

*Посвящается Шэрон,  
которая терпела меня во время карантина  
и двадцать лет до того*



# Оглавление

---

Предисловие от издательства .....	9
Благодарности .....	10
Предисловие .....	11
Об авторе .....	13
О коде .....	14
О вас .....	15
И еще одно .....	16
Задача 1. Число $\pi$ .....	17
Задача 2. С пустыми руками .....	21
Задача 3. Однажды в Krakове .....	23
Задача 4. Нет у меня ничего .....	27
Задача 5. Сыроядение .....	29
Задача 6. Мы уже приехали? .....	31
Задача 7. Могут ли числа лгать? .....	33
Задача 8. Сортировка во сне .....	37
Задача 9. Точно вовремя .....	41
Задача 10. Простое дописывание .....	45

<b>Задача 11. Что в журнале?</b> .....	<b>49</b>
<b>Задача 12. Странное число</b> .....	<b>53</b>
<b>Задача 13. Целые числа и диапазоны</b> .....	<b>57</b>
<b>Задача 14. Несколько личин</b> .....	<b>63</b>
<b>Задача 15. Повесть о двух городах</b> .....	<b>65</b>
<b>Задача 16. Что у нас в канале?</b> .....	<b>69</b>
<b>Задача 17. **Int**ересная строка</b> .....	<b>73</b>
<b>Задача 18. Отложенная работа</b> .....	<b>77</b>
<b>Задача 19. Ошибка или не ошибка</b> .....	<b>81</b>
<b>Задача 20. Что в строке?</b> .....	<b>83</b>
<b>Задача 21. Двойняшки</b> .....	<b>87</b>
<b>Задача 22. Отсчитайте мне миллион</b> .....	<b>91</b>
<b>Задача 23. Кто следующий?</b> .....	<b>97</b>
<b>Задача 24. Ох уж эти флаги</b> .....	<b>99</b>
<b>Задача 25. Даю разрешение</b> .....	<b>103</b>
<b>Предметный указатель</b> .....	<b>107</b>

# Предисловие от издательства

---

## Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте [www.dmkpress.com](http://www.dmkpress.com), зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com); при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу [http://dmkpress.com/authors/publish\\_book/](http://dmkpress.com/authors/publish_book/) или напишите в издательство по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

## Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com), и мы исправим это в следующих тиражах.

## Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

## Благодарности

---

Я благодарен за все отзывы: от указаний на ошибки в коде до исправления грамматических ошибок.

Ниже приводится список людей, помогавших мне; приношу извинения всем, кого я забыл упомянуть.

- Ади Тебека – за корректуру и комментарии.
- Дэн Аллен – за помощь на форумах asciidocor. Я пользовался этим замечательным сайтом при написании первого варианта книги.
- Дэйв Чини – за вступительное слово.
- Дэвид Бордейник – за комментарии.
- Игон Элбр – за нарисованных им прекрасных сусликов, которых он поместил также в лицензию CC0.
- Илад Эйял – за корректуру.
- Элиран Бивас – за корректуру и комментарии.
- Джаред Дэвис – за отзыв.
- Рэн Тэвори – за комментарии.
- Йони Дэвидсон – за комментарии.

Я также благодарен Брэду Фицпатрику и другим, кто публикует тесты по Go, откуда я черпал идеи для этих задач.

# Предисловие

---

Язык программирования Go прост, но, как и у любого другого языка, у него есть свои причуды. В этой книге причуды используются как дополнительная возможность чему-то научиться. Осознав пробелы в своих знаниях, вы станете лучше выполнять свою работу.

Есть немало исследований, доказывающих, что люди, делающие ошибки во время обучения, в итоге обучаются лучше, чем те, кто ошибок не делает. Если вы будете иметь это в виду при исправлении дефектов в программе, то охота за ошибками будет доставлять вам удовольствие, а после исправления каждой вы, как разработчик, будете становиться лучше.

Эти задачи помогут вам избежать ошибок. Некоторые задачи взяты из моего личного опыта поставки ПО с ошибками, а некоторые – из опыта других людей.

Все мы, профессионалы, любим решать задачки. Но вы можете также использовать их, чтобы произвести впечатление на коллег, устроить интеллектуальный бой и вместе стать лучше.

В основе многих задач лежат вопросы, которые я предлагал на конференциях и профессиональных собраниях. Оказалось, что людям это нравится и оживляет атмосферу.

В начале каждой главы приводится короткая программа на Go, и вам предлагается угадать, что произойдет при попытке ее выполнить. Ответы могут быть такими:

- не компилируется;
- паникует;
- взаимоблокировка;
- что-то печатается (например, [\[1\]](#) [\[2\]](#) [\[3\]](#)).

---

### Версия Go

---



При выполнении кода я использовал версию Go 1.14.1, но в будущих версиях результат *может* быть иным.

---

Прежде чем смотреть ответ и объяснения, попытайтесь угадать результат самостоятельно. А потом я рекомендую выполнить код и посмотреть, что получится, и лишь после этого переходить к чтению решения и объяснения. Я преподаю программирование много лет и нахожу такой порядок действий весьма эффективным.

