WB Tech: level # 1 (Golang)

# Как делать задания

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.

# Задания

1. Реализовать композицию структуры Action от родительской структуры Human.
2. Написать программу, которая конкурентно рассчитает значение квадратов значений взятых из массива (2,4,6,8,10) и выведет их квадраты в stdout.
3. Дана последовательность чисел (2,4,6,8,10) найти их сумму квадратов(22+32+42….) с использованием конкурентных вычислений.
4. Реализовать набор из N воркеров, которые читают из канала произвольные данные и выводят в stdout. Данные в канал пишутся из главного потока. Необходима возможность выбора кол-во воркеров при старте, а также способ завершения работы всех воркеров.
5. Написать программу, которая будет последовательно писать значения в канал, а с другой стороны канала — читать. По истечению N секунд программа должна завершиться.
6. Какие существуют способы остановить выполнения горутины? Написать примеры использования.
7. Реализовать конкурентную запись в map.
8. Дана переменная int64. Написать программу которая устанавливает i-й бит в 1 или 0.
9. Написать конвейер чисел. Даны 2 канала - в первый пишутся числа из массива, во второй пишется результат операции 2\*x, после чего данные выводятся в stdout.
10. Дана последовательность температурных колебаний (-25.4, -27.0 13.0, 19.0, 15.5, 24.5, -21.0, 32.5). Объединить данный значения в группы с шагов в 10 градусов. Последовательность в подмножностве не важна.

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

1. Написать пересечение двух неупорядоченных массивов.
2. Что выводит 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)

}

1. Чем завершится данная программа?

**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"**)

}

1. Имеется последовательность строк - (cat, cat, dog, cat, tree) создать для нее собственное множество.
2. Поменять местами два числа без создания временной переменной.
3. Что выведет программа данная программа?

**func** main() {

n := 0

**if *true*** {

n := 1

n++

}

fmt.Println(n)

}

1. Написать программу, которая в рантайме способна определить тип переменной — int, string, bool, channel из переменной типа interface{}.
2. 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)

}

1. К каким негативным последствиям может привести данный кусок кода и как это исправить?

**var** justString string

**func** someFunc() {

v := createHugeString(1 << 10)

justString = v[:100]

}

**func** main() {

someFunc()

}

1. Какой результат выполнения данного кода 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)

}

1. Написать программу, которая в конкурентном виде читает элементы из массива в stdout.
2. Написать быструю сортировку встроенными методами языка.
3. Написать бинарный поиск встроенными методами языка.
4. Создать слайс с предварительно выделенными 100 элементами.
5. Написать свою структуру счетчик, которая будет инкрементировать и выводить значения в конкурентной среде.
6. Написать программу, которая переворачивает строку. Символы могут быть unicode.
7. Написать программу, которая переворачивает слова в строке (snow dog sun - sun dog snow).
8. Реализовать паттерн адаптер на любом примере.
9. Написать программу, которая перемножает, делит, складывает, вычитает 2 числовых переменных a,b, значение которые > 2^20.
10. Удалить i-ый элемент из слайса.
11. Написать программу нахождения расстояния между 2 точками, которые представление в виде структуры Point с инкапсулированными параметрами x,y и конструктором.
12. Написать собственную функцию Sleep.
13. Даны 2 канала - в первый пишутся рандомные числа после чего они проверяются на четность и отправляются во второй канал. Результаты работы из второго канала пишутся в stdout.
14. Написать программу, которая проверяет, что все символы в строке уникальные.

# Устные вопросы

1. Какой самый эффективный способ работы с объединением 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 пустой структуры struct{}{}?
6. Какой способ определения двух слайсов предпочтительнее?

**var** a []int

a := []int{}

1. Есть ли в Go перегрузка методов или операторов?
2. В какой последовательности будут выведены элементы map[int]int?  
     
   *Пример:*

m[0]=1

m[1]=124

m[2]=281

1. В чем разница make и new?
2. Сколько существует способов создать переменную типа slice или map?