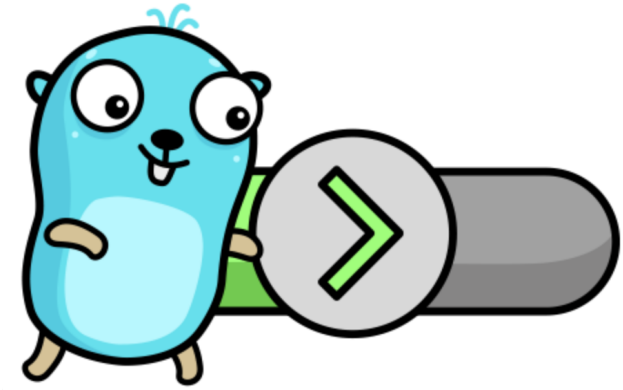
Вступительное слово

Категорически приветствуем!

В этом уроке мы поговорим о том, какие инструменты понадобятся для прохождения курса, где общаться и куда жаловаться :)

Вперёд!



Репозиторий курса

Примеры из теоретической части, а также заготовки для домашек лежат в

https://github.com/www-golang-courses-ru/advanced-dealing-with-panic-in-go

Если вы незнакомы с git, то самое время познакомиться. В целом будет достаточно пары команд:

```
$ git clone
https://github.com/www-golang-courses-ru/advanced-dealing-with-panic-in-go.git
Cloning into 'advanced-dealing-with-panic-in-go'...


$ cd advanced-dealing-with-panic-in-go
```

Форкать и загружать свои решения в GitHub особого смысла **не имеет**, так как поделиться решением, а также ознакомиться с решениями коллег и авторов можно будет с помощью платформы Stepik.

Чат курса

Общаемся в комментариях к урокам и в Telegram: https://t.me/goinpractice.

Можно задавать любые вопросы (а лучше давать ответы) по материалу, заданиям и по языку в целом.

Не забываем пользоваться правилом nometa.

У каждой задачи есть индекс урока, номер шага и имя, так что здорово, если вы их укажите, например

Привет! Не проходят тесты в задаче 2.3.6 "Transaction Rollback" ...

Так мы сразу поймём, о чём идёт речь, а вы сможете поискать, не было ли уже подобных вопросов от других студентов?

VPN

Некоторые ссылки из списков литературы, а также используемый нами гошный плейграунд https://goplay.tools/ могут быть недоступны из России и некоторых стран СНГ.

В таком случае рекомендуем пользоваться VPN, например:

- Плагин для Chrome: Touch VPN
- Windscribe

В любом случае б**о**льшую часть примеров можно найти в репозитории курса.

Как сдавать задачи?

Практически все задачи не имеют примеров ввода и вывода:

```
Sample Input:
```

```
Sample Output:

SUCCESS
```

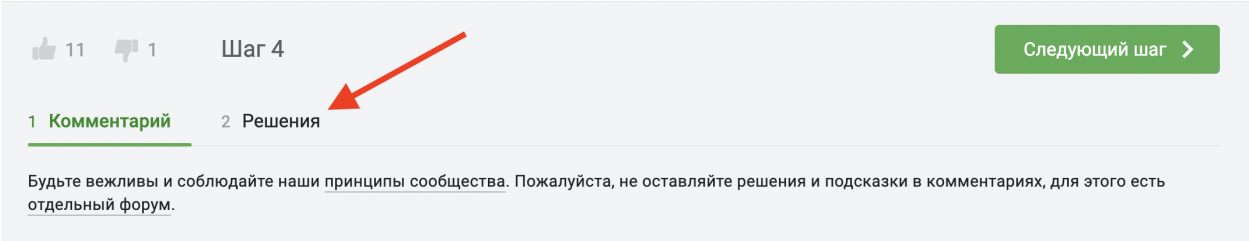
Всё потому, что они тестируются не через STDIN-STDOUT, а через полноценные гошные тесты, эквивалентные тем, что вы видите в заготовке к задаче (пример).

Поэтому правило одно – сделайте так, чтобы ваша реализация проходила тесты в заготовке и скорее всего она будет принята и Stepik'ом.

Если в какой-то задаче вам не хватает определённых импортов – пишите в комментариях к задаче или сразу в чат – мы добавим их!

Не забывайте обращаться к вкладке "Решения" (под задачей), чтобы сравнить своё решение с авторским.

А также публикуйте свои решения, если они отличаются от имеющихся.



Модули построены так, что вам не будет доступен следующий модуль, пока вы не сдадите б**о**льшую часть задач в текущем. Мы считаем практику неотъемлемой частью любого обучения и забивать на неё болт не выйдет. Если вам непонятно условие или вы имеете замечания по задаче, то не стесняйтесь писать нам или в комментариях на странице к задаче. Спасибо!

Что делать, если задача проходит локально, но не проходит в Stepik?

Скорее всего тесты задачи были изменены или дополнены, выполните

```
git pull origin master
```

в корневой директории репозитория курса.

Если же это не помогло, то пишите нам – будем разбираться вместе 🙋‍♂️

Как устроена разбалловка у задач?

На наших курсах можно встретить следующие задачи:

Тип задачи	Стоимость (баллов)
Задача со свободным ответом (поболтайка)	1
Задача на самостоятельную работу (нет возможности проверить её средствами Stepik)	1
Тест	2
Задача начального / среднего уровня	5
Задача продвинутого уровня	10

Если у вас не получается решить задачу, то не нужно "зарываться", попробуйте:

- убедиться, что вы не пропустили теорию перед задачей;
- взять паузу и вернуться к задаче позже;
- внимательнее перечитать условие;
- внимательнее посмотреть на тесты к задаче;
- спросить совета в чатике курса.

