WB Tech: level # 1 (Golang)

# Как делать задания

В заданиях никаких устных решений — только код. Одно решение — один файл с хорошо откомментированным кодом. Каждое решение или невозможность решения надо объяснить.

Разрешается и приветствуется использование любых справочных ресурсов, привлечение сторонних экспертов и т.д. и т.п.

Основной критерий оценки — четкое понимание «как это работает». Некоторые задачи можно решить несколькими способами, в этом случае требуется привести максимально возможное количество вариантов.

Можно задавать вопросы, как по условию задач, так и об их решении. Идеальный вариант — продемонстрировать свои решения и получить максимальный фидбэк от опытных разработчиков Wildberries.

# Задания

1. Дана структура Human (с произвольным набором полей и методов). Реализовать встраивание методов в структуре Action от родительской структуры Human (аналог наследования).
2. Написать программу, которая конкурентно рассчитает значение квадратов чисел взятых из массива (2,4,6,8,10) и выведет их квадраты в stdout.
3. Дана последовательность чисел: 2,4,6,8,10. Найти сумму их квадратов(22+32+42….) с использованием конкурентных вычислений.
4. Реализовать постоянную запись данных в канал (главный поток). Реализовать набор из N воркеров, которые читают произвольные данные из канала и выводят в stdout. Необходима возможность выбора количества воркеров при старте.  
     
   Программа должна завершаться по нажатию Ctrl+C. Выбрать и обосновать способ завершения работы всех воркеров.
5. Разработать программу, которая будет последовательно отправлять значения в канал, а с другой стороны канала — читать. По истечению N секунд программа должна завершаться.
6. Реализовать все возможные способы остановки выполнения горутины.
7. Реализовать конкурентную запись данных в map.
8. Дана переменная int64. Разработать программу которая устанавливает i-й бит в 1 или 0.
9. Разработать конвейер чисел. Даны два канала: в первый пишутся числа (x) из массива, во второй — результат операции x\*2, после чего данные из второго канала должны выводиться в stdout.
10. Дана последовательность температурных колебаний: -25.4, -27.0 13.0, 19.0, 15.5, 24.5, -21.0, 32.5. Объединить данные значения в группы с шагом в 10 градусов. Последовательность в подмножноствах не важна.

Пример: -20:{-25.0, -27.0, -21.0}, 10:{13.0, 19.0, 15.5}, 20: {24.5}, etc.

1. Реализовать пересечение двух неупорядоченных множеств.
2. Имеется последовательность строк - (cat, cat, dog, cat, tree) создать для нее собственное множество.
3. Поменять местами два числа без создания временной переменной.
4. Разработать программу, которая в рантайме способна определить тип переменной: int, string, bool, channel из переменной типа interface{}.
5. К каким негативным последствиям может привести данный фрагмент кода, и как это исправить? Приведите корректный пример реализации.

**var** justString string

**func** someFunc() {

v := createHugeString(1 << 10)

justString = v[:100]

}

**func** main() {

someFunc()

}

1. Реализовать быструю сортировку массива (quicksort) встроенными методами языка.
2. Реализовать бинарный поиск встроенными методами языка.
3. Реализовать структуру-счетчик, которая будет инкрементироваться в конкурентной среде. По завершению программа должна выводить итоговое значение счетчика.
4. Разработать программу, которая переворачивает подаваемую на ход строку (например: «главрыба — абырвалг»). Символы могут быть unicode.
5. Разработать программу, которая переворачивает слова в строке.   
   Пример: «snow dog sun — sun dog snow».
6. Реализовать паттерн «адаптер» на любом примере.
7. Разработать программу, которая перемножает, делит, складывает, вычитает две числовых переменных a,b, значение которых > 2^20.
8. Удалить i-ый элемент из слайса.
9. Разработать программу нахождения расстояния между двумя точками, которые представлены в виде структуры Point с инкапсулированными параметрами x,y и конструктором.
10. Реализовать собственную функцию sleep.
11. Разработать программу, которая проверяет, что все символы в строке уникальные (true — если уникальные, false etc). Функция проверки должна быть регистронезависимой.

