

# Оглавление

Об авторе .....	21
О научном редакторе.....	22
Предисловие.....	23
Для кого предназначена эта книга.....	23
О чем эта книга .....	23
Как получить максимальную пользу от этой книги.....	26
Загрузите файлы с примерами кода.....	27
Где скачать цветные иллюстрации.....	27
Условные обозначения .....	27
От издательства .....	28
<b>Глава 1. Go и операционная система.....</b>	<b>29</b>
История Go .....	29
Куда движется Go? .....	30
Преимущества Go .....	30
Идеален ли Go? .....	31
Утилита godoc.....	32
Компиляция Go-кода.....	33
Выполнение Go-кода .....	34
Два правила Go .....	35
Правило пакетов Go: не нужен — не подключай.....	35
Правильный вариант размещения фигурных скобок — всего один.....	36
Как скачивать Go-пакеты .....	37

---

Стандартные потоки UNIX: stdin, stdout и stderr .....	39
Вывод результатов .....	39
Использование стандартного потока вывода .....	41
Получение данных от пользователя.....	42
Что такое := и = .....	42
Чтение стандартного потока ввода .....	43
Работа с аргументами командной строки .....	45
Вывод ошибок .....	47
Запись в журнальные файлы .....	49
Уровни журналирования .....	49
Средства журналирования .....	50
Серверы журналов.....	50
Пример Go-программы, которая записывает информацию в журнальные файлы.....	51
Функция log.Fatal().....	54
Функция log.Panic() .....	54
Запись в специальный журнальный файл .....	56
Вывод номеров строк в записях журнала .....	58
Обработка ошибок в Go .....	59
Тип данных error.....	59
Обработка ошибок.....	61
Использование Docker .....	64
Упражнения и ссылки .....	68
Резюме .....	69
<b>Глава 2. Go изнутри .....</b>	<b>70</b>
Компилятор Go .....	71
Сборка мусора .....	72
Трехцветный алгоритм.....	74
Подробнее о работе сборщика мусора Go .....	78
Хеш-таблицы, срезы и сборщик мусора Go.....	79
Небезопасный код .....	82

Пакет unsafe .....	84
Еще один пример использования пакета unsafe .....	84
Вызов С-кода из Go .....	86
Вызов С-кода из Go в одном файле .....	86
Вызов из Go С-кода в отдельных файлах .....	87
С-код .....	87
Go-код .....	88
Сочетание кода на Go и С .....	89
Вызов Go-функций из С-кода .....	90
Go-пакет .....	90
С-код .....	91
Ключевое слово defer .....	92
Использование defer для журналирования .....	95
Функции panic() и recover() .....	97
Самостоятельное использование функции panic() .....	98
Две полезные UNIX-утилиты .....	99
Утилита strace .....	100
Утилита dtrace .....	101
Среда Go .....	102
Команда go env .....	104
Go-ассемблер .....	105
Узловые деревья .....	106
Хотите знать больше о go build? .....	111
Создание кода WebAssembly .....	113
Краткое введение в WebAssembly .....	113
Почему WebAssembly так важен .....	113
Go и WebAssembly .....	114
Пример .....	114
Использование сгенерированного кода WebAssembly .....	115
Общие рекомендации по программированию на Go .....	117
Упражнения и ссылки .....	118
Резюме .....	118

---

<b>Глава 3. Работа с основными типами данных Go</b> .....	120
Числовые типы данных .....	120
Целые числа .....	121
Числа с плавающей точкой .....	121
Комплексные числа .....	121
Числовые литералы в Go 2 .....	123
Циклы Go .....	124
Цикл for .....	124
Цикл while .....	125
Ключевое слово range .....	125
Пример применения нескольких циклов Go .....	125
Массивы в Go .....	127
Многомерные массивы .....	128
Недостатки массивов Go .....	130
Срезы в Go .....	131
Выполнение основных операций со срезами .....	131
Автоматическое расширение срезов .....	133
Байтовые срезы .....	135
Функция <code>copy()</code> .....	135
Многомерные срезы .....	137
Еще один пример использования срезов .....	137
Сортировка срезов с помощью <code>sort.Slice()</code> .....	139
Добавление массива к срезу .....	141
Хеш-таблицы Go .....	142
Запись в хеш-таблицу со значением <code>nil</code> .....	144
Когда использовать хеш-таблицы .....	145
Константы Go .....	145
Генератор констант <code>iota</code> .....	147
Указатели в Go .....	149
Зачем нужны указатели .....	152
Время и дата .....	152
Работа с временем .....	153

