

O'REILLY®

Go

ИДИОМЫ И ПАТТЕРНЫ
ПРОЕКТИРОВАНИЯ



Джон Боднер

FIRST EDITION

Learning Go

*An Idiomatic Approach to
Real-World Go Programming*

Jon Bodner

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Go

идиомы и паттерны проектирования

Джон Боднер



Санкт-Петербург • Москва • Минск

2022

ББК 32.988.02-018.1
УДК 004.738.5
Б75

Боднер Джон

Б75 Go: идиомы и паттерны проектирования. — СПб.: Питер, 2022. — 416 с.: ил. — (Серия «Бестселлеры O'Reilly»).

ISBN 978-5-4461-1885-4

Go быстро набирает популярность в качестве языка для создания веб-сервисов. Существует множество учебников по синтаксису Go, но знать его недостаточно. Автор Джон Боднер описывает и объясняет паттерны проектирования, используемые опытными разработчиками. В книге собрана наиболее важная информация, необходимая для написания чистого и идиоматического Go-кода. Вы научитесь думать как Go-разработчик, вне зависимости от предыдущего опыта программирования.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.988.02-018.1
УДК 004.738.5

Права на издание получены по соглашению с O'Reilly. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-1492077213 англ.

Authorized Russian translation of the English edition of Learning Go

ISBN 9781492077213 © 2021 Jon Bodner

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

ISBN 978-5-4461-1885-4

© Перевод на русский язык ООО «Прогресс книга», 2022

© Издание на русском языке, оформление ООО «Прогресс книга», 2022

© Серия «Бестселлеры O'Reilly», 2022

Оглавление

Предисловие	13
Для кого написана книга	14
Условные обозначения.....	14
Использование примеров программного кода.....	15
Благодарности.....	16
От издательства	17
Глава 1. Настройка среды разработки для языка Go	18
Установка средств разработки для языка Go.....	18
Рабочее пространство Go	19
Команда go.....	21
Команды go run и go build	21
Установка сторонних инструментов для Go.....	23
Форматирование кода	24
Выбор инструментов.....	26
Visual Studio Code.....	26
GoLand	28
Онлайн-песочница	29
make-файлы.....	31
Обновление средств разработки	33
Резюме	34
Глава 2. Простые типы и объявление переменных	35
Встроенные типы	35
Нулевое значение.....	35
Литералы.....	36
Логические значения	38
Числовые типы.....	38
Пробуем использовать строки и руны	46
Явное преобразование типов.....	46
var или :=	48
Использование ключевого слова const.....	51

Типизированные и нетипизированные константы	52
Неиспользуемые переменные.....	53
Именование переменных и констант	54
Резюме	56

Глава 3. Составные типы.....57

Массивы — слишком строгие для того, чтобы использовать их напрямую.....	57
Срезы	59
Функция len	61
Функция append.....	61
Емкость среза	62
Функция make.....	64
Объявление собственного среза.....	65
Срезание срезов	67
Преобразование массивов в срезы	70
Функция сору.....	70
Строки в сочетании с рунами и байтами	72
Карты	76
Чтение и запись карты.....	79
Идиома «запятая-ok».....	79
Удаление из карты	80
Использование карты в качестве множества	81
Структуры	82
Анонимные структуры	85
Сравнение и преобразование структур	86
Резюме	87

Глава 4. Блоки, затенение переменных и управляющие конструкции88

Блоки.....	88
Затенение переменных.....	89
Выявление затененных переменных.....	91
Оператор if	93
Четыре вида оператора for.....	94
Полный оператор for.....	95
Оператор for, использующий только условие	96
Бесконечный оператор for	96
Ключевые слова break и continue	97
Оператор for-range.....	99
Операторы for с метками.....	104
Выбор подходящего оператора for	105
Оператор switch.....	107
Пустые переключатели	110

