

……

<https://www.youtube.com/watch?v=tQeHFUYHD-c&list=PLoWGNURguz9W8h2j7UJnwdPZVZy2pDfMT&index=38> питон курс видео

<file:///C:/Users/User/Downloads/Раевская%20О.В.%20-%20Французско-русский.%20Русско-французский%20словарь%20(Карманная%20библиотека%20словарей)%20-%202010.pdf> франц слов

<https://ru.stackoverflow.com/questions/199624/static-cast-и-int-это-одно-и-то-же> cast that article

<http://algolist.manual.ru/ds/> перед собес стр и алг-мы

<https://prog-cpp.ru/> с++ сущ-ет

<https://pythonz.net/references/named/str.splitlines/> питон справ все о строках!!

<http://cpptasksolver.blogspot.com/2011/01/blog-post.html> шахм зад-чи на с++

……..

<https://github.com/OtusGolang/home_work> голанг задачи

<https://golang-blog.blogspot.com/2019/01/effective-go-slices.html> эффект голанг

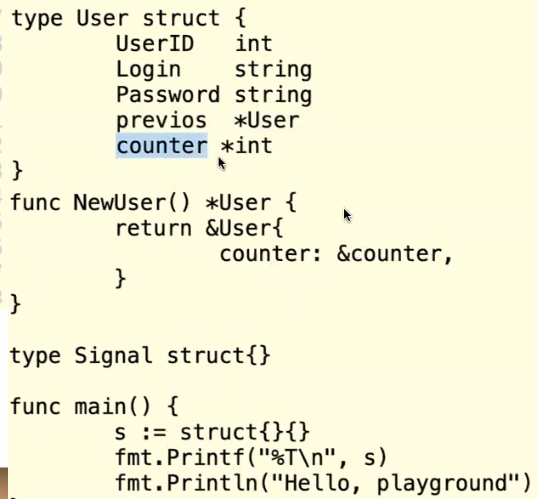
<https://golang.org/doc/> докум-ия

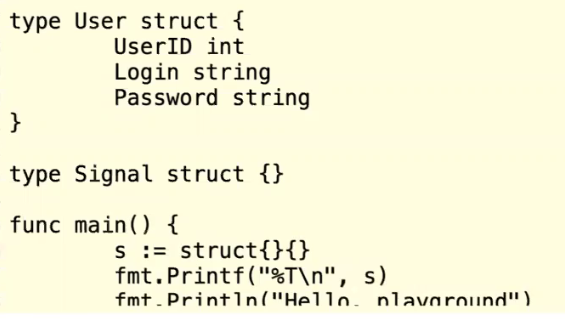
<https://metanit.com/go/tutorial/8.5.php> книга

<https://go-tour-ru-ru.appspot.com/list> тур по Go

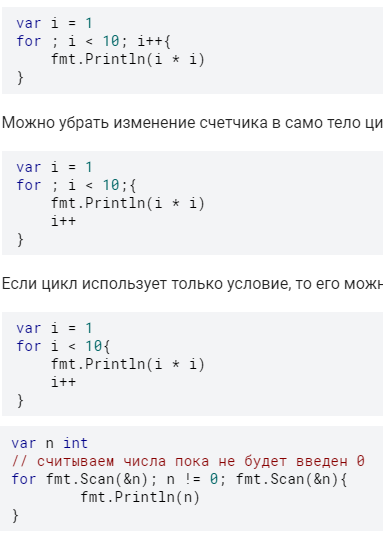
<https://go-tour-ru-ru.appspot.com/moretypes/15> где остановилась

