

# 100 ошибок Go и как их избежать

Тейва Харшани



# *100 Go Mistakes*

AND HOW TO AVOID THEM

TEIVA HARSANYI



MANNING  
SHELTER ISLAND

# *100 ошибок Go*

И КАК ИХ ИЗБЕЖАТЬ

ТЕЙВА ХАРШАНИ



Санкт-Петербург • Москва • Минск

2024

ББК 32.973.2-018.1  
УДК 004.43  
Х22

### **Харшани Тейва**

Х22 100 ошибок Go и как их избежать. — СПб.: Питер, 2024. — 480 с.: ил. — (Серия «Для профессионалов»).

ISBN 978-5-4461-2058-1

Лучший способ улучшить код — понять и исправить ошибки, сделанные при его написании. В этой уникальной книге проанализированы 100 типичных ошибок и неэффективных приемов в Go-приложениях.

Вы научитесь писать идиоматичный и выразительный код на Go, разберете десятки интересных примеров и сценариев и поймете, как обнаружить ошибки и потенциальные ошибки в своих приложениях. Чтобы вам было удобнее работать с книгой, автор разделил методы предотвращения ошибок на несколько категорий, начиная от типов данных и работы со строками и заканчивая конкурентным программированием и тестированием.

Для опытных Go-разработчиков, хорошо знакомых с синтаксисом языка.

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

ББК 32.973.2-018.1  
УДК 004.43

Права на издание получены по соглашению с Manning Publications. 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. В книге возможны упоминания организаций, деятельность которых запрещена на территории Российской Федерации, таких как Meta Platforms Inc., Facebook, Instagram и др. 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 978-1617299599 англ.  
ISBN 978-5-4461-2058-1

© 2022 Manning Publications  
© Перевод на русский язык ООО «Прогресс книга», 2023  
© Издание на русском языке, оформление ООО «Прогресс книга», 2023

# *Краткое содержание*

---

Предисловие .....	18
Благодарности .....	20
Об этой книге .....	22
Об авторе .....	25
Иллюстрация на обложке .....	26
От издательства .....	27
<b>Глава 1.</b> Go: просто научиться, но сложно освоить .....	28
<b>Глава 2.</b> Организация кода и проекта .....	36
<b>Глава 3.</b> Типы данных .....	98
<b>Глава 4.</b> Управляющие структуры .....	148
<b>Глава 5.</b> Строки .....	171
<b>Глава 6.</b> Функции и методы .....	187

**6**    Краткое содержание

<b>Глава 7.</b> Обработка ошибок.....	208
<b>Глава 8.</b> Конкурентность: основы.....	231
<b>Глава 9.</b> Конкурентность: практика.....	270
<b>Глава 10.</b> Стандартная библиотека.....	321
<b>Глава 11.</b> Тестирование .....	355
<b>Глава 12.</b> Оптимизация .....	402

# Оглавление

---

<b>Предисловие .....</b>	<b>18</b>
<b>Благодарности .....</b>	<b>20</b>
<b>Об этой книге.....</b>	<b>22</b>
Для кого эта книга .....	22
Структура книги.....	22
О коде в книге.....	23
Форум liveBook.....	24
<b>Об авторе .....</b>	<b>25</b>
<b>Иллюстрация на обложке.....</b>	<b>26</b>
<b>От издательства .....</b>	<b>27</b>
<b>Глава 1. Go: просто научиться, но сложно освоить .....</b>	<b>28</b>
1.1. Go: основные моменты .....	29
1.2. Просто не означает легко.....	30