Что лучше выбрать: if или switch?.....	111
Оператор goto.....	112
Резюме	115
Глава 5. Функции.....	116
Объявление и вызов функций	116
Имитация именованных и опциональных параметров	117
Вариативные входные параметры и срезы	118
Возврат нескольких значений	119
При возврате нескольких значений всегда возвращается несколько значений	120
Игнорирование возвращаемых значений	121
Именованные возвращаемые значения	122
Никогда не используйте пустые операторы возврата!.....	123
Функции являются значениями.....	125
Объявление функциональных типов.....	127
Анонимные функции.....	127
Замыкания.....	128
Передача функций в качестве параметров	129
Возвращение функций из функций	131
Оператор defer.....	132
Go — язык с передачей параметров по значению.....	136
Резюме	139
Глава 6. Указатели.....	140
Общие сведения об указателях	140
Не бойтесь указателей.....	144
Указатели служат для указания изменяемых параметров.....	146
Указатели — это крайняя мера.....	151
Влияние передачи указателей на производительность.....	152
Различие между нулевым значением и отсутствием значения.....	153
Различие между картами и срезами	154
Использование срезов в качестве буферов.....	157
Уменьшение нагрузки на сборщик мусора.....	159
Резюме	163
Глава 7. Типы, методы и интерфейсы.....	164
Типы в Go.....	164
Методы.....	165
Приемники указателей и приемники значений	166
Пишите код методов с расчетом на экземпляр, равный nil	168
Методы тоже являются функциями	170
Функции или методы?.....	171

Объявление типа не является наследованием.....	172
Типы являются исполняемой документацией.....	173
Йота (иногда) используется для создания перечислений	173
Используйте встраивание для реализации композиции	176
Встраивание не является наследованием.....	177
Общее представление об интерфейсах.....	179
Интерфейсы обеспечивают типобезопасную утиную типизацию	179
Встраивание и интерфейсы.....	184
Принимайте интерфейсы, возвращайте структуры.....	184
Интерфейсы и значение nil.....	186
Пустой интерфейс ничего не сообщает.....	187
Утверждения типа и переключатели типа.....	188
Используйте утверждения типа и переключатели типа как можно реже	191
Функциональные типы — ключ к интерфейсам.....	194
Неявные интерфейсы облегчают внедрение зависимостей	195
Утилита Wire.....	200
Go нельзя назвать объектно-ориентированным языком (и это здорово!)	200
Резюме	200
Глава 8. Ошибки	201
Как обрабатывать ошибки: основы	201
Используйте строки в случае простых ошибок	203
Сигнальные ошибки	204
Ошибки являются значениями	206
Обертывание ошибок	209
Функции Is и As.....	212
Обертывание ошибок с помощью оператора defer.....	215
Функции panic и recover.....	216
Извлечение трассировки стека из ошибки.....	219
Резюме	219
Глава 9. Модули, пакеты и операции импорта	220
Репозитории, модули и пакеты.....	220
Файл go.mod.....	221
Компиляция пакетов	222
Операции импорта и экспорта	222
Создание и использование пакета	222
Именованые пакеты.....	225
Как следует подходить к организации кода модуля	226
Переопределение имени пакета.....	227
Комментарии пакета и godoc.....	228
Пакет internal	230

По возможности не используйте функцию <code>init</code>	231
Циклические зависимости	232
Переименование и реорганизация API без потери работоспособности	233
Работа с модулями.....	235
Импортирование стороннего кода	235
Работа с версиями	238
Выбор минимальной версии	241
Обновление до совместимых версий	241
Обновление до несовместимых версий	242
Вендоринг	244
Сайт <code>pkg.go.dev</code>	245
Дополнительная информация.....	245
Публикация своего модуля	245
Версионирование своего модуля	246
Прокси-серверы модулей.....	248
Указание прокси-сервера	248
Закрытые репозитории.....	249
Резюме	250
Глава 10. Конкурентность в Go	251
Когда следует использовать конкурентность.....	251
Горутины	253
Каналы.....	255
Чтение, запись и буферизация	255
Цикл <code>for-range</code> и каналы.....	257
Закрытие канала.....	257
Различия в поведении каналов.....	258
Оператор <code>select</code>	259
Принципы и паттерны конкурентного программирования	262
Следите за тем, чтобы конкурентности не было в ваших API.....	263
Горутины, циклы <code>for</code> и изменяющиеся переменные	263
Всегда закрывайте горутины	264
Паттерн на основе канала <code>done</code>	266
Прекращение выполнения горутины с помощью функции отмены.....	267
Когда следует использовать буферизованные и небуферизованные каналы.....	268
Противодавление	269
Отключение ветвей оператора <code>select</code>	270
Как можно установить тайм-аут для кода.....	271
Использование типа <code>WaitGroup</code>	272
Однократное выполнение кода	274