<https://golangs.org/interface> шик уроки го

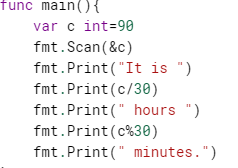


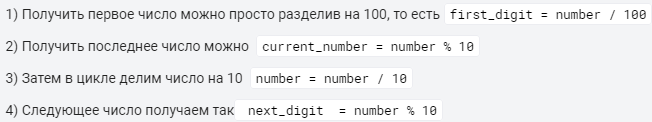


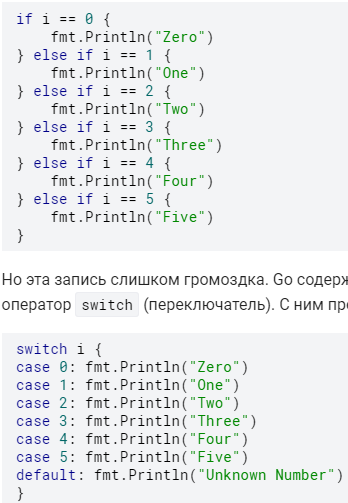


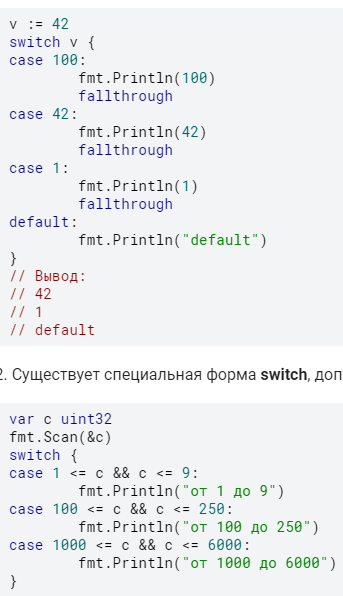


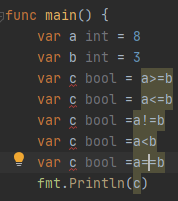






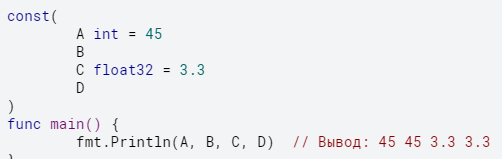


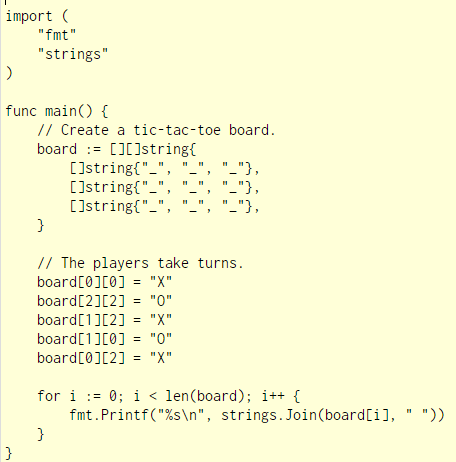


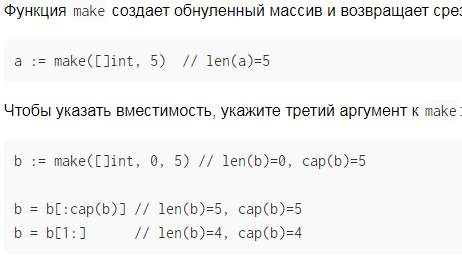


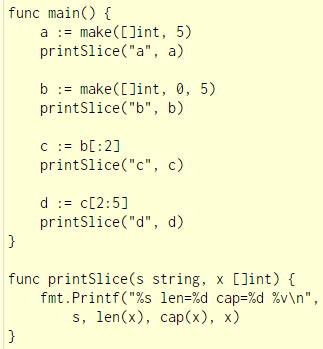


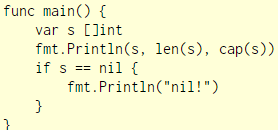


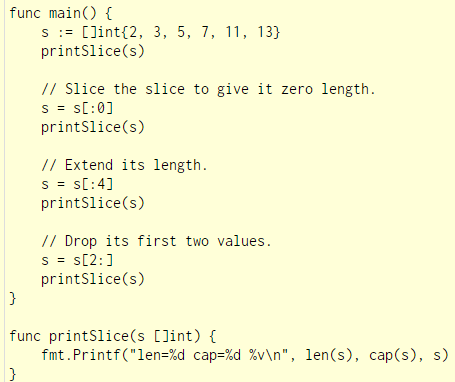


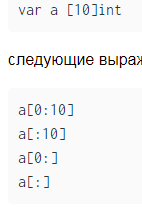


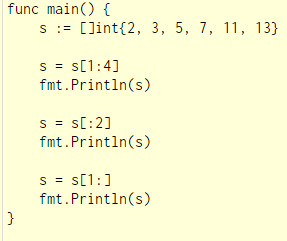




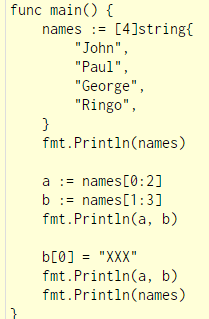


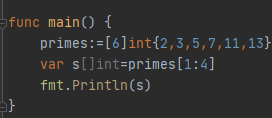


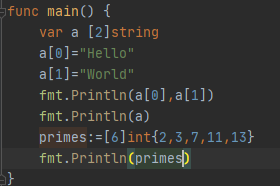


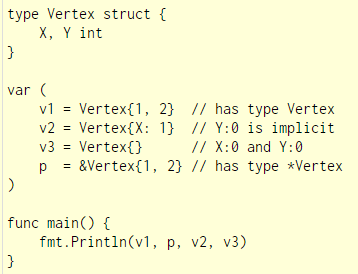


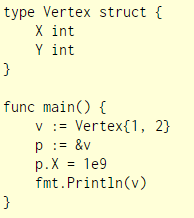


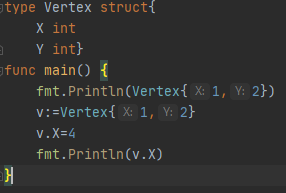


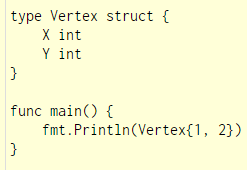


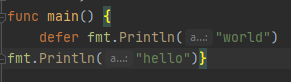


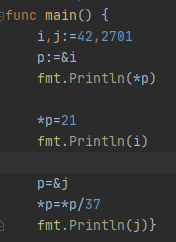


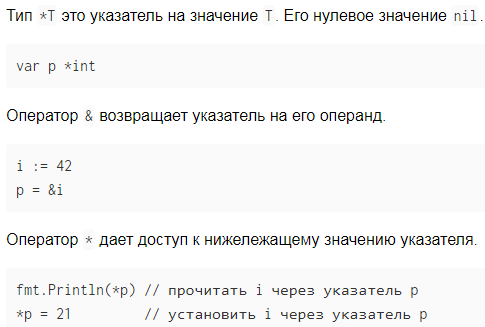


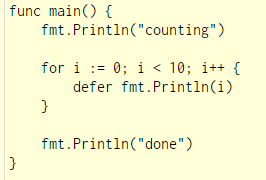


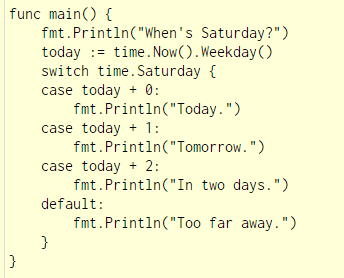


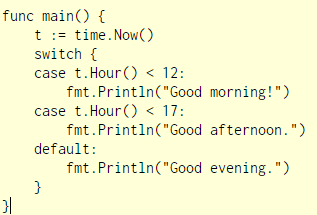


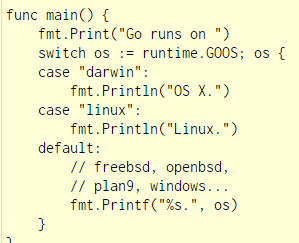


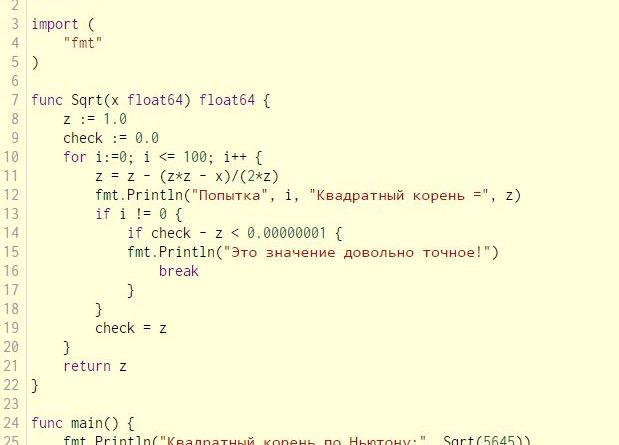


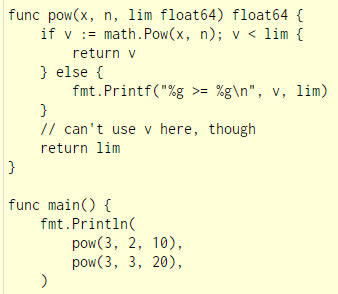


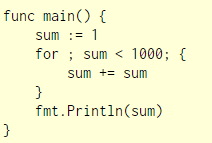


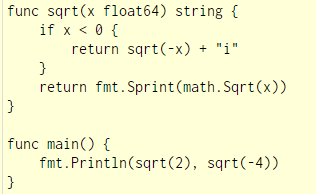




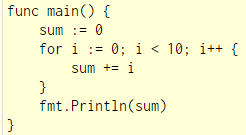


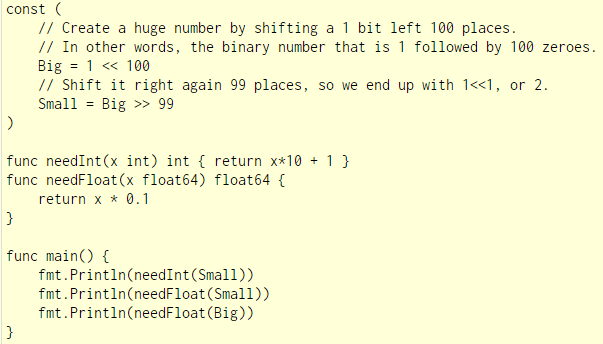


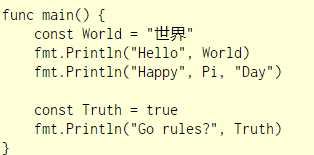




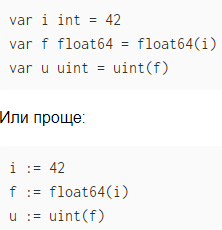


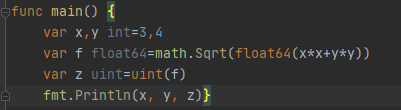


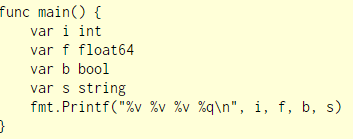


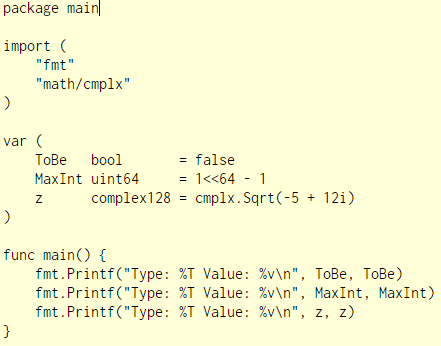


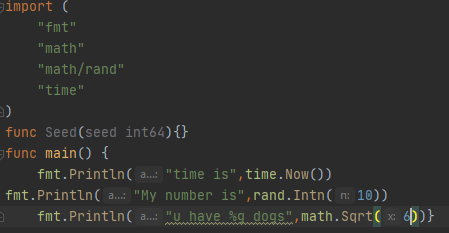


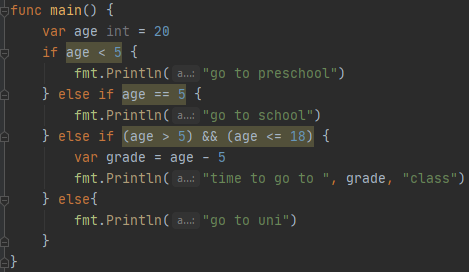


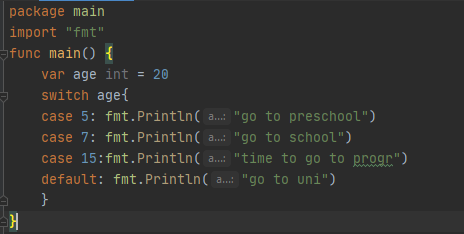


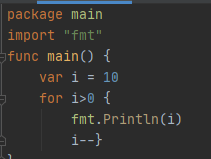


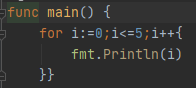


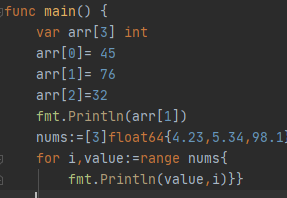


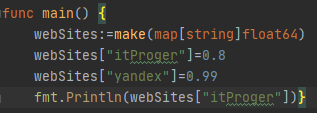


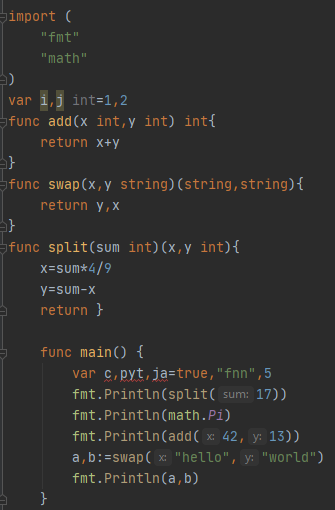


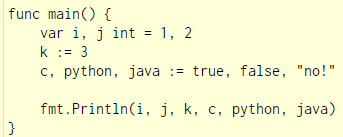


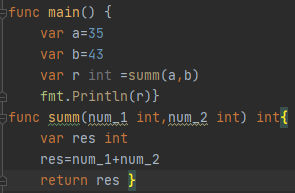


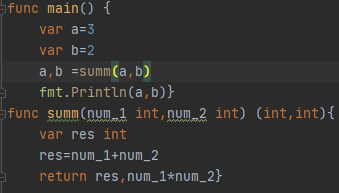


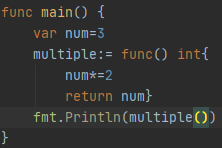


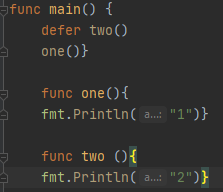


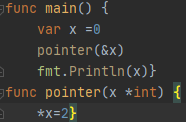


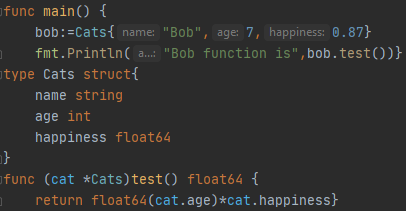


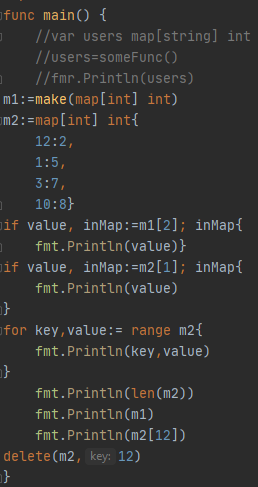


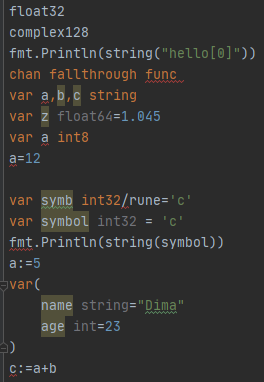


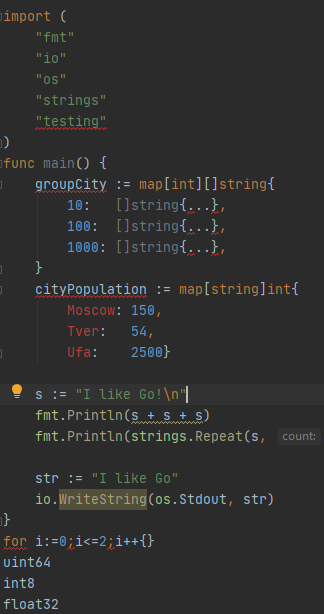


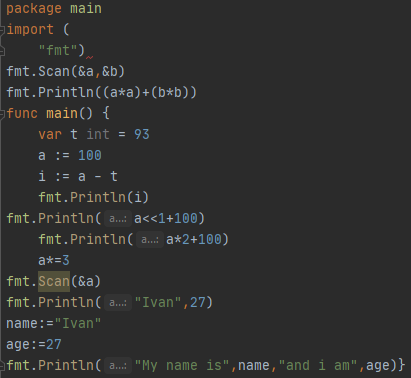












Измеряют оптическую плотность содержимого лунок планшета с помощью спектрофотометра, используя длину волны 630 нм. Минимальная подавляющая концентрация антибиотика определяется при построении графика соотношения концентрации антибиотика и оптической плотности и находится в точке выхода кривой на нулевой уровень значения оптической плотности. Способ обеспечивает повышение точности определения чувствительности культур Helicobacter pylori к антибактериальным препаратам.

