

Содержание

Введение	19
Структура книги	19
Соглашения, принятые в книге	20
Примеры кода	21
От издательства	21
ЧАСТЬ I. Основы	23
ЗАНЯТИЕ 1. Первые шаги	25
Краткий экскурс в историю языка C++	26
Связь с языком C	26
Преимущества языка C++	26
Развитие стандарта C++	26
Кто использует программы, написанные на C++?	27
Создание приложения C++	27
Этапы создания исполняемого файла	27
Анализ и устранение ошибок	28
Интегрированная среда разработки	28
Создание первого приложения C++	29
Создание и запуск вашего первого приложения C++	30
Понятие ошибок компиляции	31
Что нового в C++11	31
Резюме	32
Вопросы и ответы	32
Коллоквиум	33
Контрольные вопросы	33
Упражнения	33
ЗАНЯТИЕ 2. Структура программы на C++	35
Части программы Hello World	36
Директива препроцессора <code>#include</code>	36
Тело программы — функция <code>main()</code>	37
Возвращение значения	38
Концепция пространств имен	38
Комментарии в коде C++	40
Функции в C++	40
Простые операторы ввода <code>std::cin</code> и вывода <code>std::cout</code>	43
Резюме	45
Вопросы и ответы	45
Коллоквиум	45
Контрольные вопросы	45
Упражнения	46
ЗАНЯТИЕ 3. Использование переменных, объявление констант	47
Что такое переменная	48
Коротко о памяти и адресации	48
Объявление переменных для получения доступа и использования памяти	48
Объявление и инициализация нескольких переменных одного типа	50

Понятие области видимости переменной	50
Глобальные переменные	52
Популярные типы переменных, поддерживаемые компилятором C++	54
Использование типа <code>bool</code> для хранения логических значений	54
Использование типа <code>char</code> для хранения символьных значений	55
Концепция знаковых и беззнаковых целых чисел	55
Знаковые целочисленные типы <code>short</code> , <code>int</code> , <code>long</code> и <code>long long</code>	56
Беззнаковые целочисленные типы <code>unsigned short</code> , <code>unsigned int</code> , <code>unsigned long</code> и <code>unsigned long long</code>	56
Типы с плавающей точкой <code>float</code> и <code>double</code>	57
Определение размера переменной с использованием оператора <code>sizeof</code>	57
Использование ключевого слова <code>typedef</code> для замены типа переменной	61
Что такое константа	61
Литеральные константы	62
Объявление переменных как констант с использованием ключевого слова <code>const</code>	62
Перечисляемые константы	64
Определение констант с использованием директивы <code>#define</code>	66
Именованные переменные и константы	66
Ключевые слова, недопустимые для использования в качестве имен переменных и констант	67
Резюме	68
Вопросы и ответы	68
Коллоквиум	70
Контрольные вопросы	70
Упражнения	70
ЗАНЯТИЕ 4. Массивы и строки	71
Что такое массив	72
Необходимость в массивах	72
Объявление и инициализация статических массивов	73
Как данные хранятся в массиве	74
Доступ к данным, хранимым в массиве	75
Изменение хранимых в массиве данных	76
Многомерные массивы	78
Объявление и инициализация многомерных массивов	79
Доступ к элементам в многомерном массиве	79
Динамические массивы	81
Строки в стиле C	82
Строки C++: использование типа <code>std::string</code>	85
Резюме	86
Вопросы и ответы	87
Коллоквиум	87
Контрольные вопросы	88
Упражнения	88
ЗАНЯТИЕ 5. Команды, выражения и операторы	89
Выражения	90
Составные выражения, или блоки	91
Использование операторов	91
Оператор присвоения (<code>=</code>)	91
Понятие l- и r-значений	91
Операторы суммы (<code>+</code>), вычитания (<code>-</code>), умножения (<code>*</code>), деления (<code>/</code>) и деления по модулю (<code>%</code>)	91
Операторы инкремента (<code>++</code>) и декремента (<code>--</code>)	93
Что значит постфиксный и префиксный	93

