

# 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 рассчитывает 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( $2^2+3^2+4^2+...$ ) 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 подмножествах не важна.

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

11. Реализовать пересечение двух неупорядоченных множеств.
12. Имеется последовательность строк - (cat, cat, dog, cat, tree) создать для нее собственное множество.
13. Поменять местами два числа без создания временной переменной.
14. Разработать программу, которая в рантайме способна определить тип переменной: int, string, bool, channel из переменной типа interface{}
15. 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()
}
```

16. Реализовать быструю сортировку массива (quicksort) встроенными методами языка.
17. Реализовать бинарный поиск встроенными методами языка.
18. Реализовать структуру-счетчик, которая будет инкрементироваться в конкурентной среде. По завершению программа должна выводить итоговое

значение счетчика.

19. Разработать программу, которая переворачивает подаваемую на ход строку (например: «главрыба — абырвалг»). Символы могут быть unicode.
20. Разработать программу, которая переворачивает слова в строке.  
Пример: «snow dog sun — sun dog snow».
21. Реализовать паттерн «адаптер» на любом примере.
22. Разработать программу, которая перемножает, делит, складывает, вычитает две числовых переменных  $a, b$ , значение которых  $> 2^{20}$ .
23. Удалить  $i$ -ый элемент из слайса.
24. Разработать программу нахождения расстояния между двумя точками, которые представлены в виде структуры `Point` с инкапсулированными параметрами  $x, y$  и конструктором.
25. Реализовать собственную функцию `sleep`.
26. 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?
2. Что такое интерфейсы, как они применяются в Go?
3. Чем отличаются `RWMutex` от `Mutex`?
4. Чем отличаются буферизированные и не буферизированные каналы?
5. Какой размер у структуры `struct{}`?
6. Есть ли в Go перегрузка методов или операторов?

7. В какой последовательности будут выведены элементы map[int]int?

*Пример:*

```
m[0]=1  
m[1]=124  
m[2]=281
```

8. В чем разница make и new?
9. Сколько существует способов задать переменную типа slice или map?
10. 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)  
}
```

11. 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")  
}
```

12. Что выведет данная программа и почему?

```
func main() {  
    n := 0  
    if true {
```

```

    n := 1
    n++
}
fmt.Println(n)
}

```

13. 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)
}

```

14. 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)
}

```