МИК находится где опт пл =0 либо когда график выходит на плато

длина волны для измерения

) Определить зависимость МИК для выбранного наиболее эффективного производного пиррола по отношению к Cr. neoformans, от растворителя (ДМСО, этанол, вода), используемого для этого вещества

https://golangs.org/ курс для продв

https://www.youtube.com/watch?v=9Pk7xAT\_aCU&list=PLrCZzMib1e9q-X5V9pTM6J0AemRWseM7I&index=1 техносф

АФП могут также действовать, предотвращая образование грибковых биопленок и уничтожая предварительно сформированные биопленки с помощью механизмов, связанных с возмущением клеточной стенки, ингибированием адгезии планктонных грибковых клеток к поверхности, генной регуляцией и генерацией активных форм кислорода (АФК).

1)Pd@Ag Nanosheets in Combination with Amphotericin B Exert a Potent Anti-Cryptococcal Fungicidal Effect

Zhang, C., Chen, M., Wang, G., Fang, W., Ye, C., Hu, H., … Liao, W. (2016). Pd@Ag Nanosheets in Combination with Amphotericin B Exert a Potent Anti-Cryptococcal Fungicidal Effect. PLOS ONE, 11(6), e0157000. doi:10.1371/journal.pone.0157000

2)Thirach, S., Tragoolpua, K., Punjaisee, S., Khamwan, C., Jatisatienr, C., & Kunyanone, N. (2003). ANTIFUNGAL ACTIVITY OF SOME MEDICINAL PLANT EXTRACTS AGAINST CANDIDA ALBICANS AND CRYPTOCOCCUS NEOFORMANS. Acta Horticulturae, (597), 217–221. doi:10.17660/actahortic.2003.597.31

Эвгенол является производным фенола

вызывает повреждение мембран

, утечку цитоплазматического содержимого из клеток ,и денатурацию белков (Thirach, 2003)

Серебряные и золотые наночастицы обладают противогрибковыми свойствами.

Cryptococcus neoformans наиболее восприимчив к Pd@Ag NSs, который проявлял мощную противогрибковую активность в отношении различных криптококковых штаммов, включая устойчивые к флуконазолу изоляты.

Антикриптококковая активность Pd@Ag NSs была значительно выше, чем у флуконазола, и аналогична таковой у Амфотерицина в (AmB)

. При относительно высоких концентрациях Pd@Ag NSs проявлял фунгицидную активность в отношении Cryptococcus spp., что, вероятно, может быть связано с нарушением целостности клеток, внутриклеточного синтеза белка , энергетического обмена, внутриклеточного гомеостаза C. neoformans.

ионы Ag, высвобождаемые из наночастиц, могут прикрепляться к клеточной стенке/мембране микроорганизмов и индуцировать дозозависимое повреждение клеточной стенки и мембранной структуры ,нарушением синтеза белка и энергетического метаболизма.

Pd@Ag NSs в низких концентрациях также проявлял хороший ингибирующий эффект на FLZ-резистентные штаммы, что указывает на его потенциал в лечении криптококкоза.

