IT BOARDING

BOOTCAMP

Programación Go

Punteros y Canales en Go

// Práctica clase 3 - Go Bases

Objetivo

El objetivo de esta guía práctica es que podamos afianzar los conceptos sobre punteros y canales vistos en el módulo de Go Bases. Para esto vamos a plantear una serie de ejercicios simples e incrementales (ya que vamos a ir trabajando y agregando complejidad a lo que tenemos que construir), lo que nos permitirá repasar los temas que estudiamos.

Forma de trabajo

Los ejercicios deben ser realizados en sus computadoras. Les recordamos que generen una carpeta para cada clase y ahí dentro tengan un archivo .go para cada ejercicio.

¿Are you ready? 😎👍

Ejercicio 1 - Red social

Una empresa de redes sociales requiere implementar una estructura usuario con funciones que vayan agregando información a la estructura. Para optimizar y ahorrar memoria requieren que la estructura usuarios ocupe el mismo lugar en memoria para el main del programa y para las funciones:

La estructura debe tener los campos: Nombre, Apellido, edad, correo y contraseña

Y deben implementarse las funciones:

- cambiar nombre: me permite cambiar el nombre y apellido.
- cambiar edad: me permite cambiar la edad.
- cambiar correo: me permite cambiar el correo.
- cambiar contraseña: me permite cambiar la contraseña.

Ejercicio 2 - Ecommerce

Una importante empresa de ventas web necesita agregar una funcionalidad para agregar productos a los usuarios. Para ello requieren que tanto los usuarios como los productos tengan la misma dirección de memoria en el main del programa como en las funciones.

Se necesitan las estructuras:

- Usuario: Nombre, Apellido, Correo, Productos (array de productos).
- Producto: Nombre, precio, cantidad.

Se requieren las funciones:

- Nuevo producto: recibe nombre y precio, y retorna un producto.
- Agregar producto: recibe usuario, producto y cantidad, no retorna nada, agrega el producto al usuario.
- Borrar productos: recibe un usuario, borra los productos del usuario.

Ejercicio 3 - Calcular Precio

Una empresa nacional se encarga de realizar venta de productos, servicios y mantenimiento. Para ello requieren realizar un programa que se encargue de calcular el precio total de Productos, Servicios y Mantenimientos. Debido a la fuerte demanda y para optimizar la velocidad requieren que el cálculo de la sumatoria se realice en paralelo mediante 3 go routines.

Se requieren 3 estructuras:

- Productos: nombre, precio, cantidad.
- Servicios: nombre, precio, minutos trabajados.
- Mantenimiento: nombre, precio.

Se requieren 3 funciones:

- Sumar Productos: recibe un array de producto y devuelve el precio total (precio * cantidad).
- Sumar Servicios: recibe un array de servicio y devuelve el precio total (precio * media hora trabajada, si no llega a trabajar 30 minutos se le cobra como si hubiese trabajado media hora).
- Sumar Mantenimiento: recibe un array de mantenimiento y devuelve el precio total.

Los 3 se deben ejecutar concurrentemente y al final se debe mostrar por pantalla el monto final (sumando el total de los 3).

Ejercicio 4 - Ordenamiento

Una empresa de sistemas requiere analizar qué algoritmos de ordenamiento utilizar para sus servicios.

Para ellos se requiere instanciar 3 arreglos con valores aleatorios desordenados

- un arreglo de números enteros con 100 valores
- un arreglo de números enteros con 1000 valores
- un arreglo de números enteros con 10000 valores

Para instanciar las variables utilizar rand

```
package main

import (
   "math/rand"
)



func main() {
   variable1 := rand.Perm(100)
   variable2 := rand.Perm(1000)
   variable3 := rand.Perm(10000)
}
```

Se debe realizar el ordenamiento de cada una por:

- Ordenamiento por inserción
- Ordenamiento por burbuja
- Ordenamiento por selección

Una go routine por cada ejecución de ordenamiento

Debo esperar a que terminen los ordenamientos de 100 números para seguir el de 1000 y después el de 10000.

Por último debo medir el tiempo de cada uno y mostrar en pantalla el resultado, para saber qué ordenamiento fue mejor para cada arreglo