Собираем инструменты для обеспечения конкурентности	275
Когда вместо каналов следует использовать мьютексы.....	279
Атомарные операции — скорее всего, они вам не понадобятся.....	284
Где можно найти более подробную информацию о конкурентности.....	284
Резюме	285
Глава 11. Стандартная библиотека	286
Пакет io и его друзья	286
Пакет time	292
Монотонное время	294
Таймеры и тайм-ауты.....	295
Пакет encoding/json	295
Используйте теги структур для добавления метаданных.....	295
Демаршализация и маршализация	297
JSON, считыватели и записыватели	298
Кодирование и декодирование JSON-потокoв.....	299
Парсинг пользовательского формата JSON	301
Пакет net/http	303
Клиент	303
Сервер.....	304
Промежуточный слой	307
Резюме	310
Глава 12. Контекст.....	311
Что такое контекст.....	311
Отмена.....	314
Таймеры.....	318
Управление отменой контекста в собственном коде.....	321
Значения	322
Резюме	329
Глава 13. Написание тестов	330
Основы тестирования.....	330
Выдача сообщения о неудачном завершении теста	332
Подготовка и заключительная «уборка».....	333
Расположение образцов тестовых данных.....	335
Кэширование результатов теста	335
Тестирование своего публичного API	335
Используйте модуль go-cmp для сравнения результатов тестов	337
Табличные тесты.....	339
Проверка степени покрытия кода	341
Сравнительные тесты.....	343

Заглушки в Go	348
Пакет httptest	354
Интеграционные тесты и теги сборки	356
Выявление проблем конкурентности с помощью детектора состояний гонки	358
Резюме	360
Глава 14. Здесь водятся драконы: пакеты reflect, unsafe и sgo	361
Рефлексия позволяет нам работать с типами на этапе выполнения	362
Типы, разновидности и значения	364
Создание новых значений	369
Используйте рефлексю для проверки значения интерфейса на равенство значению nil	370
Используйте рефлексю для создания маршализатора данных	371
Создавайте с помощью рефлексии функции для автоматизации повторяющихся задач	376
Рефлексю можно использовать для создания структур, но лучше этого не делать	378
Рефлексия не позволяет создавать методы	378
Используйте рефлексю только тогда, когда в этом есть смысл	378
Использовать пакет unsafe небезопасно	380
Используйте пакет unsafe для преобразования внешних двоичных данных	381
Пакет unsafe в сочетании со строками и срезами	385
Вспомогательные инструменты для пакета unsafe	386
Пакет sgo предназначен для обеспечения интеграции, а не повышения производительности	387
Резюме	391
Глава 15. Взгляд в будущее: обобщенные типы в Go	392
Обобщенные типы уменьшают количество повторяющегося кода и повышают типобезопасность	392
Добавление обобщенных типов в Go	395
Используйте списки типов для определения операторов	401
Обобщенные функции абстрагируют алгоритмы	402
Списки типов накладывают ограничения на константы и реализации	403
Что остается «за бортом»	407
Идиоматический Go-код и обобщенные типы	409
Какие нововведения нас ожидают	410
Резюме	411
Об авторе	412
Об обложке	413

Go — уникальный язык, и даже опытным программистам приходится «забыть» о том, как делаются некоторые вещи, и научиться иному подходу к программному обеспечению. Автор хорошо объясняет основные возможности языка, сопровождая теорию примерами идиоматического кода и описанием подводных камней и шаблонов проектирования.