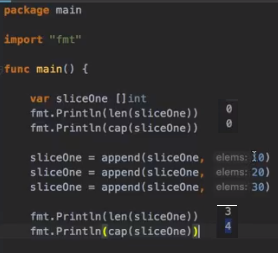
И выход с веществом и с ДМСО сосчитать в % от выхода с гликолипидом и сделать диаграмму

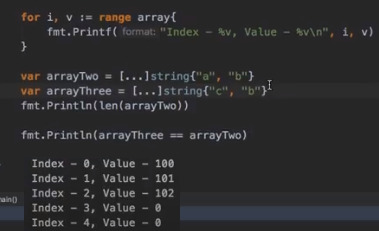
сюда же дать и вторую диаграмму - % выхода фосфата от выхода с гликолипидом – и тоже дать диаграмму

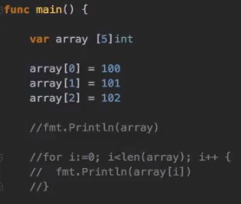
https://www.frontiersin.org/articles/10.3389/fmicb.2019.02169/full#B49 статья по катионным белкам против грибн биопленок

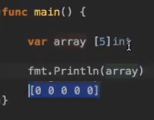
https://www.youtube.com/user/PlurrimiTube/playlists гоша дударь

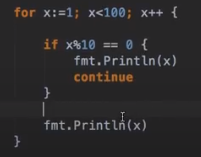
https://github.com/OtusGolang/home\_work -все дз по голанг

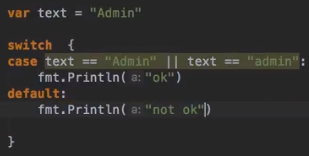


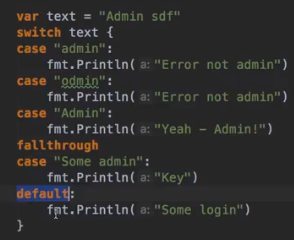


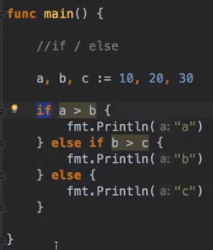


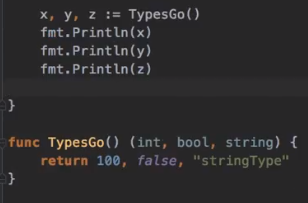


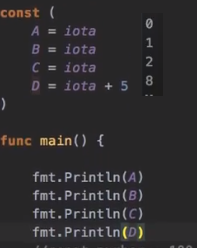


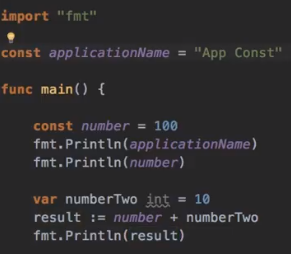


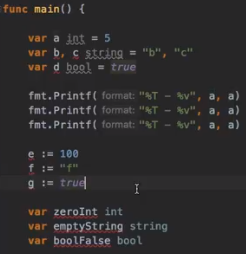


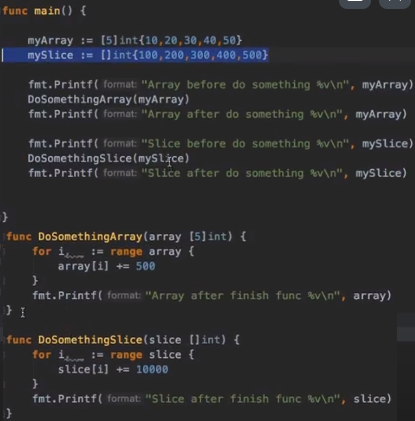


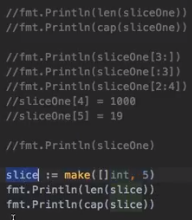




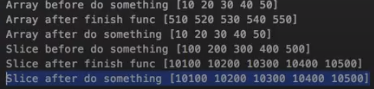


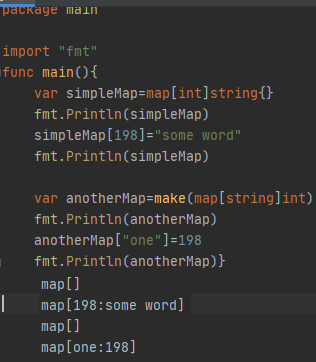


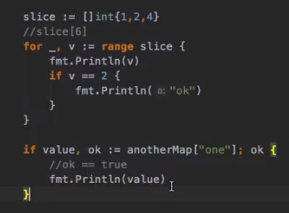


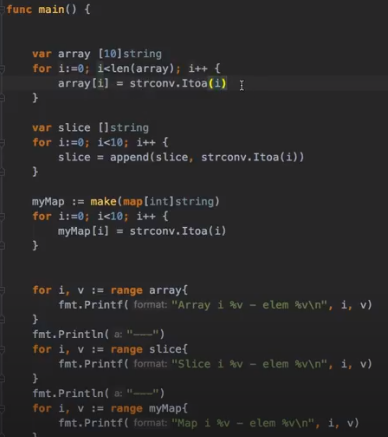


Пакетн,файлов,блок обл вид-сти.Конст можобъявл вне и внутри ф-ии. Массивы,Срезы,Структуры,Мaps-эти типы данн-главн. Чаще массивов исп-ся слайсы. Слайс сост из данн(кол-длина) и базов массива(кап-увел-ся в 2 раза-4-8-16). В слайсе посл знач искл-ся. Простые типы данн перед-ся по знач -то есть копир-ся.

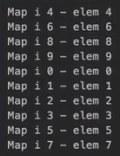


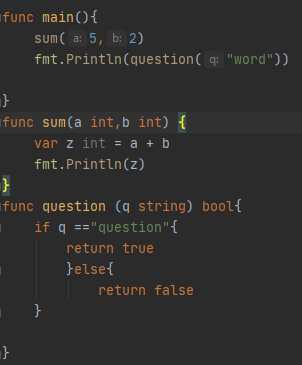




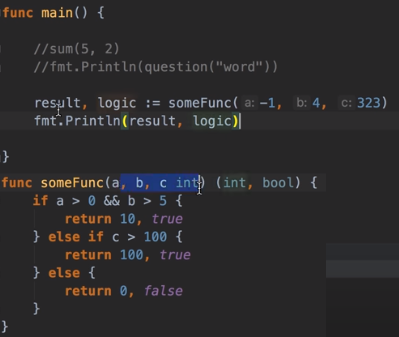
\

В мапе знач-ия возвр-ся не по пор-ку

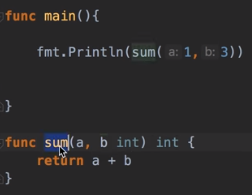


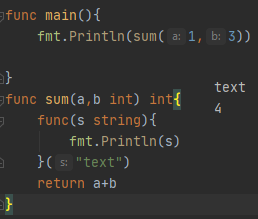


Мэп это некий словарь

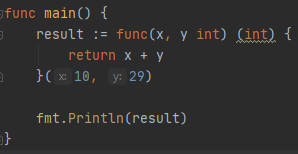


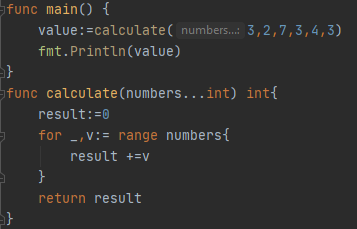
Именован ф-ия

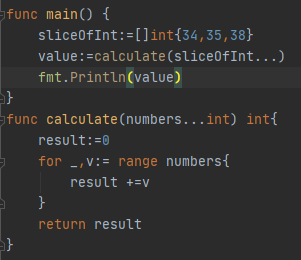




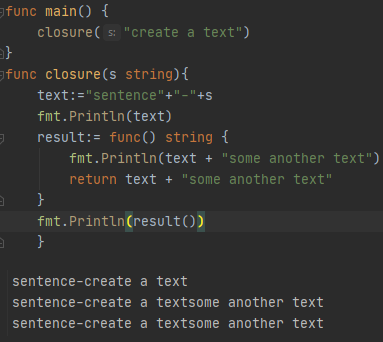
Анонимн ф-ия



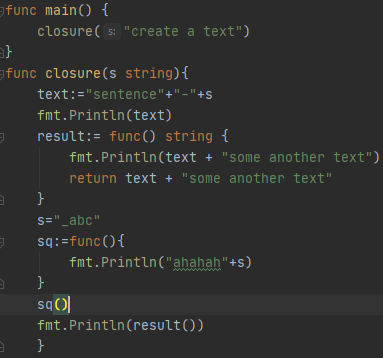


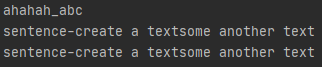


Замыкание

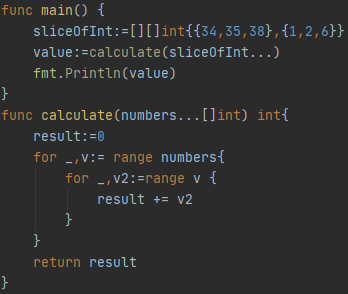


Замыкание запомин сост-ие из кот вызыв-ся. Переменные замык-ия сод то сост-ие кот-ое было у них в мом вызова.