## 8 Оглавление

1.3. 100 ошибок в Go .....	31
1.3.1. Баги .....	32
1.3.2. Излишняя сложность .....	33
1.3.3. Плохая читаемость .....	33
1.3.4. Неоптимальная или неидиоматическая организация.....	34
1.3.5. Отсутствие удобства в API .....	34
1.3.6. Неоптимизированный код .....	34
1.3.7. Недостаточная производительность .....	35
Итоги.....	35
<b>Глава 2. Организация кода и проекта .....</b>	<b>36</b>
2.1. <b>Ошибка #1:</b> непреднамеренно затенять переменные.....	36
2.2. <b>Ошибка #2:</b> лишний вложенный код .....	39
2.3. <b>Ошибка #3:</b> неправильно использовать функцию инициализации .....	42
2.3.1. Концепция .....	42
2.3.2. Когда использовать функции init.....	46
2.4. <b>Ошибка #4:</b> злоупотреблять геттерами и сеттерами.....	49
2.5. <b>Ошибка #5:</b> загрязнять интерфейсы .....	50
2.5.1. Концепции .....	50
2.5.2. Когда использовать интерфейсы .....	53
2.5.3. Загрязнение интерфейса .....	57
2.6. <b>Ошибка #6:</b> интерфейсы на стороне производителя .....	58
2.7. <b>Ошибка #7:</b> возврат интерфейсов.....	61
2.8. <b>Ошибка #8:</b> апу не говорит ни о чем.....	64
2.9. <b>Ошибка #9:</b> путаница в использовании дженериков.....	66
2.9.1. Концепция .....	67
2.9.2. Общие случаи использования и злоупотребления .....	71



2.10. <b>Ошибка #10:</b> не знать о возможных проблемах со встраиванием типов.....	73
2.11. <b>Ошибка #11:</b> не использовать паттерн функциональных опций.....	77
2.11.1. Структура Config.....	79
2.11.2. Паттерн Строитель.....	80
2.11.3. Паттерн функциональных опций.....	82
2.12. <b>Ошибка #12:</b> неорганизованность проекта.....	85
2.12.1. Структура проекта.....	85
2.12.2. Организация пакета.....	86
2.13. <b>Ошибка #13:</b> создавать пакеты утилит.....	88
2.14. <b>Ошибка #14:</b> игнорировать коллизии имен пакетов.....	90
2.15. <b>Ошибка #15:</b> не писать документацию по коду.....	91
2.16. <b>Ошибка #16:</b> не использовать линтеры.....	94
Итоги.....	95
<b>Глава 3. Типы данных.....</b>	<b>98</b>
3.1. <b>Ошибка #17:</b> путаница с восьмеричными литералами.....	98
3.2. <b>Ошибка #18:</b> игнорировать целочисленные переполнения.....	100
3.2.1. Концепции.....	100
3.2.2. Обнаружение целочисленного переполнения при инкрементировании.....	102
3.2.3. Обнаружение целочисленного переполнения при сложении.....	103
3.2.4. Обнаружение целочисленного переполнения при умножении.....	103
3.3. <b>Ошибка #19:</b> не понимать проблем, связанных с плавающей точкой.....	104
3.4. <b>Ошибка #20:</b> не понимать особенностей, связанных с длиной среза и его емкостью.....	109

## 10 Оглавление

3.5. <b>Ошибка #21:</b> неэффективная инициализация среза.....	114
3.6. <b>Ошибка #22:</b> путать пустые и нулевые срезы .....	118
3.7. <b>Ошибка #23:</b> неправильно проверять пустоту среза .....	122
3.8. <b>Ошибка #24:</b> неправильно создавать копии срезов.....	124
3.9. <b>Ошибка #25:</b> неожиданные побочные эффекты при использовании append в операциях со срезами.....	125
3.10. <b>Ошибка #26:</b> срезы и утечки памяти.....	129
3.10.1. Утечки емкости.....	129
3.10.2. Срез и указатели.....	131
3.11. <b>Ошибка #27:</b> неэффективно инициализировать карты.....	135
3.11.1. Концепции.....	135
3.11.2. Инициализация .....	137
3.12. <b>Ошибка #28:</b> карты и утечки памяти .....	139
3.13. <b>Ошибка #29:</b> некорректное сравнение значений .....	142
Итоги.....	146

## Глава 4. Управляющие структуры..... 148

4.1. <b>Ошибка #30:</b> игнорировать то, что элементы в цикле range копируются .....	148
4.1.1. Концепция .....	149
4.1.2. Копия значения.....	150
4.2. <b>Ошибка #31:</b> игнорировать то, как в циклах range вычисляются аргументы .....	152
4.2.1. Каналы .....	154
4.2.2. Массив.....	155
4.3. <b>Ошибка #32:</b> игнорировать влияние, которое оказывает использование элементов указателя в циклах range .....	157
4.4. <b>Ошибка #33:</b> делать неверные допущения во время итераций карты .....	161