Аарон Шлезингер (Aaron Schlesinger), ведущий разработчик, Microsoft

Джон на протяжении многих лет пользуется большим авторитетом в Go-сообществе; его высказывания и статьи часто несут в себе много полезного. Эта книга представляет собой руководство по изучению языка Go, рассчитанное на программистов. Она отлично сбалансирована в том плане, что хорошо объясняет необходимые вещи без пересказывания общеизвестных концепций.

Стив Франция (Steve Francia), создатель Hugo, Cobra и Viper, ведущий менеджер по Go-продуктам, Google

Боднер предлагает нам настоящий Go. В ясной, «живой» манере он учит этому языку от самых его основ до таких продвинутых тем, как рефлексия и взаимодействие с кодом, написанным на языке C. Многочисленные примеры показывают, как следует писать *идиоматический* Go-код, уделяя особое внимание ясности и простоте. Также подробно объясняется, как те или иные базовые концепции влияют на поведение программ, как, например, указатели влияют на распределение памяти и сборку мусора. Эта книга поможет новичкам очень быстро выйти на должный уровень, и даже опытные Go-программисты найдут в ней что-то полезное.

Джонатан Амстердам (Jonathan Amsterdam), сотрудник подразделения Go-разработки, Google

Книга содержит важную вводную информацию о том, что именно делает уникальным язык программирования Go, а также о паттернах проектирования и идиомах, которые делают его чрезвычайно мощным. Джону Боднеру удалось показать связь между базовыми возможностями Go и его философией, направляя читателей писать Go-код так, как это было задумано создателями языка.

Роберт Лебовиц (Robert Liebowitz), разработчик, Morning Consult

Джон написал не просто руководство по Go; издание дает идиоматически и практически ориентированное понимание этого языка. Книга написана на основе богатого опыта Джона в компьютерной отрасли и поможет тем, кто хочет очень быстро научиться эффективно пользоваться этим языком.

Уильям Кеннеди (William Kennedy), управляющий партнер, Ardan Labs

Предисловие

Изначально я хотел назвать эту книгу «Скучный Go», потому что Go — это и правда очень скучно.

Конечно, немного странно писать книгу на скучную тему, но я сейчас все объясню. Язык Go обладает небольшим набором возможностей, что идет вразрез с большинством других современных языков программирования. Хорошо написанная программа на Go, как правило, отличается простотой, а часто и некоторым однообразием. Здесь нет наследования, обобщенных типов (по крайней мере пока), аспектно-ориентированного программирования, перегрузки функций и уж точно нет перегрузки операторов. Здесь нет сравнения с паттерном, нет именованных параметров и исключений, зато есть пугающие многих *указатели*. Модель обмена данными в Go не похожа на ту, что применяется в других языках, и основана на идеях 1970-х годов, как и алгоритм, используемый для сборщика мусора. Другими словами, работа с Go выглядит откатом в прошлое.

Однако если Go скучен, это еще не значит, что он *примитивен*. Для правильного использования этого языка необходимо четко понимать, как его возможности сочетаются друг с другом. Хотя вы вполне можете написать на Go такой код, который будет выглядеть почти так же, как код на Java или Python, но результат вас не слишком обрадует, и вы так и не поймете, к чему была вся эта суета. Вот тут-то и пригодится эта книга. Она познакомит вас с возможностями Go и объяснит, как лучше их использовать для написания расширяемого идиоматического кода.