Операторы равенства (==) и неравенства (!=)	96
Операторы сравнения	96
Логические операции NOT, AND, OR и XOR	98
Использование логических операторов C++ NOT (!), AND (&&) и OR ()	99
Побитовые операторы NOT (~), AND (&), OR () и XOR (^)	102
Побитовые операторы сдвига вправо (>>) и влево (<<)	104
Составные операторы присвоения	105
Использование оператора sizeof для определения объема памяти, занятого переменной	107
Приоритет операторов	108
Резюме	110
Вопросы и ответы	110
Коллоквиум	111
Контрольные вопросы	111
Упражнения	111
ЗАНЯТИЕ 6. Ветвление процесса выполнения программ	113
Условное выполнение с использованием конструкции if...else	114
Условное программирование с использованием конструкции if...else	115
Условное выполнение нескольких операторов	117
Вложенные операторы if	118
Условная обработка с использованием конструкции switch-case	122
Троичный условный оператор (?:)	124
Выполнение кода в циклах	126
Рудиментарный цикл с использованием оператора goto	126
Цикл while	128
Цикл do...while	129
Цикл for	131
Изменение поведения цикла с использованием операторов continue и break	134
Циклы, которые не заканчиваются никогда, т.е. бесконечные циклы	135
Контроль бесконечных циклов	135
Программирование вложенных циклов	138
Использование вложенных циклов для перебора многомерного массива	139
Использование вложенных циклов для вычисления чисел Фибоначчи	141
Резюме	142
Вопросы и ответы	142
Коллоквиум	143
Контрольные вопросы	143
Упражнения	143
ЗАНЯТИЕ 7. Организация кода при помощи функций	145
Потребность в функциях	146
Что такое прототип функции	147
Что такое определение функции	148
Что такое вызов функции и аргументы	148
Создание функций с несколькими параметрами	148
Создание функций без параметров и возвращаемых значений	150
Параметры функций со значениями по умолчанию	151
Рекурсия — функция, вызывающая сама себя	152
Функции с несколькими операторами return	154
Использование функций для работы с данными различных форм	155
Перегрузка функции	155
Передача функции массива значений	157
Передача аргументов по ссылке	158

Как процессор обрабатывает вызовы функций	160
Встраиваемые функции	161
Лямбда-функции	163
Резюме	164
Вопросы и ответы	164
Коллоквиум	165
Контрольные вопросы	165
Упражнения	165
ЗАНЯТИЕ 8. Указатели и ссылки	167
Что такое указатель	168
Объявление указателя	168
Определение адреса переменной с использованием оператора ссылки (&)	169
Использование указателей для хранения адресов	170
Доступ к данным с использованием оператора обращения к значению (*)	172
Каков результат выполнения оператора sizeof() для указателя?	174
Динамическое распределение памяти	175
Использование операторов new и delete для динамического резервирования и освобождения памяти	176
Воздействие операторов инкремента и декремента (++ и --) на указатели	179
Использование ключевого слова const с указателями	182
Передача указателей в функции	183
Сходство между массивами и указателями	184
Наиболее распространенные ошибки при использовании указателей	186
Утечки памяти	187
Когда указатели указывают на недопустимые области памяти	187
Потерянные указатели (они же беспризорные или дикие)	188
Полезные советы по применению указателей	189
Проверка успешности запроса с использованием оператора new	190
Что такое ссылка	193
Зачем нужны ссылки	194
Использование ключевого слова const со ссылками	195
Передача аргументов в функции по ссылке	195
Резюме	197
Вопросы и ответы	197
Коллоквиум	198
Контрольные вопросы	198
Упражнения	199
часть II. Фундаментальные принципы объектно-ориентированного программирования на C++	201
ЗАНЯТИЕ 9. Классы и объекты	203
Концепция классов и объектов	204
Объявление класса	204
Создание экземпляра объекта класса	205
Доступ к членам класса с использованием точечного оператора (.)	205
Доступ к членам класса с использованием оператора указателя (->)	206
Ключевые слова public и private	208
Абстракция данных при помощи ключевого слова private	209
Конструкторы	211
Объявление и реализация конструктора	211
Когда и как использовать конструкторы	212