## Об авторе

---

Мики Тебека получил степень бакалавра компьютерных наук в университете имени Бен-Гуриона. Там же он прослушал курс вычислительной лингвистики.

У Мики страсть к преподаванию и наставничеству. Он ведет много семинаров по различным техническим предметам по всему миру и был наставником многих начинающих разработчиков на их пути к успеху. Мики создал ряд собственных проектов с открытым исходным кодом и принимает участие еще в нескольких, в том числе Python. Он пишет на Python более 23 лет.

Мики является автором книг «Pandas Brain Teasers», «Python Brain Teasers» и «Forging Python», он автор, зарегистрированный на сайте LinkedIn Learning, и организатор мероприятий Go Israel Meetup, GopherCon Israel и PyData Israel Conference.

# О коде

---

Код, приведенный в книге, можно найти по адресу  
<https://pragprog.com/titles/d-gobrain/go-brainteasers/>.

Я старался максимально сократить код и удалял все не имеющее прямого отношения к задаче. Обычно так код *не* пишется.

## О вас

---

Предполагается, что вы в какой-то мере знакомы с Go и имеете опыт программирования на этом языке. Эта книга не учебник по Go. Если вы совсем не знаете Go, боюсь, что эти задачи не для вас.

## И еще одно

---

Когда будете прорабатывать задачи, представляйте себя Нэнси Дрю, Шерлоком Холмсом или еще каким-нибудь детективом, который пытается раскрыть убийство, а убийцей является *вы*. Примите такую точку зрения:

«Отладка программы сродни работе кинематографического детектива в фильме, расследующего убийство, совершенное вами же» (Филипе Фортес).

Мне кажется, что при таком отношении проще понять, что происходит, да к тому же и веселее. Так что получайте удовольствие от разгадывания головоломок – глядишь, освоите пару-другую новых трюков.

Если захотите узнать больше, пишите на адрес [info@353solutions.com](mailto:info@353solutions.com), и мы организуем практический семинар на интересующую вас тему. Также на сайте <https://www.353solutions.com> есть богатая подборка семинаров.

Оставайтесь любознательными и докапывайтесь до сути вещей!

Мики Тебека, март 2020

## Задача 1

### Число π

pi.go

```
package main

import (
    "fmt"
)

func main() {
    var π = 355 / 113.0
    fmt.Println(π)
}
```

---

Угадайте результат



Попробуйте угадать результат, не переворачивая страницу.

---

Программа печатает следующий результат: `3.1415929203539825`

---

## Обсуждение

Тут у нас две достойные удивления вещи: во-первых, оказывается, что `π` – допустимый идентификатор, а во-вторых – что выражение `355 / 113.0` компилируется.

Начнем с `π`. В спецификации языка Go по поводу идентификаторов написано:

«Идентификаторы именуют такие программные сущности, как переменные и типы. Идентификатор представляет собой последовательность одной или более букв и цифр. Первый символ идентификатора должен быть буквой».

Буквами могут быть любые буквы Юникода, в том числе `π`. Это, может быть, и забавно, но определенно усложнит жизнь вашим коллегам. Я легко могу набрать `π` в редакторе Vim, в котором работаю, но в большинстве редакторов и IDE потребуется больше усилий.

Единственное место, где, на мой взгляд, идентификаторы с символами Юникода могут быть полезны, – запись математических формул в коде. Во всех остальных случаях ограничьтесь старым добрым ASCII.

Теперь что касается `355 / 113.0`. Система типов Go не допускает деления (или любых других математических операций) между целым (`355`) и числом с плавающей точкой (`113.0`). Но здесь в правой части оператора присваивания `=` находятся константы, а не переменные. Тип константы определяется в момент использования; в данном случае компилятор преобразует `355` в число с плавающей точкой, чтобы выполнить операцию.

Если вы сначала присвоите значения `355` и `113.0` переменным и попробуете выполнить ту же программу, то ничего не получится. Следующий код не откомпилируется:

**pi\_var.go**

```
package main

import (
    "fmt"
)
func main() {
    a, b := 355, 113.0
    var π = a / b
    fmt.Println(π)
}
```

## Для дополнительного чтения

Спецификация констант  
[golang.org/ref/spec#Constants](https://golang.org/ref/spec#Constants).

Константы  
[blog.golang.org/constants](https://blog.golang.org/constants).

Ardan Labs «Introduction to Numeric Constants in Go»  
[ardanlabs.com/blog/2014/04/introduction-to-numeric-constants-in-go.html](https://ardanlabs.com/blog/2014/04/introduction-to-numeric-constants-in-go.html).

«Википедия» о приближениях π  
[en.wikipedia.org/wiki/Approximations\\_of\\_%CF%80](https://en.wikipedia.org/wiki/Approximations_of_%CF%80).

Идентификаторы в спецификации языка Go  
[golang.org/ref/spec#Identifiers](https://golang.org/ref/spec#Identifiers).



Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

[e-Univers.ru](http://e-Univers.ru)