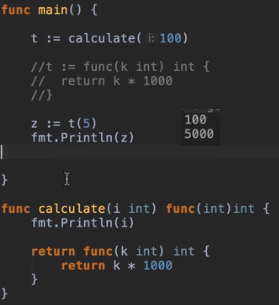


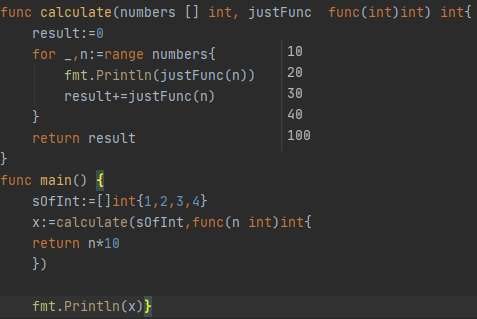
Разворачив-ие колл-ии

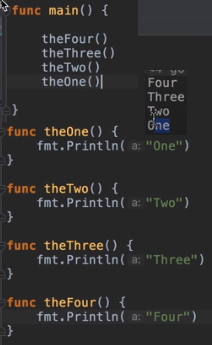


…

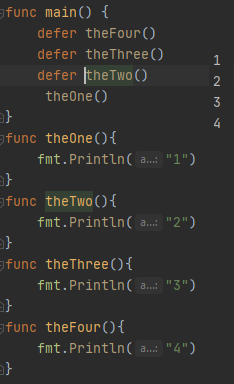
Список-это тип данн. Кажд ф-ию м-но исп-ть в кач пар-ра для др ф-ии или в кач возвращ-м знач-ия.



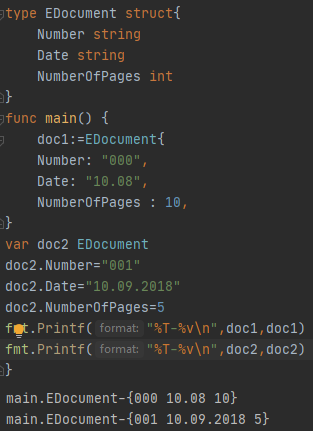




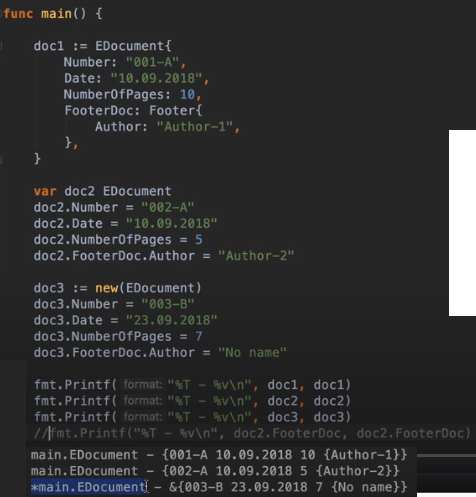
Задержка-это стек-в файл идет запись(не знаем ск-ко данн,н-но закр файл когда все записано).



Вместо ООП есть стр-ра( как и мэп,массив ,инт-ф-это сотавн тип данн).



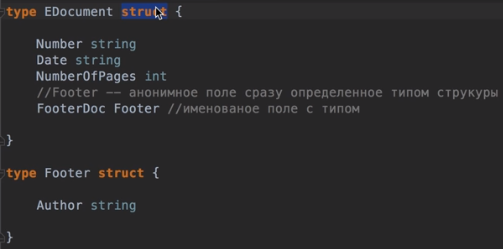
У кажд класса есть методы кот позвол менять сост-ие об-та.Иниц-ем об-т чер коротк нотацию(объявл поля)или чер перем или чер new(type)-созд-ся указ-тель на докум.



Созд-ем метод для типа и созд метод для указ-теля типа-чтобы не тратить память.



….



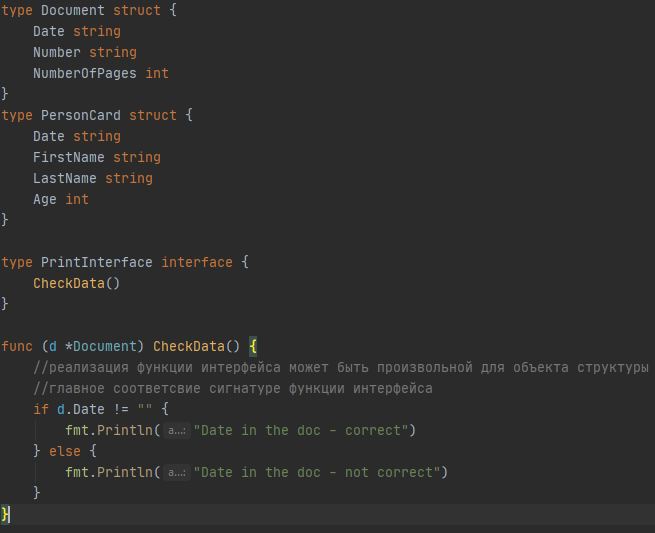
В го классов нет

интрфейс в го это поведенческий контракт, для разных типов. Типы реализующие один и тот-же интерфейс(контракт) необходимо использовать именно в рамках этого контракта, а не рефлексировать сторонними функциями.

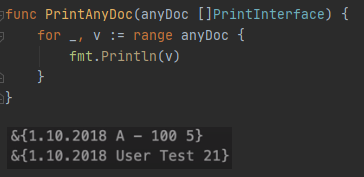
Способность объекта использовать методы... - в го это называется встраивание, кот работает и с интерфейсами и с типами, например: type Reader interface{..} type Writer interface{...} type ReadWriter interface{ Reader Writer}, так вот последний реализует контракты обоих двух первых. Встраивание, позволяет очень сильно абстрагироваться от конкретных реализаций совершенно разных экземпляров типов.

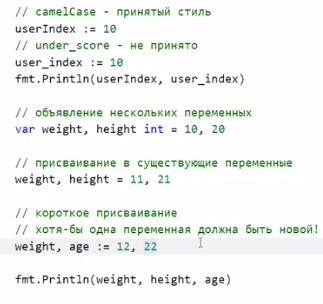
Есть пул горутин, функционал у них разный, но их роднит одно и тоже.