Перегрузка конструкторов	214
Класс без стандартного конструктора	216
Параметры конструктора со значениями по умолчанию	217
Конструкторы со списками инициализации	219
Деструктор	220
Объявление и реализация деструктора	220
Когда и как использовать деструкторы	221
Конструктор копий	223
Поверхностное копирование и связанные с ним проблемы	223
Обеспечение глубокого копирования с использованием конструктора копий	226
Конструктор перемещения улучшает производительность	230
Различные способы использования конструкторов и деструкторов	232
Класс, который не разрешает себя копировать	232
Синглетонный класс, разрешающий создание только одного экземпляра	233
Класс, запрещающий создание экземпляра в стеке	235
Указатель <code>this</code>	237
Размер класса	238
Чем структура отличается от класса	240
Объявление друзей класса	241
Резюме	243
Вопросы и ответы	243
Коллоквиум	244
Контрольные вопросы	244
Упражнения	245
ЗАНЯТИЕ 10. Реализация наследования	247
Основы наследования	248
Наследование и происхождение	248
Синтаксис наследования C++	250
Модификатор доступа <code>protected</code>	252
Инициализация базового класса — передача параметров для базового класса	254
Производный класс, переопределяющий методы базового класса	256
Вызов переопределенных методов базового класса	258
Вызов методов базового класса в производном классе	259
Производный класс, скрывающий методы базового класса	261
Порядок создания	263
Порядок удаления	263
Закрытое наследование	265
Защищенное наследование	267
Проблема отсечения	270
Множественное наследование	271
Резюме	273
Вопросы и ответы	274
Коллоквиум	274
Контрольные вопросы	274
Упражнения	275
ЗАНЯТИЕ 11. Полиморфизм	277
Основы полиморфизма	278
Потребность в полиморфном поведении	278
Полиморфное поведение, реализованное при помощи виртуальных функций	279
Потребность в виртуальных деструкторах	281
Как работают виртуальные функции. Понятие таблицы виртуальной функции	285
Абстрактные классы и чистые виртуальные функции	288

Использование виртуального наследования для решения проблемы ромба	291
Виртуальные конструкторы копий?	295
Резюме	298
Вопросы и ответы	298
Коллоквиум	299
Контрольные вопросы	299
Упражнения	300
ЗАНЯТИЕ 12. Типы операторов и их перегрузка	301
Что такое операторы C++	302
Унарные операторы	303
Типы унарных операторов	303
Создание унарного оператора инкремента или декремента	303
Создание операторов преобразования	306
Создание оператора обращения к значению (*) и оператора обращения к члену класса (->)	308
Бинарные операторы	312
Типы бинарных операторов	312
Создание бинарных операторов сложения (a + b) и вычитания (a - b)	313
Реализация операторов сложения с присвоением (+=) и вычитания с присвоением (--=)	315
Перегрузка операторов равенства (==) и неравенства (!=)	317
Перегрузка операторов <, >, <= и >=	320
Перегрузка оператора присвоения копии (=)	322
Оператор индексирования ([])	325
Оператор функции ()	328
Операторы, которые не могут быть перегружены	335
Резюме	336
Вопросы и ответы	336
Коллоквиум	337
Контрольные вопросы	337
Упражнения	337
ЗАНЯТИЕ 13. Операторы приведения	339
Потребность в приведении типов	340
Почему приведения в стиле C не нравятся некоторым программистам C++	340
Операторы приведения C++	341
Использование оператора <code>static_cast</code>	341
Использование оператора <code>dynamic_cast</code> и идентификация типа времени выполнения	342
Использование оператора <code>reinterpret_cast</code>	345
Использование оператора <code>const_cast</code>	346
Проблемы с операторами приведения C++	347
Резюме	349
Вопросы и ответы	349
Коллоквиум	349
Контрольные вопросы	349
Упражнения	350
ЗАНЯТИЕ 14. Макросы и шаблоны	351
Препроцессор и компилятор	352
Использование директивы <code>#define</code> для определения константы	352
Использование макроса для защиты от множественного включения	354
Использование директивы <code>#define</code> для написания макрофункции	355
Зачем все эти скобки?	357