Синтаксический анализ времени.....	154
Работа с датами.....	156
Синтаксический анализ дат.....	156
Изменение формата даты и времени.....	157
Измерение времени выполнения программы.....	159
Измерение скорости работы сборщика мусора Go.....	160
Веб-ссылки и упражнения.....	161
Резюме.....	161
<b>Глава 4. Использование составных типов данных.....</b>	<b>163</b>
Составные типы данных.....	164
Структуры.....	164
Указатели на структуры.....	166
Ключевое слово new.....	168
Кортежи.....	169
Регулярные выражения и сопоставление с образцом.....	170
Немного теории.....	171
Простой пример.....	171
Более сложный пример.....	174
Проверка IPv4-адресов.....	176
Строки.....	180
Что такое руны.....	183
Пакет unicode.....	184
Пакет strings.....	185
Оператор switch.....	189
Вычисление числа $\pi$ с высокой точностью.....	192
Разработка на Go хранилища типа «ключ — значение».....	195
Go и формат JSON.....	200
Чтение данных из формата JSON.....	200
Сохранение данных в формате JSON.....	202
Использование функций Marshal() и Unmarshal().....	203
Синтаксический анализ данных в формате JSON.....	205

---

Go и XML.....	207
Чтение XML-файла .....	210
Настройка вывода данных в формате XML .....	211
Go и формат YAML .....	213
Дополнительные ресурсы .....	213
Упражнения.....	214
Резюме .....	214
<b>Глава 5. Как улучшить код Go с помощью структур данных.....</b>	<b>216</b>
О графах и узлах .....	217
Сложность алгоритма .....	217
Двоичные деревья в Go .....	218
Реализация двоичного дерева в Go.....	219
Преимущества двоичных деревьев.....	221
Пользовательские хеш-таблицы в Go.....	222
Реализация пользовательской хеш-таблицы в Go .....	222
Реализация функции поиска .....	225
Преимущества пользовательских хеш-таблиц .....	226
Связные списки в Go.....	226
Реализация связного списка в Go.....	227
Преимущества связных списков .....	231
Двусвязные списки в Go .....	231
Реализация двусвязного списка в Go .....	232
Преимущества двусвязных списков.....	235
Очереди в Go.....	236
Реализация очереди в Go.....	236
Стеки в Go .....	239
Реализация стека в Go.....	239
Пакет container .....	242
Использование пакета container/heap .....	242
Использование пакета container/list .....	245
Использование пакета container/ring.....	247

Генерация случайных чисел.....	248
Генерация случайных строк .....	251
Генерация безопасной последовательности случайных чисел.....	253
Выполнение матричных вычислений.....	255
Сложение и вычитание матриц .....	255
Умножение матриц .....	258
Деление матриц.....	261
Разгадывание головоломок sudoku.....	267
Дополнительные ресурсы .....	270
Упражнения.....	271
Резюме .....	272
<b>Глава 6. Неочевидные знания о пакетах и функциях Go.....</b>	<b>273</b>
Что такое Go-пакеты.....	274
Что такое функции Go .....	274
Анонимные функции .....	275
Функции, которые возвращают несколько значений .....	275
Функции, возвращающие именованные значения .....	277
Функции, принимающие указатели .....	279
Функции, которые возвращают указатели.....	280
Функции, которые возвращают другие функции .....	281
Функции, которые принимают другие функции в качестве параметров....	282
Функции с переменным числом параметров.....	283
Разработка Go-пакетов .....	285
Компиляция Go-пакета.....	287
Закрытые переменные и функции.....	287
Функция init().....	287
Go-модули .....	290
Создание и использование Go-модулей.....	290
Использование двух версий одного и того же Go-модуля.....	298
Где хранятся Go-модули .....	299
Команда go mod vendor .....	300

