

Добрый день. И всем добро пожаловать в 6 модуль лекций по языку Golang. И в первый блок этого модуля «Go modules и структура проекта». О чем же я сегодня расскажу? Итак, я расскажу сегодня: как организовывается ваш код; что такое модуль и его структура; для чего нужны переменные `goroot`, `gorath` и `goroxu`; посмотрим на то, как модулизируется проект, на примере маленького реального проекта, и вообще поймем, как работает с зависимостями в golang-е. Поехали. Какие же принципы существуют у нас в организации кода? Давайте начнем с этого.

Итак, первый принцип. Вот мы пишем функцию. Наша функция должна делать что-то одно, то есть что в ней написано, то она и должна делать. Если у нас функция сложить 2 числа, она должна складывать 2 числа, не умножать, не делить, не делать какой-то там дискриминант. Она должна делать именно это, выполнять 1 свою функцию. Ну как бы из названия понятно. Если у нас метод погавкать у собачки, собачка должна делать «гав-гав». Все, больше никаких других эффектов у нее быть не должно. Соответственно после того, как мы пишем такие функции, у нас может появиться какой-то объект, с которым мы работаем. Ну так называемый класс, да, то есть, пользуясь объектно-ориентированным программированием, то значит у нас появляется объект класса, у которого какие-то свои функции. И все эти функции должны относиться к самому объекту и делать что-то с ним связанное. Если у вас есть кошечка, она не должна летать, как белка-летяга. Если у вас есть модуль, который, вернее у вас есть объект, который отвечает за склеивание 2 строк, и он не должен делать перемножение 2 чисел, ну и так далее. Понятно.

И, соответственно, весь go состоит из такого рода скомпонованных компонентов, ну, в принципе, это не новая какая-то вещь. Точно также работает Java, Python и прочие объектно-ориентированные языки. И, соответственно, это все структурируется определенным образом. Каким образом?

Для организации кода, у нас существуют 3 структуры. Соответственно, Source file – исходный файл, файл исходников. Просто в нем написаны ваши какие-то ваши функции, ваши структуры, над которыми эти функции выполняются. Ну то есть как-то объединено это единым назначением. Это 1 объект или какой-то просто набор функций, который выполняет операцию над чем-то одним. И множество таких исходных файлов объединяются у нас в Package, то есть, в так называемый, пакет. И этот Package написан в начале каждого Source file. Это обязательно. Вы должны всегда указывать имя этого самого Package, и по умолчанию программа Golang начинается с Package main, которая является главной точкой входа в приложение. Ну и, соответственно, все это вместе, все эти Package-и, объединенные вместе, являются модулем, то есть выглядит это как-то так. Package – это просто хранилище исходных файлов. Соответственно, Module – это какое-то хранилище Package-ей.