Использование макроса <code>assert</code> для проверки выражений	358
Преимущества и недостатки использования макрофункций	359
Введение в шаблоны	360
Синтаксис объявления шаблона	361
Различные типы объявлений шаблона	361
Шаблон функции	362
Шаблоны и безопасность типов	364
Шаблон класса	364
Создание и специализация экземпляра шаблона	365
Объявление шаблонов с несколькими параметрами	366
Объявление шаблонов с заданными по умолчанию параметрами	366
Простой шаблон класса <code>HoldPair</code>	367
Шаблоны классов и статические члены	368
Использование шаблонов в практическом программировании на C++	371
Резюме	371
Вопросы и ответы	372
Коллоквиум	372
Контрольные вопросы	372
Упражнения	373

ЧАСТЬ III. Знакомство со стандартной библиотекой шаблонов (STL) 375

ЗАНЯТИЕ 15. Введение в стандартную библиотеку шаблонов 377

Контейнеры STL	378
Последовательные контейнеры	378
Ассоциативные контейнеры	379
Выбор правильного контейнера	380
Итераторы STL	382
Алгоритмы STL	383
Взаимодействие контейнеров и алгоритмов с использованием итераторов	384
Классы строк библиотеки STL	386
Резюме	386
Вопросы и ответы	387
Коллоквиум	387
Контрольные вопросы	387

ЗАНЯТИЕ 16. Классы строк библиотеки STL 389

Потребность в классах обработки строк	390
Работа с классами строк библиотеки STL	391
Создание экземпляров и копий строк STL	391
Доступ к символу в строке класса <code>std::string</code>	393
Конкатенация строк	395
Поиск символа или подстроки в строке	396
Усечение строк STL	398
Обращение строки	400
Смена регистра символов	401
Реализация строки на базе шаблона STL	402
Резюме	403
Вопросы и ответы	403
Коллоквиум	403
Контрольные вопросы	403
Упражнения	404

ЗАНЯТИЕ 17. Классы динамических массивов библиотеки STL	405
Характеристики класса <code>std::vector</code>	406
Типичные операции с вектором	406
Создание экземпляра вектора	406
Вставка элементов в конец с использованием метода <code>push_back()</code>	408
Вставка элементов в определенную позицию с использованием метода <code>insert()</code>	409
Доступ к элементам вектора с использованием семантики массива	412
Доступ к элементам вектора с использованием семантики указателя	413
Удаление элементов из вектора	414
Концепция размера и емкости	416
Класс <code>deque</code> библиотеки STL	418
Резюме	420
Вопросы и ответы	421
Коллоквиум	421
Контрольные вопросы	421
Упражнения	422
ЗАНЯТИЕ 18. Классы двухсвязного и односвязного списков библиотеки STL	423
Характеристики класса <code>std::list</code>	424
Основные операции со списком	424
Создание экземпляра класса <code>std::list</code>	424
Вставка элементов в начало и в конец списка	426
Вставка в середину списка	428
Удаление элементов из списка	430
Обращение списка и сортировка его элементов	431
Обращение элементов списка с использованием метода <code>list::reverse()</code>	432
Сортировка элементов	433
Сортировка и удаление элементов из списка, который содержит объекты класса	435
Резюме	440
Вопросы и ответы	441
Коллоквиум	441
Контрольные вопросы	441
Упражнения	442
ЗАНЯТИЕ 19. Классы наборов библиотеки STL	443
Введение в классы наборов библиотеки STL	444
Простые операции с классами <code>set</code> и <code>multiset</code> библиотеки STL	444
Создание экземпляра объекта <code>std::set</code>	444
Вставка элементов в набор и мультимножество	447
Поиск элементов в наборе и мультимноестве	449
Удаление элементов в наборе и мультимноестве	450
Преимущества и недостатки использования наборов и мультимноеств	455
Резюме	458
Вопросы и ответы	459
Коллоквиум	459
Контрольные вопросы	459
Упражнения	460
ЗАНЯТИЕ 20. Классы карт библиотеки STL	461
Введение в классы карт библиотеки STL	462
Простые операции с классами <code>std::map</code> и <code>std::multimap</code> библиотеки STL	463
Создание экземпляров классов <code>std::map</code> и <code>std::multimap</code>	463
Вставка элементов в карту или мультикарту библиотеки STL	465
Поиск элементов в карте STL	467