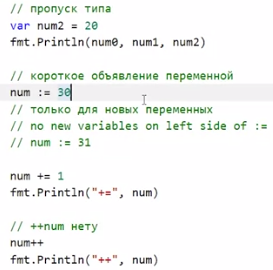
Пример 1



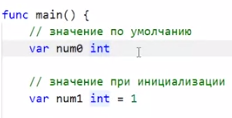




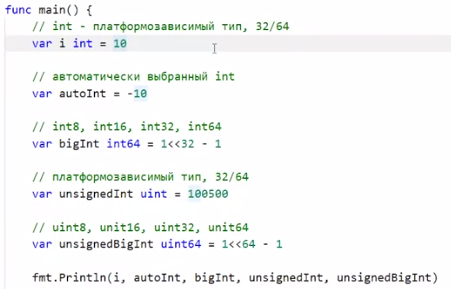


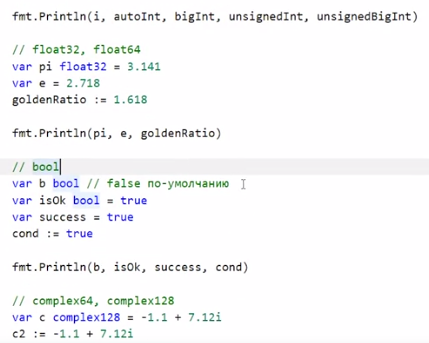
 В инт-фейсах опис-ся повед-ие(набор ф-ий-арг кот приним и кот возвр) чтобы стр-ра нач реализ этот интерф и нельзя указ поля в инт-ф.

М-но не указ перем тогда будет 0. Коротк об-ие перем -если надо что-то верн из ф-ии. Но переприсв знач нельзя =31. Нет префиксн инкремента.Если некорот присв-ие изнач то м-но переопр.

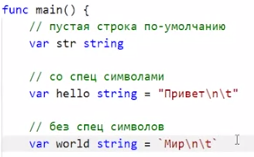


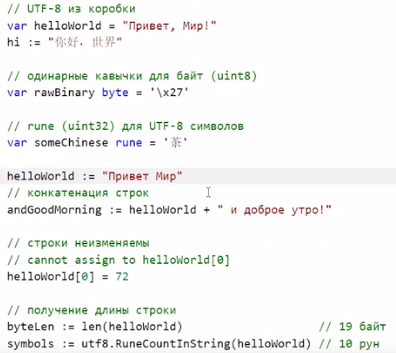
…

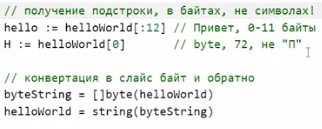




…..

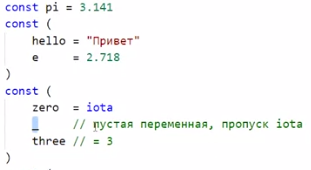






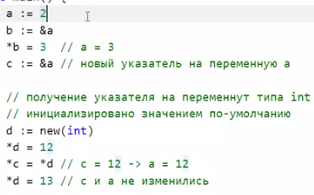
…

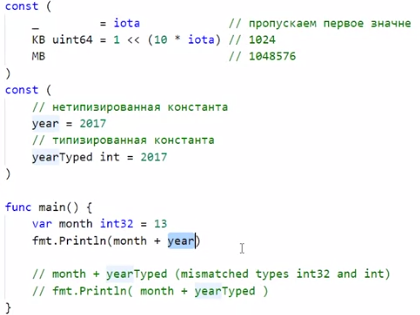
Спец символы внутри кавычек. Йота мож участв в магич выраж. За счет нетипизирован констант-они мог быть больше 64 бит.



…

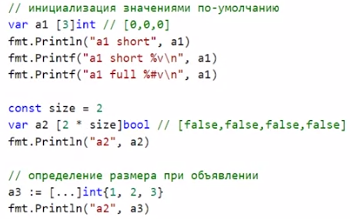
Типы для модел-ия сущностей. В го указ-тель это отд тип данн.В значение кот леж перемен кладем др знач-ие.





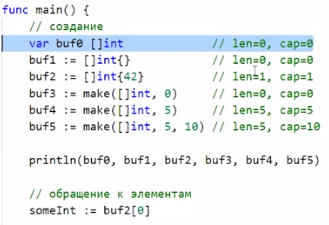
..

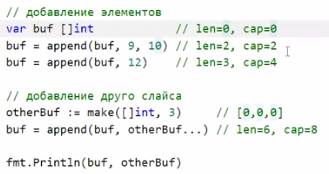
Массив-это данн одн-го типа. Размер массива это часть типа данн. М-но для об-вл размера исп т-ко конст-ту.

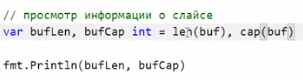


….

У слайса есть длина и кап(ск-ко еще влез)-в 2 раза.







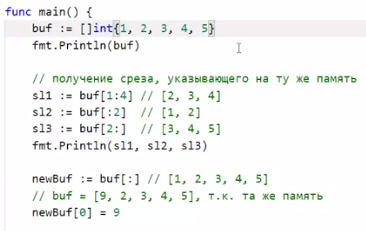
…

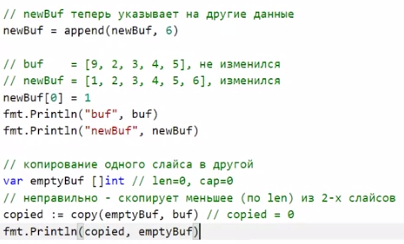
Прав коп-ие слайса



…

Слайс

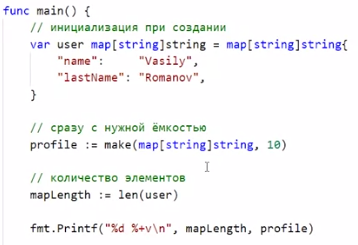


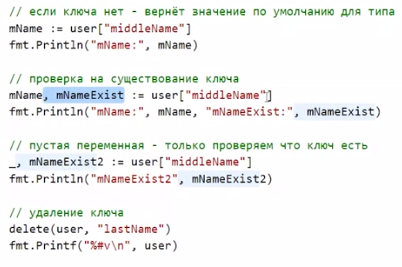


….

Мэп

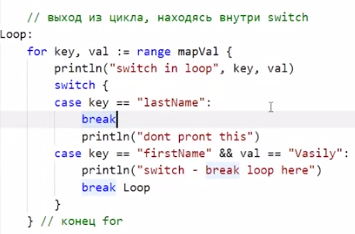
нулев индекс сущ. Мэп позв по кл-чу пол знач быстро.make для нужн емкости.



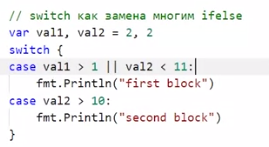


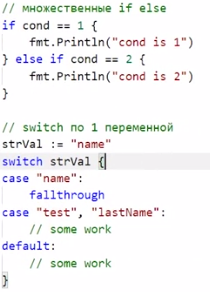
…

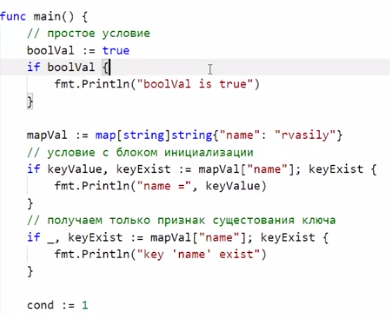
Циклы



Управл-щ стр-ры- провалив-ие в след условие отс-ет, надо писать fallthrough. Для заверш кэйса насередине break.

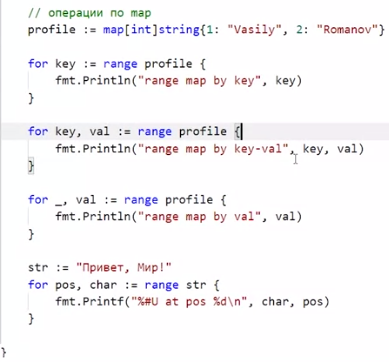




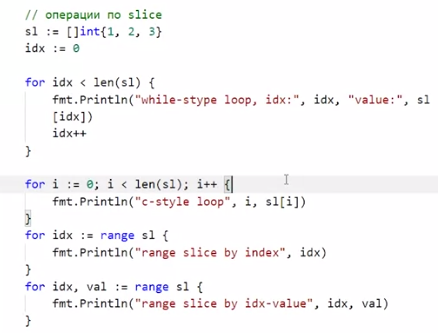


…

В map пор-к ключей неопр-н.