4.4.1. Упорядочивание .....	161
4.4.2. Вставка карты во время итераций.....	163
4.5. <b>Ошибка #34:</b> игнорировать особенности работы оператора break.....	165
4.6. <b>Ошибка #35:</b> использовать defer внутри циклов.....	167
Итоги.....	169
<b>Глава 5. Строки .....</b>	<b>171</b>
5.1. <b>Ошибка #36:</b> не понимать концепции рун.....	172
5.2. <b>Ошибка #37:</b> неточная итерация строк.....	174
5.3. <b>Ошибка #38:</b> неправильно использовать функции обрезки .....	177
5.4. <b>Ошибка #39:</b> недостаточная степень оптимизации при конкатенации строк .....	179
5.5. <b>Ошибка #40:</b> бесполезные преобразования строк .....	182
5.6. <b>Ошибка #41:</b> подстроки и утечки памяти.....	183
Итоги.....	186
<b>Глава 6. Функции и методы .....</b>	<b>187</b>
6.1. <b>Ошибка #42:</b> не знать, какой тип получателя использовать .....	188
6.2. <b>Ошибка #43:</b> не использовать именованные параметры результата....	191
6.3. <b>Ошибка #44:</b> побочные эффекты от именованных параметров результата .....	194
6.4. <b>Ошибка #45:</b> возврат получателя nil .....	196
6.5. <b>Ошибка #46:</b> использовать имя файла в качестве входных данных функции.....	200
6.6. <b>Ошибка #47:</b> игнорировать то, как вычисляются аргументы и получатели оператора defer .....	202
6.6.1. Вычисление аргументов .....	203
6.6.2. Получатели значений или указателей.....	205
Итоги.....	207

<b>Глава 7. Обработка ошибок.....</b>	<b>208</b>
7.1. <b>Ошибка #48:</b> паника .....	209
7.2. <b>Ошибка #49:</b> игнорировать обрачивание ошибки.....	211
7.3. <b>Ошибка #50:</b> неточная проверка типа ошибки .....	215
7.4. <b>Ошибка #51:</b> неточная проверка значения ошибки .....	219
7.5. <b>Ошибка #52:</b> двойная обработка ошибки .....	221
7.6. <b>Ошибка #53:</b> не выполнять обработку ошибки .....	224
7.7. <b>Ошибка #54:</b> не выполнять обработку ошибки оператора defer .....	226
Итоги.....	229
<b>Глава 8. Конкурентность: основы .....</b>	<b>231</b>
8.1. <b>Ошибка #55:</b> путать конкурентность и параллелизм .....	232
8.2. <b>Ошибка #56:</b> полагать, что конкурентность быстрее.....	236
8.2.1. Планирование в Go.....	236
8.2.2. Параллельная сортировка слиянием .....	239
8.3. <b>Ошибка #57:</b> путаться в том, когда использовать каналы, а когда мьютексы.....	244
8.4. <b>Ошибка #58:</b> не понимать проблем гонки .....	246
8.4.1. Гонка данных и состояние гонки.....	246
8.4.2. Модель памяти Go.....	252
8.5. <b>Ошибка #59:</b> не понимать влияние типа рабочей нагрузки на конкурентность .....	255
8.6. <b>Ошибка #60:</b> неверно понимать контексты Go.....	261
8.6.1. Крайний срок.....	262
8.6.2. Сигналы отмены .....	263
8.6.3. Значения контекстов .....	264
8.6.4. Перехват отмены контекста.....	266
Итоги.....	268

<b>Глава 9. Конкурентность: практика .....</b>	<b>270</b>
9.1. <b>Ошибка #61:</b> передавать неподходящий контекст .....	270
9.2. <b>Ошибка #62:</b> запускать горутину и не знать, когда ее остановить.....	273
9.3. <b>Ошибка #63:</b> неосторожно обращаться с горутинами и переменными цикла .....	276
9.4. <b>Ошибка #64:</b> ожидать детерминированное поведение при использовании select и каналов.....	278
9.5. <b>Ошибка #65:</b> не использовать каналы уведомлений .....	283
9.6. <b>Ошибка #66:</b> не использовать нулевые каналы .....	285
9.7. <b>Ошибка #67:</b> гадать насчет размера канала.....	291
9.8. <b>Ошибка #68:</b> забывать о возможных побочных эффектах при форматировании строк.....	294
9.8.1. Гонка данных в etcd .....	295
9.8.2. Взаимоблокировка.....	296
9.9. <b>Ошибка #69:</b> создавать ситуацию гонки данных из-за оператора append.....	299
9.10. <b>Ошибка #70:</b> неверно использовать мьютексы со срезами и картами .....	301
9.11. <b>Ошибка #71:</b> неправильно использовать sync.WaitGroup.....	304
9.12. <b>Ошибка #72:</b> забывать о sync.Cond.....	307
9.13. <b>Ошибка #73:</b> не использовать errgroup .....	313
9.14. <b>Ошибка #74:</b> копировать тип sync .....	317
Итоги.....	319
<b>Глава 10. Стандартная библиотека .....</b>	<b>321</b>
10.1. <b>Ошибка #75:</b> неправильно задавать промежуток времени.....	322
10.2. <b>Ошибка #76:</b> time.After и утечки памяти .....	323
10.3. <b>Ошибка #77:</b> типичные ошибки при обработке JSON.....	326
10.3.1. Неожиданное поведение из-за встраивания типов.....	326