Поиск элементов в мультикарте STL	470
Стирание элементов из карты или мультикарты STL	470
Предоставление специального предиката сортировки	472
Как работают хеш-таблицы	476
Использование хеш-таблиц C++11: <code>unordered_map</code> и <code>unordered_multimap</code>	477
Резюме	480
Вопросы и ответы	481
Коллоквиум	481
Контрольные вопросы	482
Упражнения	482
часть IV. Подробней о библиотеке STL	483
ЗАНЯТИЕ 21. Понятие объектов функций	485
Концепция объектов функций и предикатов	486
Типичные приложения объектов функций	486
Унарные функции	486
Унарный предикат	490
Бинарные функции	492
Бинарный предикат	495
Резюме	497
Вопросы и ответы	497
Коллоквиум	497
Контрольные вопросы	498
Упражнения	498
ЗАНЯТИЕ 22. Лямбда-выражения языка C++11	499
Что такое лямбда-выражение	500
Как определить лямбда-выражение	500
Лямбда-выражение для унарной функции	501
Лямбда-выражение для унарного предиката	502
Лямбда-выражение с состоянием и списки захвата [. . .]	504
Обобщенный синтаксис лямбда-выражений	505
Лямбда-выражение для бинарной функции	507
Лямбда-выражение для бинарного предиката	508
Резюме	511
Вопросы и ответы	511
Коллоквиум	512
Контрольные вопросы	512
Упражнения	512
ЗАНЯТИЕ 23. Алгоритмы библиотеки STL	513
Что такое алгоритмы STL	514
Классификация алгоритмов STL	514
Не изменяющие алгоритмы	514
Изменяющие алгоритмы	515
Использование алгоритмов STL	517
Поиск элементов по заданному значению или условию	517
Подсчет элементов по заданному значению или условию	519
Поиск элемента или диапазона в коллекции	521
Инициализация элементов в контейнере заданным значением	523
Использование алгоритма <code>std::generate()</code> для инициализации элементов значениями, созданными во время выполнения	525

Обработка элементов диапазона с использованием алгоритма <code>for_each()</code>	526
Выполнение преобразований в диапазоне с использованием алгоритма <code>std::transform()</code>	528
Операции копирования и удаления	531
Замена значений и элементов по заданному условию	534
Сортировка, поиск в отсортированной коллекции и удаление дубликатов	535
Разделение диапазона	538
Вставка элементов в отсортированную коллекцию	540
Резюме	542
Вопросы и ответы	542
Коллоквиум	543
Контрольные вопросы	543
Упражнения	544
ЗАНЯТИЕ 24. Адаптивные контейнеры: стек и очередь	545
Поведенческие характеристики стеков и очередей	546
Стеки	546
Очереди	546
Использование класса <code>stack</code> библиотеки STL	547
Создание экземпляра стека	547
Функции-члены класса <code>stack</code>	548
Вставка и извлечение из вершины с использованием методов <code>push()</code> и <code>pop()</code>	549
Использование класса <code>queue</code> библиотеки STL	550
Создание экземпляра очереди	550
Функции-члены класса <code>queue</code>	552
Вставка в конец и извлечение из начала очереди с использованием методов <code>push()</code> и <code>pop()</code>	552
Использование класса <code>priority_queue</code> библиотеки STL	554
Создание экземпляра приоритетной очереди	554
Функции-члены класса <code>priority_queue</code>	555
Вставка в конец и извлечение из начала приоритетной очереди с использованием методов <code>push()</code> и <code>pop()</code>	556
Резюме	558
Вопросы и ответы	558
Коллоквиум	558
Контрольные вопросы	559
Упражнения	559
ЗАНЯТИЕ 25. Работа с битовыми флагами при использовании библиотеки STL	561
Класс <code>bitset</code>	562
Создание экземпляра класса <code>std::bitset</code>	562
Использование класса <code>std::bitset</code> и его членов	563
Вспомогательные операторы, предоставляемые классом <code>std::bitset</code>	563
Методы класса <code>std::bitset</code>	564
Класс <code>vector<bool></code>	566
Создание экземпляра класса <code>vector<bool></code>	567
Функции и операторы класса <code>vector<bool></code>	567
Резюме	569
Вопросы и ответы	569
Коллоквиум	569
Контрольные вопросы	569
Упражнения	570