Например:   
abcd — true

abCdefAaf — false  
 aabcd — false

# Устные вопросы

1. Какой самый эффективный способ конкатенации строк?

Strings.Join (https://andrey.4devs.io/ru/golang/string-concatenation-golang.html)

1. Что такое интерфейсы, как они применяются в Go?

Интерфейс – тип (или класс), который определяет действия, которые должны быть реализованы у наследователей.

Интефрейсы используются для:

1. Уменьшения количества кода
2. Для того, чтобы придерживаться чистой архитектуры – с помощью интерфейса клиент может обращаться к программе. То есть, мы знаем, что должна делать программа, но не должны знать как именно.
3. Чем отличаются RWMutex от Mutex?

Концептуально RWMutex похож на Mutex, но RWMutex добавляет возможность параллельного чтения из канала.

1. Чем отличаются буферизированные и не буферизированные каналы?

Буферизированный канал имеет размер буффера, тогда как в небуферезированном канале он по умолчанию равен 0. Когда мы работаем с небуферезированным каналом, то, что мы записываем в него, сразу становится доступным для чтения. Это значит, что если мы ничего не записываем в канал, горутина, читсающая из канала, заблокирована. При записи в буферезированный канал, мы будем записывать значения, пока не запишем N значений, а после того, как канал заполнится, горутина, пишущая в канал, будем ждать освобождения места. А горутина, которая читает из канала, не будет заблокирована, пока не считает все значения из канала. Таким образом, если мы ничего не записываем, горутина не блокируется, пока не прочитает до конца.

1. Какой размер у структуры struct{}{}?

0

1. Есть ли в Go перегрузка методов или операторов?

Перегрузки операторов нет, операторы работают строго и выполняют операции только с определенными типами данных аргументов.

Перегрузки методов тоже нет, можно лишь менять названия или использовать интерфейсы

1. В какой последовательности будут выведены элементы map[int]int?  
     
   *Пример:*

m[0]=1

m[1]=124

m[2]=281

По возрастанию ключей

1. В чем разница make и new?

Make используется для инициализации слайса, мапа и канала и возвращает тип, в то время как new возвращает указатель.

1. Сколько существует способов задать переменную типа slice или map?

Slice:

1) x := make([]float64, 5)

2) arr := [5]float64{1,2,3,4,5}

x := arr[0:5]

3) slice1 := []int{1,2,3}

Map:

1) x := make(map[string]int)

2) elements := map[string]int{

“g”: 1

}

1. Что выведет данная программа и почему?

**func** update(p \*int) {

b := 2

p = &b

}

**func** main() {

**var** (

a = 1

p = &a

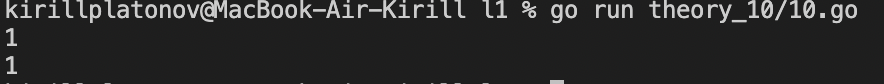
)

fmt.Println(\*p)

update(p)

fmt.Println(\*p)

}



Потому что мы не возвращаем значение p из функции update

1. Что выведет данная программа и почему?

**func** main() {

wg := sync.WaitGroup{}

**for** i := 0; i < 5; i++ {

wg.Add(1)

**go func**(wg sync.WaitGroup, i int) {

fmt.Println(i)

wg.Done()