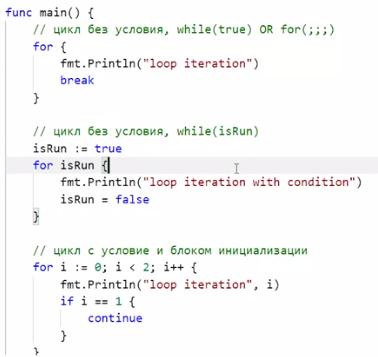


…



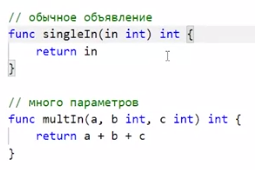
..

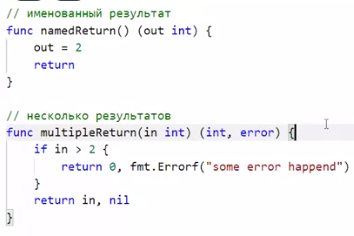
Циклы варианты



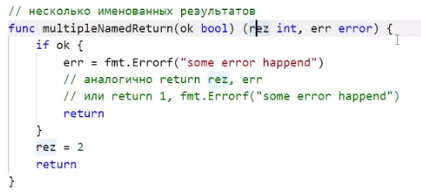
…

Функции

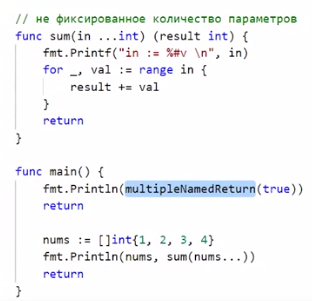




…

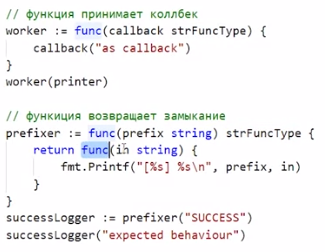


Вариативн ф-ии-приним люб кол пар-ров одн типа.



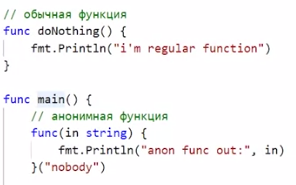
….

Сигн-ра ф-ии это баз тип. Колбэк для вып-ия какой-то ф-ии послезаверш раб-ты. Замык-ие -это ф-ия кот-ая обр-щ-ся к переменнным кот-ые были объевлены вне ее блока-объявл-ия. Зам-ие это когда ф-ия ,кот-ую вернула др-ая ф-ия, напеч-ла рез-т.

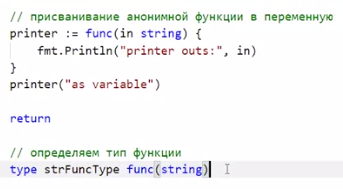


….

Анонимн ф-ия

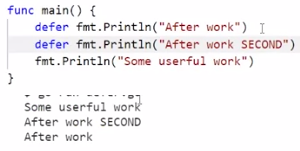


Ф-ия как объект-м-но присв в др перемен,приним ф-ию как арг-нт др ф-ии,возвр ф-ию как рез-т раб-ты др-гой ф-иии.Или исп-ть ф-ию как поле какой-то стр-ры. Анонимной ф-ии перед пар-р в конце. Или обращ-ся к перем-ой как к ф-ии.



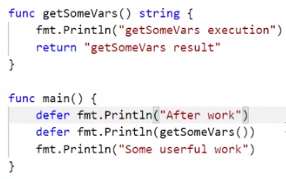
…

Отложен выполн-ие ф-ии -какая-то раб-та вып-ся после заверш-ия ф-ии (посчит время раб-ты ф-ии или закр какой-то ресурс=сетев соед или файлов дескриптор).

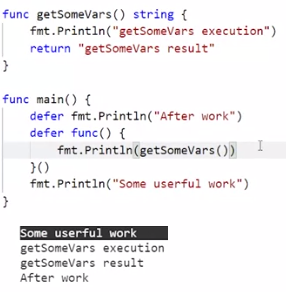


..

Аргументы отложенных функций вычисляются при объявлении блока defer, не когда функция будет вызвана, а при объявлении.



Из мэйн выз-ся втор ф-ия.



….

Defer полезно для обр-б-ки и отлавл-ия паники.Паника-это служебн ф-ия кот остан-ет работу всей прогр-мы. Если получ ошибку-если ошибка вообще была.

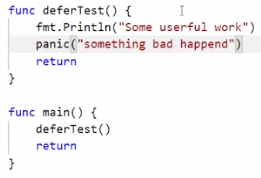


….

Струк-ра

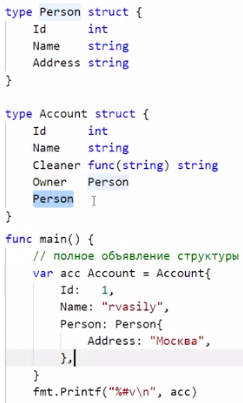


…



…

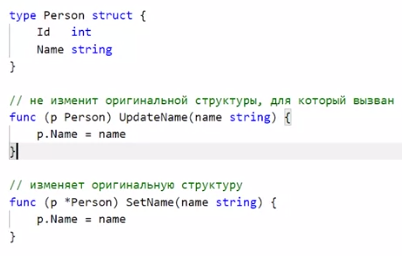
М-но встр-ть одну стр-ру в др-ую-можно обр-щ-ся к полям стр-ры напрямую.



…

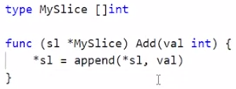
Метод-это ф-ия кот-ая привяз к опр типу данн.

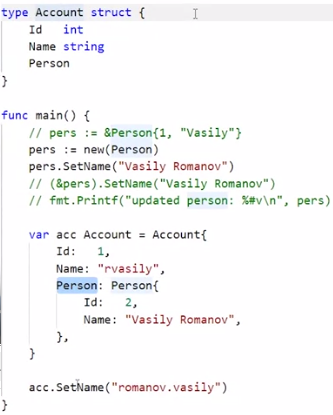
В метод перед-ся копия типа(люб изм-ия в мет-де оригин-льную стр-ру не затронут) или адрес на этот тип. Вызов по указ-телю такой же как new.



…

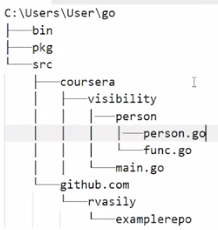
М-но объявл методы у типов и у стр-р.



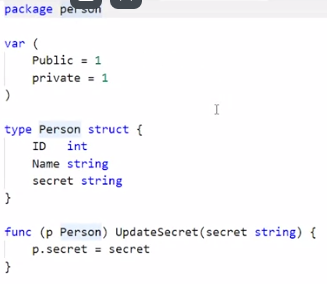


…

папка bin, в нее кладутся собранные бинарники, то есть либо .exe, либо l-файлы. pkg, в pkg кладутся временные объектные файлы, а src непосредственно содержит исходники вашей программы.Если вы вдруг клонируете какой-то пакет, какую-то библиотеку, например, с github, то она также будет копироваться в src.



Если перемен, стр-ра,поле стр-ры, конст-та, ф-ия нач-ся с заглавн буквы, то это экспортируем сущность-принадл-ит 2 пакетам=2 директориям.Если с мал-ой буквы перемен ,то обр-щ-ся к ней т-ко из функций эт пакета.



…

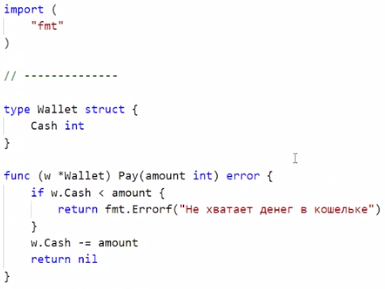
Через интерфейсы в Go реализован полиморфизм, то есть возможность функции принимать в себя аргументы различных типов(утин типизация-стр-ра с методами не знает какому инт-фейсу соотв-ет-передаем метод с интерфейсом).Явн типиз-ия это когда созд класс и явно указ какой интерфейс он реализ-ет.