часть V. Передовые концепции языка C++	571
ЗАНЯТИЕ 26. Понятие интеллектуальных указателей	573
Что такое интеллектуальный указатель	574
Проблема с использованием обычных (простых) указателей	574
Чем помогут интеллектуальные указатели	574
Как реализованы интеллектуальные указатели	575
Типы интеллектуальных указателей	576
Глубокое копирование	577
Механизм копирования при записи	578
Интеллектуальные указатели подсчета ссылок	579
Интеллектуальный указатель списка ссылок	580
Деструктивное копирование	580
Популярные библиотеки интеллектуальных указателей	584
Резюме	584
Вопросы и ответы	584
Коллоквиум	585
Контрольные вопросы	585
Упражнения	585
ЗАНЯТИЕ 27. Применение потоков для ввода и вывода	587
Концепция потоков	588
Важнейшие классы и объекты потоков C++	588
Использование объекта <code>std::cout</code> для вывода отформатированных данных на консоль	590
Изменение формата представления чисел	590
Выравнивание текста и установка ширины поля с использованием объекта <code>std::cout</code>	592
Использование объекта <code>std::cin</code> для ввода	593
Использование объекта <code>std::cin</code> для ввода старых простых типов данных	593
Использование метода <code>std::cin::get()</code> для ввода в символьный буфер стиля C	594
Использование объекта <code>std::cin</code> для ввода в переменную типа <code>std::string</code>	595
Использование объекта <code>std::fstream</code> для работы с файлом	597
Открытие и закрытие файла с использованием методов <code>open()</code> и <code>close()</code>	597
Создание и запись текстового файла с использованием метода <code>open()</code> и оператора <code><<</code>	598
Чтение текстового файла с использованием метода <code>open()</code> и оператора <code>>></code>	599
Запись и чтение из двоичного файла	600
Использование объекта <code>std::stringstream</code> для преобразования строк	602
Резюме	604
Вопросы и ответы	604
Коллоквиум	604
Контрольные вопросы	605
Упражнения	605
ЗАНЯТИЕ 28. Обработка исключений	607
Что такое исключение	608
Что вызывает исключения	608
Реализация устойчивости к исключениям при помощи блоков <code>try</code> и <code>catch</code>	608
Использование блока <code>catch(...)</code> для обработки всех исключений	609
Обработка исключения конкретного типа	610
Передача исключения конкретного типа с использованием оператора <code>throw</code>	612
Как действует обработка исключений	613
Класс <code>std::exception</code>	615
Ваш собственный класс исключения, производный от класса <code>std::exception</code>	616

Резюме	618
Вопросы и ответы	618
Коллоквиум	619
Контрольные вопросы	619
Упражнения	619
ЗАНЯТИЕ 29. Что дальше	621
Чем отличаются современные процессоры	622
Как лучше использовать несколько ядер	623
Что такое поток	623
Зачем создавать многопоточные приложения	624
Как потоки осуществляют транзакцию данных	625
Использование мьютексов и семафоров для синхронизации потоков	626
Проблемы, вызванные многопоточностью	626
Как писать отличный код C++	627
Изучение C++ на этом не заканчивается	628
Сетевая документация	628
Сетевые сообщества и помощь	629
Резюме	629
Вопросы и ответы	629
Коллоквиум	630
Контрольные вопросы	630
часть VI. Приложения	631
ПРИЛОЖЕНИЕ А. Двоичные и шестнадцатеричные числа	633
Десятичная система счисления	634
Двоичная система счисления	634
Почему компьютеры используют двоичные числа	635
Что такое биты и байты	635
Сколько байт в килобайте	635
Шестнадцатеричная система счисления	635
Зачем нужна шестнадцатеричная система	636
Преобразование в различные системы счисления	636
Обобщенный процесс преобразования	636
Преобразование десятичного числа в двоичное	637
Преобразование десятичного числа в шестнадцатеричное	637
ПРИЛОЖЕНИЕ Б. Ключевые слова языка C++	639
ПРИЛОЖЕНИЕ В. Приоритет операторов	641
ПРИЛОЖЕНИЕ Г. Ответы	643
ПРИЛОЖЕНИЕ Д. Коды ASCII	681
Таблица ASCII отображаемых символов	682
Предметный указатель	685