Компилятор

Go >= 1.18

На момент завершения работы над курсом последней версией Go являлась **1.17.6**, но затем вышел **Go 1.18** и Stepik перевёл свою песочницу на него, поэтому у вас не остаётся выбора 🙂.

Установить актуальную версию компилятора можно с помощью страницы https://golang.org/doc/install.

Также курс изредка касается версий 1.12, 1.14 и 1.17. Если вам хочется самим воспроизвести какие-то примеры на конкретной версии языка, то её можно доставить себе в систему, следуя данной инструкции.

Stepik будет запускать ваши программы с помощью команды

```
# go 1.14.4
```

```
$ go build -ldflags "-extldflags -static"
```

Среда разработки

В общем случае для выполнения домашних заданий достаточно простого текстового редактора и компилятора.

Но мы крайне не рекомендуем разрабатывать в неполноценных IDE (nano, ~~vim~~, Sublime Text, ~~Emacs~~ и т.д.), особенно если они плохо настроены. Часто это влечёт к примитивным ошибкам, связанных с отсутствием подсказок от среды, хоть и при должной сноровке ускоряет скорость разработки.

В течение курса мы неоднократно увидим, как здорово иметь IDE под рукой.

На данный момент наиболее популярными ср**е**дами являются:

- GoLand - платная, но очень мощная приблуда от JetBrains. К сожалению, получить лицензионный ключ от Stepik можно только для открытых курсов.

- Visual Studio Code - популярная, благодаря своей бесплатности, IDE. Расширение для Go официально поддерживается командой Google. Использует gopls - официальный language server, который вы, в принципе, можете запустить самостоятельно и внедрить в привычный вам редактор (инструкции здесь: поддерживаются VSCode, Vim/ Neovim, Emacs, Atom, Sublime Text, Acme).

Больше про редакторы и IDE в официальной документации.

Напишите в комментариях, чем пользуетесь и что посоветовали бы коллегам!

github.com/golang/go

На курсе мы часто обращаемся к исходному коду стандартной библиотеки и самого компилятора Go.

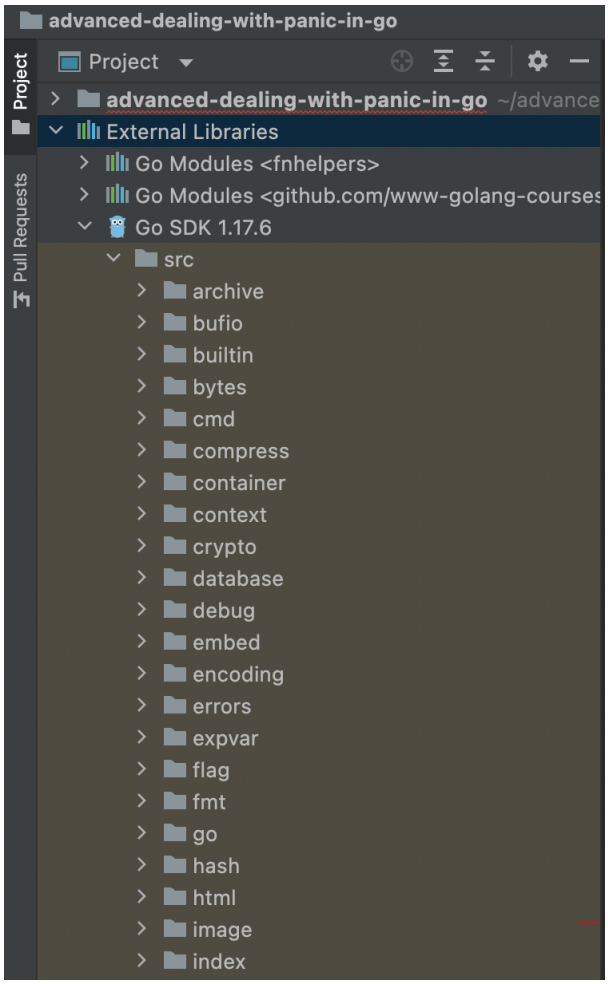
Так как вы установили Go, его исходники (а не только бинарник компилятора) уже можно найти в системе:

```
$ go env GOROOT
/usr/local/go

$ ls -1a $(go env GOROOT)
.
..
AUTHORS
CONTRIBUTING.md
CONTRIBUTORS
LICENSE
PATENTS
README.md
SECURITY.md
VERSION
api
bin
doc
favicon.ico
lib
misc
pkg
robots.txt
src

test
```

Например, в них можно копаться через GoLand:



Или можно выкачать нужную версию сорцов в удобное для себя место через https://github.com/golang/go.

Дополнительно не забываем про:

- спецификацию языка Go;
- документацию стандартной библиотеки языка Go;
- подсказки по написанию эффективного кода на Go;
- ответы на часто задаваемые вопросы про Go.

Не проходите курс с мобильного устройства

Мы не рекомендуем проходить курс с мобильного устройства, так как могут быть проблемы с:

- отображением комментариев к урокам;
- отображением форума решений;
- процессом создания и форматирования собственных комментариев / решений.

Комментарии к шагам и решениям составляют огромный пласт информации, дополняющей курс и затрагивающей смежные темы. **Мы бы не хотели, чтобы они остались без вашего внимания.**

Заключение

Теперь, когда всё выкачено, установлено и настроено, можно переходить к работе с паникой в Go.

Но перед этим мы познакомимся с **механизмом отложенных функций**.