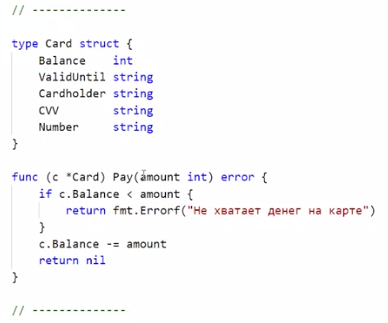
Пример как одна стр-ра реализ-ет инт-фейс

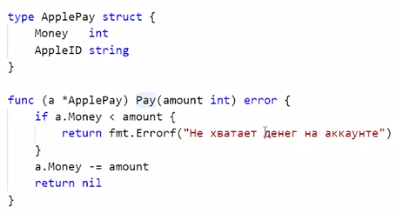


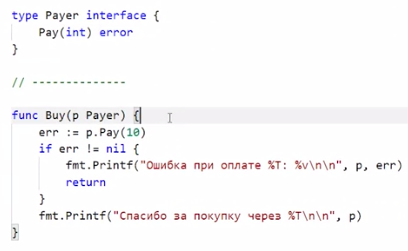
…

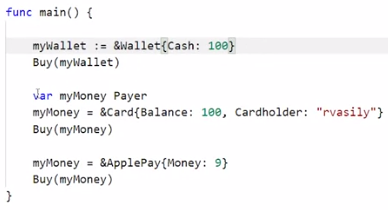
Реализ-ия неск-ми стр-рами инт-фейса





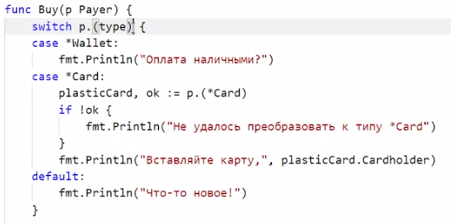






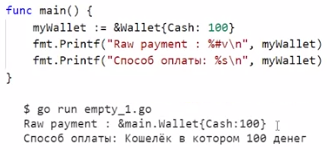
…

Узнать какой тип пришел под инт-фейсом.

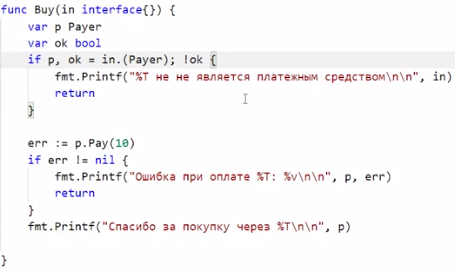


…

Пустой инт-фейс мож принять люб перемен так как не им никаких треб-ий к реализ-ии. Первый формат — это полное Go-шное представление структуры, второй формат — это строка.



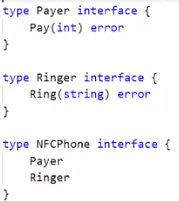
…





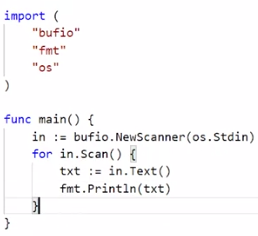
…

Композиция это встраивание интерфейсов. Подобно структурам, когда вы можете вложить, embed, одну структуру в другую, иметь доступ к её полям.



…

«os» для того, чтобы получить доступ к спинартному вводу, «fnt» для того, чтобы делать форматированный вывод, и буферизированный ввод-вывод.



…

Асинхр прогр-ие это когда не строго посл-но, может быть первано др-ми.Самым известным примером асинхронного программирования является технология Ajax — асинхронный JavaScript и XML, когда во время запроса на сервер ваша страница не замораживается, а продолжает работать.Скорость современного процессора гораздо больше, чем скорость оперативной памяти. Для того чтобы как-то компенсировать это, был введен кэш процессора, который располагается вместе с ним на одном кристалле. Например, для того чтобы получить доступ из ядра процессора в основную оперативную память, может затребоваться до 100 наносекунд(переключ-ие констекста- планировщик сохр состояния потоков.Может возникнуть потребность в обращении к основной оперативной памяти, потому что в кэше данных для этого процесса нет(вытесняющая многозад-чность).

Методы обработки запросов в веб-сервере. Итак, начнем мы с технологии cgi-bin. Что это такое? cgi-bin — это когда на каждый запрос п создается новый процесс, нужно довольно много памяти.

Эволюцией этого подхода является worker pool, когда у нас есть некое количество процессов, которые не убиваются после завершения работы, а остаются в ожидании следующего запроса.

Подход мультитрединг- создаем не целый процесс на один запрос или одно соединение, а всего лишь тред.

Тред имеет доступ к памяти своего процесса, переиспользовать какие-то соединения. Тред занимает меньше памяти, но для процессора -можем обработать большее количество запросов.

Можем создать worker pool и обрабатывать запросы, не плодя бесконечно новые треды, а распределять запросы по фиксированному количеству.

Оновное время на современном web api уходит на ожидание запроса от какой-то удаленной базы данных, от какого-то веб-сервиса.

Тред блокируется на ожидание этого ответа и не выполняет никакой другой полезной работы.

Событийный подход к обработке запросов-это неблокирующий ввод-вывод. Когда мы отправили запрос в базу данных, мы продолжаем выполнять какие-то другие запросы, можем обрабатывать много запросов внутри одного треда. Но нельзя выполнять параллельно запросы.

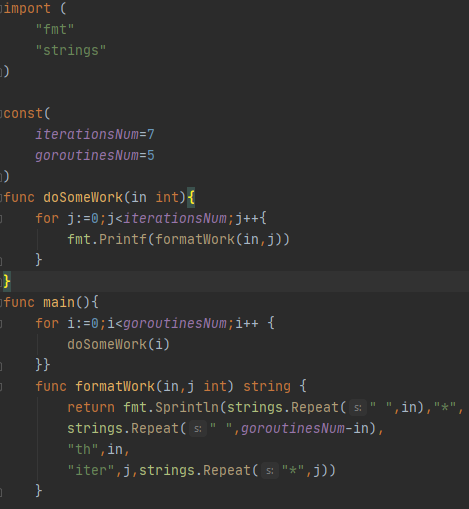
В случае с вытесняющей многозадачностью, когда тред блокируется, какой-то другой тред работает. В случае с кооперативной многозадачностью в этом случае, то мы должны дождаться окончания работы запроса № 1 для того, чтобы выполнять запрос № 2.

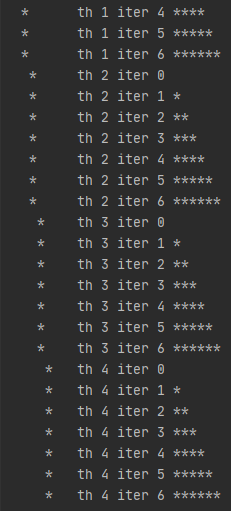
communicating sequential processes- горутина — это аналог сопрограммы, когда в одном системном треде может выполняться несколько горутин, несколько сопрограмм. Как-то разнести это на несколько ядер, для того чтобы, пока один тред занят, мы могли выполнять что-то в другом треде.

Наша горутина может начать выполняться на одном треде, потом уйти в ожидание данных из базы и продолжить выполняться в другом системном треде, потому что первый системный тред занят уже какой-то другой горутиной.

..

Для того чтобы запустить какую-либо функцию внутри отдельной горутины, нужно написать просто слово go. В go есть возможность передать управление другой горутине. Это находится в пакете runtime, функция Goshed(), то есть вызвать планировщик задач.Из горутины вы не можете вернуть никакое значение в основную программу, туда, где была вызвана эта горутина. Для этих целей используются каналы.





Канал создается конструкцией make, после идет ключевое слово chan и тип канала.программа использует несколько ядер процессоров или даже несколько процессоров, состоит в том, что вам нужно как-то их синхронизировать между собой и каким-то образом передавать данные между разными потоками.

я запускаю горутину, которая принимает на вход канал (in — это имя канала, chan int — это тип канала), вычитывает оттуда какое-то значение.

конструкция make (chan int), она делает небуферизированный канал. Что это значит? Это значит, что, когда я кладу в канал какое-то значение, кто-то с другой стороны этого канала должен его прочитать.

И если никто не читает, то горутина блокируется, до тех пор пока на той стороне не появится тот, кто вычтет это значение.deadlock -утечка горутин — когда горутина ждет, но не может дождаться. Буфферизирован каналы мог приним неск знач-ий.