Как писать хорошие Go-пакеты.....	300
Пакет <code>syscall</code> .....	302
Как на самом деле работает <code>fmt.Println()</code> .....	304
Пакеты <code>go/scanner</code> , <code>go/parser</code> и <code>go/token</code> .....	306
Пакет <code>go/ast</code> .....	307
Пакет <code>go/scanner</code> .....	307
Пакет <code>go/parser</code> .....	309
Практический пример .....	311
Поиск имен переменных заданной длины .....	313
Шаблоны для текста и HTML.....	318
Вывод простого текста.....	318
Вывод текста в формате HTML.....	320
Дополнительные ресурсы .....	327
Упражнения.....	328
Резюме .....	328
<b>Глава 7. Рефлексия и интерфейсы на все случаи жизни.....</b>	<b>330</b>
Методы типов .....	330
Интерфейсы в Go.....	332
Операции утверждения типа.....	333
Как писать свои интерфейсы .....	335
Использование интерфейса Go .....	336
Использование переключателей для интерфейсов и типов данных.....	338
Рефлексия .....	340
Простой пример рефлексии .....	340
Более сложный пример рефлексии .....	343
Три недостатка рефлексии.....	345
Библиотека <code>reflectwalk</code> .....	346
Объектно-ориентированное программирование на Go .....	348
Основы <code>git</code> и <code>GitHub</code> .....	351
Использование <code>git</code> .....	351



Отладка с помощью Delve.....	357
Пример отладки.....	358
Дополнительные ресурсы.....	362
Упражнения.....	362
Резюме.....	362
<b>Глава 8.</b> Как объяснить UNIX-системе, что она должна делать.....	364
О процессах в UNIX.....	365
Пакет flag.....	365
Пакет viper.....	370
Простой пример использования viper.....	371
От flag к viper.....	372
Чтение конфигурационных файлов в формате JSON.....	373
Чтение конфигурационных файлов в формате YAML.....	375
Пакет cobra.....	377
Простой пример cobra.....	378
Создание псевдонимов команд.....	382
Интерфейсы io.Reader и io.Writer.....	385
Буферизованный и небуферизованный ввод и вывод в файл.....	385
Пакет bufio.....	386
Чтение текстовых файлов.....	386
Построчное чтение текстового файла.....	386
Чтение текстового файла по словам.....	388
Посимвольное чтение текстового файла.....	390
Чтение из /dev/random.....	392
Чтение заданного количества данных.....	393
Преимущества двоичных форматов.....	395
Чтение CSV-файлов.....	396
Запись в файл.....	399
Загрузка и сохранение данных на диске.....	401
И снова пакет strings.....	404

Пакет bytes.....	406
Полномочия доступа к файлам .....	407
Обработка сигналов в UNIX .....	408
Обработка двух сигналов.....	409
Обработка всех сигналов .....	411
Программирование UNIX-каналов на Go.....	413
Реализация утилиты cat(1) на Go .....	414
Структура syscall.PtraceRegs.....	416
Отслеживание системных вызовов.....	418
Идентификаторы пользователя и группы.....	422
Docker API и Go.....	423
Дополнительные ресурсы .....	426
Упражнения.....	427
Резюме .....	428
<b>Глава 9. Конкурентность в Go: горутин, каналы и конвейеры .....</b>	<b>429</b>
О процессах, потоках и горутин .....	430
Планировщик Go.....	430
Конкурентность и параллелизм.....	431
Горутин .....	431
Создание горутин .....	432
Создание нескольких горутин .....	433
Как дождаться завершения горутин, прежде чем закончить программу .....	435
Что происходит, если количество вызовов Add() и Done() не совпадает.....	437
Каналы .....	439
Запись в канал.....	439
Чтение из канала.....	440
Прием данных из закрытого канала .....	442
Каналы как аргументы функции.....	443
Конвейеры.....	444

Состояние гонки .....	447
Сравнение моделей конкурентности в Go и Rust .....	449
Сравнение моделей конкурентности в Go и Erlang .....	449
Дополнительные ресурсы .....	450
Упражнения .....	450
Резюме .....	451
<b>Глава 10.</b> Конкурентность в Go: расширенные возможности .....	452
И снова о планировщике Go .....	453
Переменная среды GOMAXPROCS .....	455
Ключевое слово select .....	456
Принудительное завершение горутины .....	459
Принудительное завершение горутины, способ 1 .....	459
Принудительное завершение горутины, способ 2 .....	461
И снова о Go-каналах .....	463
Сигнальные каналы .....	464
Буферизованные каналы .....	464
Нулевые каналы .....	466
Каналы каналов .....	467
Выбор последовательности исполнения горутин .....	470
Как не надо использовать горутины .....	472
Общая память и общие переменные .....	473
Тип sync.Mutex .....	474
Тип sync.RWMutex .....	478
Пакет atomic .....	481
Совместное использование памяти с помощью горутин .....	483
И снова об операторе go .....	485
Распознавание состояния гонки .....	488
Пакет context .....	493
Расширенный пример использования пакета context .....	497
Еще один пример использования пакета context .....	502
Пулы обработчиков .....	503