Когда речь идет о создании чего-то, что должно служить нам долгие годы, такое качество, как «обычность», становится не минусом, а плюсом. Никто не захочет первым прокатиться на своей машине по мосту, созданному с использованием непроверенных технологий, которые инженер посчитал «крутыми». Современный мир зависит от программного обеспечения так же, как и от мостов, если не больше. Тем не менее во многие языки программирования добавляют массу новых компонентов, не задумываясь об их влиянии на «ремонтпригодность» кодовой базы, в то время как Go рассчитан на создание надежных программ, которые смогут улучшать десятки программистов на протяжении десятилетий.

Так что да, Go скучный и это здорово! Я надеюсь, книга научит вас создавать с помощью такого «скучного» кода потрясающие проекты.

Для кого написана книга

Целевая аудитория — разработчики, желающие изучить второй (или пятый) язык программирования. Она подойдет как новичкам, которые знают лишь то, что у Go есть забавный талисман, так и тем, кто уже успел прочесть краткое руководство по этому языку или даже написать на нем несколько фрагментов кода. Эта книга не просто рассказывает, как следует писать программы на Go, она показывает, как можно делать это *идиоматически*. Более опытные Go-разработчики найдут здесь советы по использованию новых возможностей языка.

Предполагается, что читатель уже умеет пользоваться таким инструментом разработчика, как система контроля версий (желательно Git) и IDE. Читатель должен быть знаком с такими базовыми понятиями информатики, как конкурентность и абстракция, поскольку в книге мы будем касаться этих тем применительно к Go. Одни примеры кода можно скачать с GitHub, а десятки других — запустить в онлайн-песочнице *The Go Playground*. Наличие соединения с интернетом необязательно, однако с ним будет проще изучать исполняемые примеры кода. Поскольку язык Go часто используется для создания и вызова HTTP-серверов, для понимания некоторых примеров читатель должен иметь общее представление о протоколе HTTP и связанных с ним концепциях.

Несмотря на то что большинство функций Go встречаются и в других языках, программы, написанные на Go, обладают иной структурой. В книге мы начнем с настройки среды разработки для языка Go, после чего поговорим о переменных, типах, управляющих конструкциях и функциях. Если вам захочется пропустить этот материал, постарайтесь удержаться от этого и все-таки ознакомиться с ним. Зачастую именно такие детали делают Go-код идиоматическим. Простые на первый взгляд вещи могут оказаться очень интересными, если вы присмотритесь к ним получше.

Условные обозначения

В этой книге используются следующие типографские обозначения.

Курсивный шрифт

Используется для выделения новых терминов.

Шрифт без засечек

Применяется для выделения URL-адресов, адресов электронной почты.

Моноширинный шрифт

Используется для записи примеров программ, а также для выделения в тексте таких элементов, как имена переменных и функций, базы данных, типы данных, переменные среды, операторы и ключевые слова, имена и расширения файлов.

Полужирный моноширинный шрифт

Применяется для выделения команд и другого текста, который должен вводиться пользователем без каких-либо изменений.

Курсивный моноширинный шрифт

Используется для обозначения в коде элементов, вместо которых следует подставить предоставленные пользователем значения или значения, зависящие от контекста.



Так обозначается совет или предложение.



Так обозначается примечание общего характера.



Так обозначается предупреждение.

Использование примеров программного кода

Вспомогательные материалы (примеры программного кода, упражнения и т. д.) доступны для скачивания по адресу <https://github.com/learning-go-book>.

При возникновении технических вопросов или проблем, связанных с использованием примеров кода, пожалуйста, обращайтесь по адресу электронной почты bookquestions@oreilly.com.

Книга призвана помочь вам в решении ваших задач. В большинстве случаев вы можете свободно использовать приводимые в этой книге примеры кода в своих программах и документации. Вам не нужно обращаться в издательство

за разрешением, если вы не собираетесь воспроизводить существенную часть программного кода. Например, если вы разрабатываете программу и используете в ней несколько фрагментов кода из книги, вам не нужно обращаться за разрешением. Однако в случае продажи или распространения примеров из этой книги вы должны получить разрешение от издательства O'Reilly. Если вы отвечаете на вопросы, цитируя данную книгу или примеры из нее, то для этого не требуется разрешение. Но при включении значительного объема программного кода из этой книги в документацию вашего продукта необходимо получить разрешение издательства.