## 14 Оглавление

10.3.2. JSON и монотонные часы .....	329
10.3.3. Карта типа any.....	332
10.4. <b>Ошибка #78:</b> типичные ошибки, связанные с SQL.....	333
10.4.1. Не знать, что sql.Open не всегда устанавливает соединение с базой данных.....	333
10.4.2. Забывать о пуле соединений.....	334
10.4.3. Не использовать подготовленные операторы.....	336
10.4.4. Неправильная обработка нулевых значений .....	337
10.4.5. Не обрабатывать ошибки итерации строк .....	339
10.5. <b>Ошибка #79:</b> не закрывать временные ресурсы .....	340
10.5.1. Тело HTTP .....	340
10.5.2. sql.Rows.....	343
10.5.3. os.File.....	344
10.6. <b>Ошибка #80:</b> забывать об операторе return после ответа на HTTP-запрос .....	346
10.7. <b>Ошибка #81:</b> использовать стандартные HTTP-клиент и сервер.....	348
10.7.1. HTTP-клиент .....	348
10.7.2. HTTP-сервер .....	351
Итоги.....	353
<b>Глава 11. Тестирование .....</b>	<b>355</b>
11.1. <b>Ошибка #82:</b> не распределять тесты по категориям .....	356
11.1.1. Теги сборки.....	356
11.1.2. Переменные среды.....	358
11.1.3. Короткий режим.....	359
11.2. <b>Ошибка #83:</b> не включать флаг -race.....	360

11.3. <b>Ошибка #84:</b> не использовать режимы выполнения тестов .....	363
11.3.1. Флаг parallel.....	363
11.3.2. Флаг -shuffle.....	365
11.4. <b>Ошибка #85:</b> не использовать табличные тесты .....	366
11.5. <b>Ошибка #86:</b> задержки в юнит-тестах .....	371
11.6. <b>Ошибка #87:</b> неэффективная работа с API времени .....	374
11.7. <b>Ошибка #88:</b> не использовать пакеты утилит для тестирования.....	379
11.7.1. Пакет httptest .....	379
11.7.2. Пакет iotest.....	381
11.8. <b>Ошибка #89:</b> писать неточные бенчмарки .....	384
11.8.1. Не сбрасывать или не ставить на паузу таймер .....	385
11.8.2. Делать неверные предположения о микробенчмарках .....	386
11.8.3. Небрежное отношение к оптимизациям компилятора .....	389
11.8.4. Эффект наблюдателя.....	391
11.9. <b>Ошибка #90:</b> не изучать все возможности тестирования в Go .....	395
11.9.1. Покрытие тестами .....	395
11.9.2. Тестирование из другого пакета .....	396
11.9.3. Вспомогательные функции .....	397
11.9.4. Настройка и демонтаж.....	398
Итоги.....	399
<b>Глава 12. Оптимизация .....</b>	<b>402</b>
12.1. <b>Ошибка #91:</b> не понимать устройство кэша CPU .....	403
12.1.1. Архитектура CPU .....	403
12.1.2. Кэш-линия.....	405
12.1.3. Срез структур и структура срезов.....	408