Дополнительные ресурсы .....	508
Упражнения .....	508
Резюме .....	509
<b>Глава 11. Тестирование, оптимизация и профилирование кода</b> .....	510
Оптимизация .....	511
Оптимизация кода Go .....	512
Профилирование кода Go .....	513
Стандартный Go-пакет net/http/pprof .....	513
Простой пример профилирования .....	513
Удобный внешний пакет для профилирования .....	521
Веб-интерфейс Go-профилировщика .....	523
Утилита go tool trace .....	527
Тестирование кода Go .....	532
Написание тестов для существующего кода Go .....	532
Тестовое покрытие кода .....	536
Тестирование HTTP-сервера с базой данных .....	539
Пакет testing/quick .....	545
Бенчмаркинг кода Go .....	551
Простой пример бенчмаркинга .....	552
Неправильно определенные функции бенчмаркинга .....	557
Бенчмаркинг буферизованной записи .....	558
Обнаружение недоступного кода Go .....	562
Кросс-компиляция .....	564
Создание примеров функций .....	565
От кода Go до машинного кода .....	567
Использование ассемблера в Go .....	568
Генерация документации .....	570
Использование образов Docker .....	575
Дополнительные ресурсы .....	577
Упражнения .....	578
Резюме .....	579

<b>Глава 12. Основы сетевого программирования на Go</b> .....	580
Что такое net/http, net и http.RoundTripper .....	581
Тип http.Response .....	581
Тип http.Request .....	582
Тип http.Transport .....	582
Что такое TCP/IP .....	583
Что такое IPv4 и IPv6.....	584
Утилита командной строки nc(1).....	584
Чтение конфигурации сетевых интерфейсов.....	585
Выполнение DNS-поиска .....	589
Получение NS-записей домена.....	591
Получение MX-записей домена .....	593
Создание веб-сервера на Go .....	594
Использование пакета atomic.....	597
Профилирование HTTP-сервера .....	599
Создание веб-сайта на Go.....	604
HTTP-трассировка .....	613
Тестирование HTTP-обработчиков .....	616
Создание веб-клиента на Go .....	618
Как усовершенствовать наш веб-клиент Go .....	620
Задержки HTTP-соединений.....	623
Подробнее о SetDeadline.....	625
Установка периода ожидания на стороне сервера .....	625
Еще один способ определить период ожидания .....	627
Инструменты Wireshark и tshark .....	629
Go и gRPC.....	629
Определение файла описания интерфейса .....	629
gRPC-клиент .....	632
gRPC-сервер .....	633
Дополнительные ресурсы .....	635
Упражнения.....	636
Резюме .....	637

<b>Глава 13. Сетевое программирование: создание серверов и клиентов</b> .....	638
Работа с HTTPS-трафиком .....	639
Создание сертификатов .....	639
HTTPS-клиент .....	640
Простой HTTPS-сервер .....	642
Разработка TLS-сервера и TLS-клиента .....	643
Стандартный Go-пакет net .....	646
TCP-клиент .....	646
Другая версия TCP-клиента .....	648
TCP-сервер .....	650
Другая версия TCP-сервера .....	652
UDP-клиент .....	654
Разработка UDP-сервера .....	656
Конкурентный TCP-сервер .....	658
Удобный конкурентный TCP-сервер .....	662
Создание образа Docker для TCP/IP-сервера на Go .....	668
Дистанционный вызов процедур .....	670
RPC-клиент .....	671
RPC-сервер .....	672
Низкоуровневое сетевое программирование .....	674
Получение необработанных сетевых данных ICMP .....	676
Дополнительные ресурсы .....	680
Упражнения .....	681
Резюме .....	682
<b>Глава 14. Машинное обучение на Go</b> .....	683
Вычисление простых статистических показателей .....	684
Регрессия .....	688
Линейная регрессия .....	688
Реализация линейной регрессии .....	688
Вывод данных .....	690
Классификация .....	694

Кластеризация.....	698
Выявление аномалий.....	700
Нейронные сети.....	702
Анализ выбросов.....	704
Работа с TensorFlow.....	707
Поговорим о Kafka.....	712
Дополнительные ресурсы.....	716
Упражнения.....	717
Резюме.....	717
Что дальше?.....	718