Мы не требуем, чтобы вы давали ссылку на первоисточник при цитировании, но будем признательны, если вы будете это делать. При этом обычно указывается название книги, имя ее автора, название издательства и номер ISBN. Например: «Go: идиомы и паттерны проектирования», Джон Боднер (Питер). Copyright 2021 Jon Bodner, 978-5-4461-1885-4».

Если у вас возникнут какие-либо сомнения относительно правомерности использования примеров кода, вы всегда можете связаться с нами по адресу permissions@oreilly.com.

Благодарности

Хоть и считается, что писательство — это индивидуальное занятие, книга не может появиться на свет без помощи множества других людей. Однажды я сказал Кармен Андо (Carmen Andoh), что собираюсь написать книгу по языку Go, и на конференции GopherCon 2019 она познакомила меня с Зан Маккуэйд (Zan McQuade) из компании O'Reilly. Зан помогла мне заключить договор с издательством, после чего регулярно консультировала меня по мере моего прогресса в написании книги. Мишель Кронин (Michele Cronin) редактировала мои тексты, высказывала замечания и выслушивала меня, когда я неизбежно сталкивался с трудностями. Технический редактор Тоня Трибула (Tonya Trubula) и литературный редактор Бет Келли (Beth Kelly) довели мою рукопись до пригодного для печати вида.

По мере написания книги я получал важные замечания (и слова поддержки) от многих людей, в числе которых были Джонатан Алтман (Jonathan Altman), Джонатан Амстердам (Jonathan Amsterdam), Джонни Рэй Остин (Johnny Ray Austin), Крис Фойербах (Chris Fauerbach), Крис Хайнс (Chris Hines), Билл Кеннеди (Bill Kennedy), Тони Нельсон (Tony Nelson), Фил Перл (Phil Pearl), Лиз Райс (Liz Rice), Аарон Шлезингер (Aaron Schlesinger), Крис Стаут (Chris Stout), Капил Тхангавелу (Kapil Thangavelu), Клер Тривисонно (Claire Trivisonno), Фолькер Уриг (Volker Uhrig), Джефф Уэндлинг (Jeff Wendling) и Крис Сара-

госа (Kris Zaragoza). Особых слов признательности заслуживает Роб Лебовиц (Rob Liebowitz), подробные замечания и быстрые ответы которого сделали эту книгу гораздо лучше.

Моей семье приходилось мириться с тем, что я проводил вечера и выходные дни за компьютером вместо того, чтобы проводить это время с ними. Моя жена Лора великодушно делала вид, что я не разбудил ее, когда я добирался до кровати в час ночи или еще позже.

Наконец, хочется вспомнить о тех людях, которые направили меня по этому пути четыре десятилетия назад. Первым из них был отец моего друга детства Пол Голдштейн (Paul Goldstein). В 1982 году он показал нам компьютер Commodore PET, ввел команду `PRINT 2 + 2` и нажал клавишу ввода. Я был поражен, когда на экране появилась цифра 4, и заболел этим раз и навсегда. Спустя какое-то время Пол научил меня программировать и даже на несколько недель одолжил свой компьютер. А вторым человеком была моя мама, которую я должен поблагодарить за то, что она поощряла мой интерес к программированию и компьютерам, даже толком не понимая, что это такое. В свое время она купила мне картридж для игровой приставки Atari 2600, позволявший писать программы на языке BASIC, компьютеры Commodore VIC-20 и Commodore 64, а также книги по программированию, после прочтения которых у меня появилась мечта когда-нибудь написать свою собственную книгу.

Спасибо всем вам за то, что помогли мне сделать эту мечту реальностью!

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.

Согласно комментарию автора на странице книги на сайте O'Reilly, инструмент `golint` не поддерживается. По этой причине мы удалили из главы 1 подраздел, посвященный `golint`.