## 16 Оглавление

12.1.4. Предсказуемость .....	410
12.1.5. Стратегия размещения кэша .....	412
12.2. <b>Ошибка #92:</b> писать конкурентный код, который приводит к ложному совместному использованию.....	418
12.3. <b>Ошибка #93:</b> не учитывать параллелизм на уровне инструкций.....	423
12.4. <b>Ошибка #94:</b> не знать о выравнивании данных.....	430
12.5. <b>Ошибка #95:</b> не понимать различий между стеком и кучей.....	435
12.5.1. Стек и куча .....	435
12.5.2. Эскейл-анализ.....	440
12.6. <b>Ошибка #96:</b> не знать, как сократить число выделений памяти .....	443
12.6.1. Изменения API.....	443
12.6.2. Приемы оптимизации компилятора .....	444
12.6.3. sync.Pool .....	445
12.7. <b>Ошибка #97:</b> не полагаться на встраивание .....	448
12.8. <b>Ошибка #98:</b> не использовать диагностический инструментарий Go.....	451
12.8.1. Профилирование .....	451
12.8.2. Трассировщик выполнения .....	460
12.9. <b>Ошибка #99:</b> не понимать, как работает сборщик мусора .....	465
12.9.1. Концепции.....	465
12.9.2. Примеры .....	467
12.10. <b>Ошибка #100:</b> не понимать особенностей запуска Go внутри Docker и Kubernetes.....	471
Итоги.....	474
Заключение .....	475



*Дэйву Харшани: продолжай оставаться тем, кто ты есть, братик.  
Твой потолок — звезды.*

*Милой Мелиссе.*

# Предисловие

---

В 2019 году я во второй раз начал профессионально заниматься работой на Go в качестве основного языка программирования. Тогда я заметил некоторые закономерности, связанные с ошибками написания кода на Go. Я подумал, что обобщение информации о таких частых ошибках было бы полезно для разработчиков.

В своем блоге я сделал пост «10 самых распространенных ошибок, с которыми я сталкивался в проектах на Go» («The Top 10 Most Common Mistakes I've Seen in Go Projects»). Пост стал популярным: его прочитали более 100 000 человек, он был выбран новостным бюллетенем Golang Weekly как один из лучших за 2019 год. Мне льстили положительные отзывы, которые я получал от сообщества Go.

Я понял, что обсуждение типичных ошибок — это мощный инструмент разработки. Сопровождаемый конкретными примерами, он поможет им эффективно осваивать новые навыки, облегчать запоминание как контекста, в котором эти ошибки встречаются, так и способов, позволяющих их избежать.

Около года я собирал примеры типичных ошибок: из профессиональных проектов других разработчиков, из репозитория опенсорсных программ, из книг, блогов, исследований и обсуждений в сообществе Go. Могу сказать, что я и сам был «достойным источником информации» в плане подобных ошибок.

К концу 2020 года размер моей коллекции ошибок достиг 100 штук, и это показалось мне подходящим, чтобы предложить идею публикации какому-либо издательству. В результате я связался с Manning, которое считал высококлассным

издательством, публиковавшим качественные книги, — для меня оно стало идеальным партнером. Потребовалось почти два года и бесчисленное количество итераций, чтобы четко сформулировать суть каждой из 100 ошибок вместе с релевантными примерами и несколькими решениями, где контекст — это ключевой фактор.

Очень надеюсь, что моя книга поможет вам избежать этих распространенных ошибок и улучшить владение языком Go.

# Благодарности

---

Хочу выразить свою признательность многим людям. Моим родителям — за то, что поддержали меня в тот момент, когда во время учебы я ощутил себя так, как будто нахожусь в ситуации полного провала. Моему дяде Жан-Полю Демону (Jean-Paul Demont) за то, что помог увидеть свет в конце туннеля. Пьеру Готье (Pierre Gautier) за то, что был замечательным вдохновителем и помог мне поверить в себя. Дэмиену Шамбону (Damien Chambon) за то, что заставлял меня постоянно поднимать планку и подталкивал меня к лучшему. Лорану Бернару (Laurent Bernard) за то, что был образцом для подражания и привел меня к осознанию того, что навыки социального общения очень важны. Валентину Делеплацу (Valentin Deleplace) за последовательность и логичность его исключительно полезных отзывов. Дугу Раддеру (Doug Rudder) за то, что обучил меня тонкому искусству передачи идей в письменной форме. Тиффани Тейлор (Tiffany Taylor) и Кэти Теннант (Katie Tennant) за высококачественное редактирование и корректуру текста, а также Тиму ван Дерзену (Tim van Deurzen) за глубину и качество профессионального рецензирования.

