла с использованием метода <code>open()</code> и оператора <code>>></code>	655
Запись и чтение из бинарного файла	656
Использование <code>std::stringstream</code> для преобразования строк	658
Резюме	660
Вопросы и ответы	660
Коллоквиум	660
Контрольные вопросы	660
Упражнения	661
ЗАНЯТИЕ 28. Обработка исключений	663
Что такое исключение	664
Что вызывает исключения	664
Реализация безопасности в отношении исключений с помощью блоков <code>try</code> и <code>catch</code>	665
Использование блока <code>catch(...)</code> для обработки всех исключений	665
Обработка исключения конкретного типа	666
Генерация исключения с помощью оператора <code>throw</code>	668
Как работает обработка исключений	669
Класс <code>std::exception</code>	671
Пользовательский класс исключения, производный от <code>std::exception</code>	672
Резюме	674
Вопросы и ответы	675
Коллоквиум	675
Контрольные вопросы	676
Упражнения	676

ЗАНЯТИЕ 29. Что дальше	677
Чем отличаются современные процессоры	678
Как лучше использовать несколько ядер	679
Что такое поток	679
Зачем создавать многопоточные приложения	680
Как потоки осуществляют транзакцию данных	681
Использование мьютексов и семафоров для синхронизации потоков	682
Проблемы, вызываемые многопоточностью	682
Как писать отличный код C++	683
C++17: что новенького	684
Инициализация в if и switch	684
Гарантия устранения копирования	685
Устранение накладных расходов выделения памяти	
с помощью <code>std::string_view</code>	686
<code>std::variant</code> как безопасная с точки зрения типов альтернатива объединению	686
Условная компиляция с использованием <code>if constexpr</code>	687
Усовершенствованные лямбда-выражения	688
Автоматический вывод типа для конструкторов	688
<code>template<auto></code>	688
Изучение C++ на этом не заканчивается	688
Документация в вебе	688
Сетевые сообщества и помощь	689
Резюме	689
Вопросы и ответы	689
Коллоквиум	690
Контрольные вопросы	690
ЧАСТЬ VI. Приложения	691
ПРИЛОЖЕНИЕ А. Двоичные и шестнадцатеричные числа	693
Десятичная система счисления	694
Двоичная система счисления	694
Почему компьютеры используют двоичные числа	695
Что такое биты и байты	695
Сколько байтов в килобайте	695
Шестнадцатеричная система счисления	696
Зачем нужна шестнадцатеричная система	696
Преобразование в различные системы счисления	697
Обобщенный процесс преобразования	697
Преобразование десятичного числа в двоичное	697
Преобразование десятичного числа в шестнадцатеричное	698
ПРИЛОЖЕНИЕ Б. Ключевые слова языка C++	699
ПРИЛОЖЕНИЕ В. Приоритет операторов	701
ПРИЛОЖЕНИЕ Г. Коды ASCII	703
Таблица ASCII отображаемых символов	704

ПРИЛОЖЕНИЕ Д. Ответы	707
Ответы к занятию 1	707
Контрольные вопросы	707
Упражнения	707
Ответы к занятию 2	708
Контрольные вопросы	708
Упражнения	708
Ответы к занятию 3	709
Контрольные вопросы	709
Упражнения	709
Ответы к занятию 4	710
Контрольные вопросы	710
Упражнения	711
Ответы к занятию 5	711
Контрольные вопросы	711
Упражнения	712
Ответы к занятию 6	712
Контрольные вопросы	712
Упражнения	713
Ответы к занятию 7	716
Контрольные вопросы	716
Упражнения	716
Ответы к занятию 8	717
Контрольные вопросы	717
Упражнения	717
Ответы к занятию 9	717
Контрольные вопросы	717
Упражнения	718
Ответы к занятию 10	719
Контрольные вопросы	719
Упражнения	719
Ответы к занятию 11	720
Контрольные вопросы	720
Упражнения	720
Ответы к занятию 12	722
Контрольные вопросы	722
Упражнения	722
Ответы к занятию 13	723
Контрольные вопросы	723
Упражнения	723
Ответы к занятию 14	724
Контрольные вопросы	724
Упражнения	724
Ответы к занятию 15	725
Контрольные вопросы	725
Ответы к занятию 16	726
Контрольные вопросы	726
Упражнения	726

Ответы к занятию 17	729
Контрольные вопросы	729
Упражнения	729
Ответы к занятию 18	732
Контрольные вопросы	732
Упражнения	733
Ответы к занятию 19	734
Контрольные вопросы	734
Упражнения	734
Ответы к занятию 20	737
Контрольные вопросы	737
Упражнения	738
Ответы к занятию 21	738
Контрольные вопросы	738
Упражнения	738
Ответы к занятию 22	740
Контрольные вопросы	740
Упражнения	740
Ответы к занятию 23	741
Контрольные вопросы	741
Упражнения	742
Ответы к занятию 24	743
Контрольные вопросы	743
Упражнения	743
Ответы к занятию 25	743
Контрольные вопросы	743
Упражнения	744
Ответы к занятию 26	744
Контрольные вопросы	744
Упражнения	744
Ответы к занятию 27	745
Контрольные вопросы	745
Упражнения	746
Ответы к занятию 28	746
Контрольные вопросы	746
Упражнения	746
Ответы к занятию 29	746
Контрольные вопросы	746

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ**747**