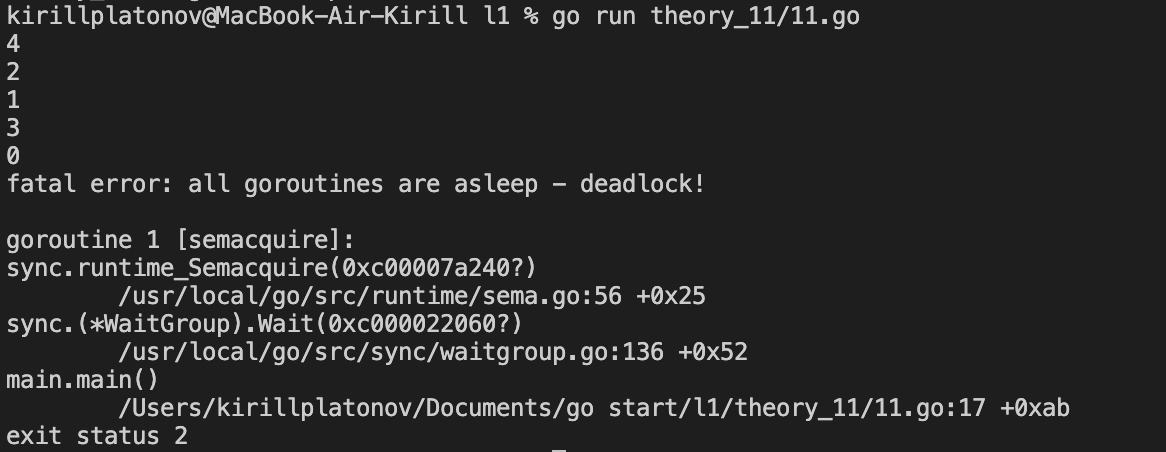
}(wg, i)

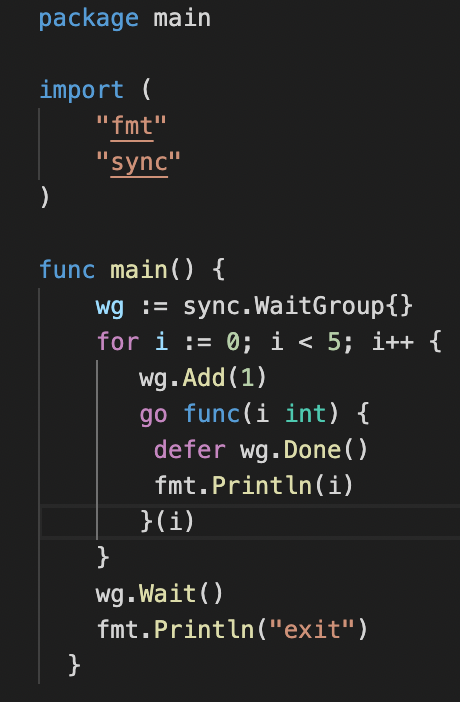
}

wg.Wait()

fmt.Println(**"exit"**)

}





Потому что мы передали waitgroup по значению и у нас в функции создался новый инстанс вг.

1. Что выведет данная программа и почему?

**func** main() {

n := 0

**if *true*** {

n := 1

n++

}

fmt.Println(n)

}  


Потому что эти переменные n хранятся по разным адресам, так как мы заново инициализировали переменную в блоке иф и она находится в его зоне видимости.

1. Что выведет данная программа и почему?

**func** someAction(v []int8, b int8) {

v[0] = 100

v = append(v, b)

}

**func** main() {

**var** a = []int8{1, 2, 3, 4, 5}

someAction(a, 6)

fmt.Println(a)

}



6 не выводится, так как мы сделали аппенд. (14 задание)

1. Что выведет данная программа и почему?

**func** main() {

slice := []string{**"a"**, **"a"**}

**func**(slice []string) {

slice = append(slice, **"a"**)

slice[0] = **"b"**

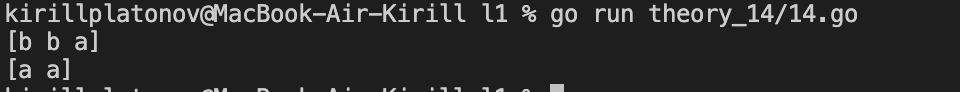
slice[1] = **"b"**

fmt.Print(slice)

}(slice)

fmt.Print(slice)

}



Когда мы делаем append, нам нужно увеличить длину родительского массива, на которй ссылается изначальный слайс, у которого лен=2 и кэп=2.Но увеличивать длину массива мы не можем, можем только создать новый. И аппенд создаст новый массив, увеличит кэп в 2 раза и добавит новый элемент. И теперь слайс будет указывать на новый массив.