Хочу также поблагодарить Клару Шамбон (Clara Chambon) — мою любимую маленькую крестницу, Виржини Шамбон (Virginie Chambon) — милейшего человека на свете, всю семью Харшани, Афродити Катику (Afroditi Katika), Серхио Гарсеца (Sergio Garcez) и Каспера Бентсена (Kasper Bentsen) — замечательных инженеров-разработчиков, а также все сообщество Go.

Наконец, я хотел бы поблагодарить своих рецензентов: Адама Ванадамайкена (Adam Wanadamaiken), Алессандро Кампейса (Alessandro Campeis), Аллена Гуча (Allen Gooch), Андреса Сакко (Andres Sacco), Анупама Сенгупту (Anupam Sengupta), Борко Джурковича (Borko Djurkovic), Брэда Хоррокса (Brad Horrocks), Камала Какара (Samal Sakar), Чарльза М. Шелтона (Charles

М. Shelton), Криса Аллана (Chris Allan), Клиффорда Тербера (Clifford Thurber), Козимо Дамиано Прете (Cosimo Damiano Prete), Дэвида Кронкайта (David Cronkite), Дэвида Джейкобса (David Jacobs), Дэвида Моравека (David Moravec), Фрэнсиса Сеташа (Francis Setash), Джанлуиджи Спаньоло (Gianluigi Spagnuolo), Джузеппе Максиа (Giuseppe Maxia), Хироюки Мушу (Hiroyuki Musha), Джеймса Бишопа (James Bishop), Джерома Майера (Jerome Meyer), Джоэля Холмса (Joel Holmes), Джонатана Р. Чоута (Jonathan R. Choate), Йорга Роденбурга (Jort Rodenburg), Кита Кима (Keith Kim), Кевина Ляо (Kevin Liao), Лева Вайде (Lev Veyde), Мартина Денерта (Martin Dehnert), Мэтта Велке (Matt Welke), Нираджа Шаха (Neeraj Shah), Оскара Утбулта (Oscar Utbult), Пейти Ли (Peiti Li), Филиппа Джанертка (Philipp Janertq), Роберта Веннера (Robert Wenner), Райана Барроуска (Ryan Burrowsq), Райана Хубера (Ryan Huber), Санкета Найка (Sanket Naik), Сагадру Ройя (Satadru Roy), Шона Д. Вика (Shon D. Vick), Тада Майера (Thad Meyer) и Вадима Туркова. Все ваши предложения и замечания помогли сделать эту книгу лучше.

# Об этой книге

---

Книга «100 ошибок Go и как их избежать» содержит описание 100 распространенных ошибок, которые допускают Go-разработчики. Она в значительной степени сосредоточена на самом языке и его стандартной библиотеке, а не на внешних библиотеках или фреймворках. Обсуждения большинства ошибок сопровождаются конкретными примерами, иллюстрирующими те обстоятельства, когда такие ошибки могут совершаться. Эта книга — не какая-то догма. Каждое предлагаемое решение детализировано в той мере, чтобы передать контекст.

## Для кого эта книга

Эта книга предназначена для разработчиков, уже знакомых с языком Go. В ней не рассматриваются его основные понятия — синтаксис или ключевые слова. Предполагается, что вы уже занимались реальным проектом на Go. Но прежде чем углубляться в большинство конкретных тем, удостоверимся, что некоторые базовые вещи понимаются ясно и четко.

## Структура книги

Книга состоит из 12 глав:

Глава 1 «Go: просто научиться, но сложно освоить» объясняет, почему, несмотря на то что Go считается простым языком, его нелегко освоить досконально. В ней также приведены типы ошибок, которые мы рассмотрим в книге.

Глава 2 «Организация кода и проекта» содержит описание распространенных ошибок, которые могут помешать организовать программный код чистым, идиоматичным, удобным для дальнейшей обработки и поддержки образом.

В главе 3 «Типы данных» обсуждаются ошибки, связанные с основными типами, срезами и картами.

В главе 4 «Управляющие структуры» исследуются распространенные ошибки, связанные с циклами и другими управляющими структурами.

В главе 5 «Строки» рассматривается принцип представления строк и связанные с ним распространенные ошибки, приводящие к неточности или неэффективности кода.

В главе 6 «Функции и методы» обсуждаются распространенные проблемы, связанные с функциями и методами, такие как выбор типа получателя и предотвращение распространенных ошибок отложенного выполнения (`defer`).

