

- 1) Golang es un lenguaje de programación compilado, concurrente y open source creado por Google en 2009. Su objetivo es ser un lenguaje simple, práctico y eficiente. Su sintaxis está basada en C.

- a) Golang posee alcance estático, por lo que el alcance de una variable está determinado por el bloque en el cual es declarado y solo pueden ser accedidas en dicho ámbito.

Go es un lenguaje orientado al desarrollo de sistemas y aplicaciones web, por lo que al tener alcance estático, los desarrolladores pueden ver fácilmente dónde se declaran las variables y cómo se relacionan entre sí. Esto tiene la finalidad de evitar errores y problemas de seguridad, que son importantes para los desarrolladores de sistemas y aplicaciones web.

La asociación profunda permite que compartir variables sea más sencillo y aumenta el rendimiento.

El alcance estático y la asociación profunda hacen que el código sea más fácil de mantener.

Por otro lado, estas elecciones de diseño generan ciertos problemas generales, como mayor uso de memoria (por la asociación profunda) lo cual puede limitar el diseño de páginas web dependiendo del tipo de servidor donde estas sean alojadas.

- b) Go tiene un sistema de módulos que permite organizarlos en paquetes y ser compartidos. Los módulos están definidos en un archivo `go.mod` que se encuentra en la root de un proyecto.

Estos pueden ser públicos o privados.

Los módulos Go se importan utilizando la palabra clave **import**. La declaración de importación especifica el nombre del módulo que se desea importar, así como los nombres de los paquetes que se desean importar del módulo.

Los nombres de los paquetes se exportan utilizando la palabra clave **export**. La declaración de exportación especifica los nombres de las variables, funciones y tipos que se desean exportar del paquete.

- c)
 - i) Go ofrece la posibilidad de crear alias.

- Ejemplo de alias:

```
type test int
```

```
var a int  
var b test
```

```
a = 10
```

```
b = a // b y a ahora se refieren al mismo valor.
```

- ii) Go no soporta la sobrecarga directamente, sin embargo, se pueden crear funciones con nombres similares que funcionen con diferentes tipos de parámetros. Esto se conoce como coincidente.

- Por ejemplo:

```
func concatenar(s1 string, s2 string) string {  
    return s1 + s2  
}
```

```
func concatenar(i1 int, i2 int) int {  
    return i1 + i2  
}
```

En este ejemplo, hay dos funciones llamadas concatenar. La primera función concatena dos cadenas, mientras que la segunda función suma dos enteros.

- iii) Por último, el polimorfismo se puede lograr mediante el uso de interfaces. Las interfaces son colecciones de métodos.

- Ejemplo:

```
type Printable interface {  
    Print()  
}
```

```
type Stringer interface {  
    String() string  
}
```

```
type MyString string
```

```

func (s MyString) Print() {
    fmt.Println(s)
}

func (s MyString) String() string {
    return string(s)
}

func main() {
    var p Printable = MyString("Hello, world!")
    p.Print() // Prints "Hello, world!"
}

```

En este ejemplo, la interfaz Printable define un método Print(). La interfaz Stringer define un método String().

La clase MyString implementa ambas interfaces. La función Print() puede imprimir cualquier objeto que implemente la interfaz Printable. En este ejemplo, Print() se utiliza para imprimir un objeto MyString.

- d) **Compilador:** Golang posee un compilador que produce eficientemente código binario y es de un solo paso, pues no requiere paso de compilación intermedio.

Intérprete: No hay un intérprete oficial de Go, pero hay varios intérpretes de Go de terceros disponibles.

Debugger: Go posee un debugger integrado que permite depurar código Go. El debugger proporciona una variedad de funciones, incluyendo la capacidad de establecer puntos de interrupción, inspeccionar variables y ejecutar código paso a paso.

Profiler: El perfilador de Go proporciona varios datos, incluyendo el tiempo de ejecución de las funciones, el uso de la memoria y el uso de la CPU.

Frameworks: tiene gran variedad de frameworks disponibles, incluyendo frameworks para desarrollo web, desarrollo de aplicaciones móviles y desarrollo de aplicaciones de escritorio.