

GO

a)

i.

Secuenciación: En Go se denota una línea de ejecución con un punto y coma “;”, o implícitamente mediante un salto de línea

Selección simple: Bloques condicionales. Se pueden hacer declaraciones antes de los condicionales; todas las variables declaradas ahí pueden ser utilizadas en las ramas del bloque. No hay un operador ternario en Go por lo que debe usarse un bloque if entero para condiciones de asignación básicas.

Selección múltiple: Hay bloques switch case que permiten expresar condicionales para varios valores. Se pueden evaluar varias expresiones en un mismo case separándolas por comas. También hay un type switch que permite evaluar tipos de un dato en lugar de su valor

Repetición: En Go hay un ciclo for en el que se puede indicar una variable de iteración, la condición de salida, y la operación sobre la variable de iteración antes de la siguiente iteración. No hay una palabra dedicada para un ciclo while, en su lugar se puede hacer un for en el cual únicamente se indica la condición de salida.

Abstracción procedural: Las funciones en Go se declaran con la palabra reservada “func”, seguida de un paréntesis donde se pueden declarar parámetros. Hay funciones variádicas que pueden recibir una lista de tamaño indeterminado de variables del mismo tipo. Soporta también funciones anónimas y clausuras. Es posible implementar recursión llamando a una función dentro de sí misma

Concurrencia: Go implementa concurrencia mediante las goroutines. Para ejecutar una función f() en una goroutine, se llama go f(), y la función f se ejecutará de manera concurrente con la función llamadora. Los canales son pipes que permiten que se pueda enviar y recibir información entre dos goroutines.

Excepciones: En Go no hay excepciones como en otros lenguajes. En su lugar hay una interfaz llamada error, y se comunican los errores mediante el retorno explícito de un valor error por las funciones, y son manejados por el control de flujo usado por cualquier otro proceso normal.

ii.

A la hora de evaluar los operandos de una expresión, asignación, o sentencias de retorno, todas las llamadas a funciones, llamadas a métodos, y operaciones de comunicación de canales, son evaluadas en orden léxico de izquierda a derecha

iii.

Números: Go ofrece números enteros con y sin signo de 8, 16, 32 y 64 bits; y números flotantes IEEE 754 de 32 y 64 bits. También ofrece los tipos complex que representan números complejos mediante partes reales e imaginarias de 32 o 64 bit ambos.

Booleanos: Se representan como un bit de información que denota true o false. No se pueden convertir implícitamente a ningún otro tipo

Arreglos: Los arreglos en GO se implementan de manera que su tamaño es parte de su declaración, es decir son de tamaño estático

Slice: El slice es una interfaz que funciona como un arreglo de tamaño mutable, y sobre estos se pueden realizar varias operaciones como append y copy

Strings: Los strings en Go son slices de bytes de solo lectura, cada byte representando un carácter, llamado Rune en Go, encodeado en UTF-8

Maps: Estructura de datos asociativa, como los diccionarios o hashes en otros lenguajes

Structs: Colecciones de campos tipados de datos. Go soporta la creación de structs anidados. También soporta definir métodos para los structs, y la creación de interfaces para especificar la implementación de métodos para un struct