В главе 7 «Обработка ошибок» рассматривается идиоматическая и точная обработка ошибок в Go.

В главе 8 «Конкурентность: основы» представлены основные концепции конкурентности. Мы разберем, почему конкурентность не всегда быстрее, в чем различия между конкурентностью и параллелизмом, а также обсудим типы рабочей нагрузки.

В главе 9 «Конкурентность: практика» рассмотрены примеры ошибок, связанных с конкурентностью при использовании каналов, горутин и других примитивов Go.

Глава 10 «Стандартная библиотека» содержит описание распространенных ошибок, допускаемых при использовании стандартной библиотеки с HTTP, JSON или (например) `time API`.

В главе 11 «Тестирование» обсуждаются ошибки, которые делают тестирование и бенчмаркинг менее универсальными, эффективными и точными.

Глава 12 «Оптимизация» завершает книгу. В ней исследуются способы того, как оптимизировать приложение для повышения его производительности, — от понимания основ функционирования центрального процессора до конкретных тем, связанных с Go.

## О коде в книге

Книга содержит множество примеров исходного кода как в нумерованных листингах, так и в тексте. В обоих случаях исходный код форматируется моноширинным шрифтом, в отличие от обычного текста. Иногда для кода также применяется

**жирный шрифт**, чтобы выделить фрагменты, изменившиеся по сравнению с предыдущими шагами, — например, при добавлении новой функциональности в существующую строку кода.

Во многих случаях оригинальная версия исходного кода переформатируется; добавляются разрывы строк и измененные отступы, чтобы код помещался на странице. Иногда даже этого оказывается недостаточно и в листинги включаются маркеры продолжения строк (➡). Также из исходного кода часто удаляются комментарии, если код описывается в тексте.

Исполняемые фрагменты кода можно загрузить из версии liveBook (электронной) по адресу <https://livebook.manning.com/book/100-go-mistakes-how-to-avoid-them>. Полный код примеров книги доступен для загрузки на сайте Manning по адресу <https://www.manning.com/books/100-go-mistakes-how-to-avoid-them> и GitHub <https://github.com/teivah/100-go-mistakes>.

## Форум liveBook

Приобретая книгу «100 ошибок Go и как их избежать», вы получаете бесплатный доступ к закрытому веб-форуму издательства Manning (на английском языке), на котором можно оставлять комментарии о книге, задавать технические вопросы и получать помощь от автора и других пользователей. Чтобы получить доступ к форуму, откройте страницу <https://livebook.manning.com/book/100-go-mistakes-how-to-avoid-them/discussion>. Информацию о форумах Manning и правилах поведения на них см. на <https://livebook.manning.com/#!/discussion>.

В рамках своих обязательств перед читателями издательство Manning предоставляет ресурс для содержательного общения читателей и авторов. Эти обязательства не подразумевают конкретную степень участия автора, которое остается добровольным (и неоплачиваемым). Задавайте автору хорошие вопросы, чтобы он не терял интереса к происходящему! Форум и архивы обсуждений доступны на веб-сайте издательства, пока книга продолжает издаваться.



## *Об авторе*

---

**ТЕЙВА ХАРШАНИ** — старший инженер-программист в Docker. Работал в области страхования, транспорта и в отраслях, где критически важна безопасность, например в управлении воздушным движением. Увлечен языком Go и тем, как разрабатывать и реализовывать на нем надежные приложения.

## *Иллюстрация на обложке*

---

На обложке книги — рисунок под названием «Femme de Buccari en Croatie» («Женщина из Бакара, Хорватия»).

Иллюстрация взята из вышедшего в 1797 году каталога национальных костюмов, составленного Жаком Грассе де Сен-Савьером. Каждая иллюстрация этого каталога тщательно прорисована и раскрашена от руки. В прежние времена по одежде человека можно было легко определить, где он живет и какова его профессия или положение. Manning отдает дань изобретательности и инициативности компьютерных технологий, используя для своих изданий обложки, демонстрирующие богатое вековое разнообразие региональных культур, оживающее на изображениях из собраний, подобных этому.

## *От издательства*

---

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

Мы будем рады узнать ваше мнение!

На веб-сайте издательства [www.piter.com](http://www.piter.com) вы найдете подробную информацию о наших книгах.