
СОДЕРЖАНИЕ

ВВЕДЕНИЕ	15
Новый подход к программированию на C++	15
Наша книга полезна как для новичков, так и для опытных программистов	15
Абстракция	16
Охват материала	16
Несколько слов для опытных C- и C++-программистов	17
Структура книги	17
Как получить максимальную пользу от этой книги	18
Благодарности	18
ГЛАВА 0. ИТАК, НАЧНЕМ	19
0.1. Комментарии	19
0.2. Директива #include	19
0.3. Функция main	20
0.4. Фигурные скобки	20
0.5. Использование стандартной библиотеки для вывода данных	21
0.6. Инструкция return	21
0.7. Копнем чуть-чуть глубже	22
0.8. Резюме	23
Упражнения	25
ГЛАВА 1. РАБОТА СО СТРОКАМИ	27
1.1. Ввод данных	27
1.2. Выделение текста с помощью рамок	29
1.3. Резюме	32
Упражнения	34
ГЛАВА 2. ОРГАНИЗАЦИЯ ЦИКЛОВ И ВЫЧИСЛЕНИЙ	37
2.1. В чем суть проблемы	37
2.2. Общая структура программы	37
2.3. Вывод неизвестного числа строк	38
2.3.1. Понятие о while-инструкции	39
2.3.2. Разработка while-инструкции	40
2.4. Вывод строки	42
2.4.1. Вывод символов рамки	43
2.4.2. Вывод символов, не относящихся к рамочке	45
2.5. Полная программа вывода приветствия в рамочке	46
2.5.1. Устранение многократного повторения префикса std::	47
2.5.2. Использование for-инструкций для компактности кода	47
2.5.3. Объединение нескольких проверок в одну	48
2.5.4. Полная программа вывода приветствия в рамочке	49

2.6. С чего начать отсчет	50
2.7. Резюме	51
Упражнения	54
ГЛАВА 3. РАБОТА С ГРУППАМИ ДАННЫХ	57
3.1. Вычисление оценок студентов	57
3.1.1. Как определить конец ввода данных	61
3.1.2. Инвариант цикла	62
3.2. Использование медианы вместо среднеарифметического	63
3.2.1. Сохранение коллекции данных в векторе	63
3.2.2. Генерирование выходных данных	65
3.2.3. Еще несколько замечаний	69
3.3. Резюме	70
Упражнения	71
ГЛАВА 4. ОРГАНИЗАЦИЯ ПРОГРАММ И ДАННЫХ	73
4.1. Организация вычислений	73
4.1.1. Вычисление медиан	75
4.1.2. Пересмотр политики вычисления оценок	76
4.1.3. Считывание оценок за выполнение домашних заданий	78
4.1.4. Три вида параметров функции	80
4.1.5. Использование функций для вычисления итоговой оценки студента	81
4.2. Организация данных	83
4.2.1. Соберем-ка все данные о студентах в одну кучу!	84
4.2.2. Управление записями с данными о студентах	84
4.2.3. Построение отчета	86
4.3. А теперь соберем все вместе	88
4.4. Декомпозиция программы вычисления итоговых оценок	90
4.5. Исправленная версия программы вычисления итоговых оценок	92
4.6. Резюме	93
Упражнения	95
ГЛАВА 5. ИСПОЛЬЗОВАНИЕ ПОСЛЕДОВАТЕЛЬНЫХ КОНТЕЙНЕРОВ	97
5.1. Разделение студентов на категории	97
5.1.1. Удаление элементов из вектора	98
5.1.2. Последовательный и произвольный доступ к данным	101
5.2. Итераторы	101
5.2.1. Типы итераторов	102
5.2.2. Операции, выполняемые итераторами	103
5.2.3. О некоторых синтаксических нюансах	103
5.2.4. Что означает выражение <code>students.erase(students.begin() + i)</code>	104
5.3. Использование итераторов вместо индексов	104
5.4. Изменение структуры данных для повышения производительности	106
5.5. Тип <code>list</code>	107
5.5.1. На некоторые различия стоит обратить особое внимание	108
5.5.2. Зачем так беспокоиться?	108
5.6. Разберем <code>string</code> -объект на части	109
5.7. Тестирование функции <code>split</code>	112
5.8. Сборка <code>string</code> -объектов	113

5.8.1. Опять рамочка	114
5.8.2. Вертикальная конкатенация	115
5.8.3. Горизонтальная конкатенация	116
5.9. Резюме	118
Упражнения	121
ГЛАВА 6. ИСПОЛЬЗОВАНИЕ БИБЛИОТЕЧНЫХ АЛГОРИТМОВ	125
6.1. Анализ string-объектов	125
6.1.1. Еще один вариант функции split	127
6.1.2. Палиндромы	129
6.1.3. Поиск URL-адресов	129
6.2. Сравнение схем вычисления оценок	134
6.2.1. Обработка записей с оценками студентов	134
6.2.2. Анализ оценок	135
6.2.3. Вычисление итоговых оценок на основе среднего арифметического значения оценок за домашние задания	139
6.2.4. Медиана оценок, полученных за выполненные домашние задания	140
6.3. Новый вариант классификации студентов	141
6.3.1. Решение в два прохода	141
6.3.2. Решение в один проход	143
6.4. Алгоритмы, контейнеры и итераторы	144
6.5. Резюме	145
Упражнения	146
ГЛАВА 7. ИСПОЛЬЗОВАНИЕ АССОЦИАТИВНЫХ КОНТЕЙНЕРОВ	149
7.1. Контейнеры, поддерживающие эффективный поиск	149
7.2. Подсчет слов	150
7.3. Генерирование таблицы перекрестных ссылок	152
7.4. Генерирование предложений	155
7.4.1. Представление правил	157
7.4.2. Чтение грамматических правил	157
7.4.3. Генерирование предложения	158
7.4.4. Выбор случайного элемента	161
7.5. Вспомним о производительности	163
7.6. Резюме	163
Упражнения	164
ГЛАВА 8. СОЗДАНИЕ ОБОБЩЕННЫХ ФУНКЦИЙ	167
8.1. Что такое обобщенная функция	167
8.1.1. Медианы неизвестного типа	168
8.1.2. Реализация шаблонов	170
8.1.3. Обобщенные функции и типы	170
8.2. Независимость структур данных	171
8.2.1. Алгоритмы и итераторы	172
8.2.2. Последовательный доступ только для чтения	173
8.2.3. Последовательный доступ только для записи	174
8.2.4. Последовательный доступ для чтения и записи	175
8.2.5. Двухнаправленный доступ	176
8.2.6. Произвольный доступ	176
8.2.7. Диапазоны итераторов и оконечные значения	177

8.3. Входные и выходные итераторы	178
8.4. Использование итераторов для повышения гибкости программирования	180
8.5. Резюме	181
Упражнения	182
ГЛАВА 9. ОПРЕДЕЛЕНИЕ НОВЫХ ТИПОВ	185
9.1. Вернемся к структуре Student_info	185
9.2. Типы классов	186
9.2.1. Функции-члены	187
9.2.2. Функции-не-члены	189
9.3. Средства защиты	189
9.3.1. Функции доступа	191
9.3.2. Тестирование на пустоту	192
9.4. Класс Student_info	193
9.5. Конструкторы	193
9.5.1. Конструктор по умолчанию	195
9.5.2. Конструкторы с аргументами	196
9.6. Использование класса Student_info	196
9.7. Резюме	197
Упражнения	198
ГЛАВА 10. УПРАВЛЕНИЕ ПАМЯТЬЮ И ИСПОЛЬЗОВАНИЕ СТРУКТУР ДАНЫХ НИЗКОГО УРОВНЯ	199
10.1. Указатели и массивы	199
10.1.1. Указатели	200
10.1.2. Указатели на функции	201
10.1.3. Массивы	203
10.1.4. Арифметические операции с указателями	204
10.1.5. Индексирование	205
10.1.6. Инициализация массивов	206
10.2. Снова о строковых литералах	206
10.3. Инициализация массивов указателей на символы	207
10.4. Аргументы для функции main	209
10.5. Чтение и запись файлов	210
10.5.1. Стандартный поток ошибок	210
10.5.2. Использование нескольких входных и выходных файлов	210
10.6. Три вида управления памятью	212
10.6.1. Размещение объекта в памяти и освобождение этой памяти	213
10.6.2. Размещение в памяти массива	214
10.7. Резюме	215
Упражнения	216
ГЛАВА 11. ОПРЕДЕЛЕНИЕ АБСТРАКТНЫХ ТИПОВ ДАННЫХ	217
11.1. Класс Vec	217
11.2. Реализация класса Vec	218
11.2.1. Распределение памяти	219
11.2.2. Конструкторы	220
11.2.3. Определение типов	221
11.2.4. Индексирование и определение размера	222
11.2.5. Операции, возвращающие итераторы	224

11.3. Управление копированием	225
11.3.1. Конструктор копирования	225
11.3.2. Присваивание	226
11.3.3. Присваивание — это не инициализация	228
11.3.4. Деструктор	230
11.3.5. Операции по умолчанию	231
11.3.6. Тройное правило	231
11.4. Динамические Vec-объекты	232
11.5. Гибкое управление памятью	233
11.5.1. Конечный вариант класса Vec	235
11.6. Резюме	239
Упражнения	240
ГЛАВА 12. СОЗДАНИЕ ОБЪЕКТОВ КЛАССОВ, ИСПОЛЪЗУЕМЫХ КАК ЗНАЧЕНИЯ	241
12.1. Простой класс string	242
12.2. Автоматические преобразования	243
12.3. Операции над Str-объектами	244
12.3.1. Операторы ввода-вывода	245
12.3.2. “Друзья”	246
12.3.3. Другие бинарные операторы	248
12.3.4. Выражения смешанного типа	249
12.3.5. Разработка бинарных операторов	250
12.4. Некоторые преобразования просто опасны	251
12.5. Операторы преобразования	252
12.6. Преобразования и управление памятью	253
12.7. Резюме	255
Упражнения	256
ГЛАВА 13. НАСЛЕДОВАНИЕ И ДИНАМИЧЕСКОЕ СВЯЗЫВАНИЕ	257
13.1. Наследование	257
13.1.1. Снова о защите	259
13.1.2. Операции	259
13.1.3. Наследование и конструкторы	261
13.2. Полиморфизм и виртуальные функции	262
13.2.1. Получение значения при неизвестном типе объекта	263
13.2.2. Динамическое связывание	264
13.2.3. Подведем некоторые итоги	266
13.3. Использование наследования для решения нашей “старой” задачи	267
13.3.1. Контейнеры (фактически) неизвестного типа	269
13.3.2. Виртуальные деструкторы	272
13.4. Простой дескрипторный класс	273
13.4.1. Считывание дескриптора	275
13.4.2. Копирование дескрипторных объектов	275
13.5. Использование дескрипторного класса	277
13.6. Вникнем в некоторые подробности	278
13.6.1. Наследование и контейнеры	278
13.6.2. Какая функция вам нужна	279
13.7. Резюме	280
Упражнения	281

ГЛАВА 14. ПОЧТИ АВТОМАТИЧЕСКОЕ УПРАВЛЕНИЕ ПАМЯТЬЮ	283
14.1. Дескрипторы, которые копируют свои объекты	283
14.1.1. Обобщенный дескрипторный класс	284
14.1.2. Использование обобщенного дескриптора	287
14.2. Дескрипторы с подсчетом количества ссылок	290
14.3. Дескрипторы для решения проблемы совместного использования данных	293
14.4. Усовершенствование управляемых дескрипторов	294
14.4.1. Копирование типов, которыми мы не можем управлять	296
14.4.2. Когда в копии есть необходимость	297
14.5. Резюме	298
Упражнения	298
ГЛАВА 15. ВОЗВРАЩАЯСЬ К СИМВОЛЬНЫМ ИЗОБРАЖЕНИЯМ	299
15.1. Проект системы	299
15.1.1. Использование наследования при моделировании структуры	300
15.1.2. Класс Pic_base	302
15.1.3. Производные классы	304
15.1.4. Управление копированием	307
15.2. Реализация	307
15.2.1. Реализация пользовательского интерфейса	307
15.2.2. Класс String_Pic	310
15.2.3. Дополнение выходных данных пробелами	311
15.2.4. Класс VCat_Pic	312
15.2.5. Класс HCat_Pic	313
15.2.6. Класс Frame_Pic	314
15.2.7. Не забывайте о друзьях	315
15.3. Резюме	317
Упражнения	318
ГЛАВА 16. КУДА ТЕПЕРЬ ДЕРЖАТЬ НАМ ПУТЬ	319
16.1. Используйте уже освоенные абстракции	319
16.2. Стремитесь узнать больше	321
Упражнения	322
ПРИЛОЖЕНИЕ А. ЯЗЫК C++ (ПОДРОБНО)	323
A.1. Объявления	323
A.1.1. Спецификаторы	324
A.1.2. Описатели	325
A.2. Типы	328
A.2.1. Целые типы	328
A.2.2. Тип значений с плавающей точкой	331
A.2.3. Константные выражения	331
A.2.4. Преобразования	332
A.2.5. Перечислимые типы	333
A.2.6. Перегрузка	333
A.3. Выражения	334
A.3.1. Операторы	337
A.4. Инструкции	337

ПРИЛОЖЕНИЕ Б. СТАНДАРТНАЯ БИБЛИОТЕКА (КРАТКИЙ ОБЗОР)	341
Б.1. Ввод-вывод информации	341
Б.2. Контейнеры и итераторы	344
Б.2.1. Общие контейнерные операции	344
Б.2.2. Последовательные контейнеры	345
Б.2.3. Дополнительные последовательные операции	346
Б.2.4. Ассоциативные контейнеры	346
Б.2.5. Итераторы	347
Б.2.6. Класс vector	349
Б.2.7. Класс list	349
Б.2.8. Класс string	350
Б.2.9. Класс pair	351
Б.2.10. Класс map	351
Б.3. Алгоритмы	352
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	357