



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

Actividad:

Sentencias de Block

Estudiante:

Ángel Amaury Tienda Lezama

Grupo:

9B

Materia:

Web

Docente:

Ray Brunett Parra Galaviz

Tijuana, Baja California, 02 de Octubre del 2024

Las sentencias de iteración, también conocidas como bucles, son estructuras de control utilizadas para repetir un bloque de instrucciones en un programa. Estas sentencias permiten

ejecutar un conjunto de acciones mientras se cumpla una condición específica. Al ser una forma de control condicional, las iteraciones se repiten siempre que se satisfaga la condición indicada. En muchos casos, es necesario repetir un número finito o infinito de veces un conjunto de instrucciones, y para ello se utilizan estos ciclos. En C++, los bucles se implementan a través de las sentencias `for`, `while` y `do-while`. La sentencia `for` es particularmente útil cuando conocemos de antemano la cantidad de veces que se debe ejecutar el bloque de instrucciones. Por otro lado, la sentencia `while` es más adecuada para situaciones donde el número de repeticiones no es conocido desde el principio.

Ejemplo:

```
fun main() {  
    var suma = 0  
    var num = 1  
  
    while (num <= 10) {  
        suma += num++  
    }  
  
    println("La sumatoria es: $suma")  
}
```

Sentencias de control

Las sentencias de control son instrucciones que permiten modificar el flujo normal de un programa, permitiendo que el código tome decisiones o repita acciones basadas en ciertas condiciones. Estas sentencias se dividen en dos categorías principales:

● **Sentencias condicionales:** Se utilizan para ejecutar un bloque de código si una condición es verdadera o ejecutar otro bloque si es falsa. Algunos ejemplos incluyen: ○

`if / else`: Evalúa una condición y ejecuta un bloque de código si es verdadera; en caso contrario, ejecuta un bloque alternativo.

○ `switch`: Evalúa una expresión y ejecuta el código correspondiente a uno de los casos predefinidos.

● **Sentencias de bucle:** Se utilizan para repetir un bloque de código múltiples veces mientras se cumpla una condición o durante un número determinado de repeticiones.

Algunos ejemplos son:

○ `for`: Repite un bloque de código un número específico de veces.

○ `while`: Ejecuta el bloque de código mientras una condición se mantenga verdadera.

○ `do-while`: Ejecuta el bloque de código al menos una vez y luego verifica la condición antes de repetir.

Ejemplo:

```
fun main() {  
  
    val semaforo = "Verde"  
  
  
    if (semaforo == "Rojo") {  
        println("Detente")  
  
    } else if (semaforo == "Verde") {
```

```
        println("Avanza")

    } else {

        println("Precaución")

    }

}
```

Funciones

Una función es un conjunto de instrucciones que realizan una tarea específica, encapsuladas bajo un nombre para su reutilización. Cuando se necesita ejecutar el conjunto de instrucciones definido en una función, se invoca su nombre en el flujo del programa, pasándole los argumentos correspondientes. Esto permite organizar el código y evitar repeticiones innecesarias.

Ejemplo:

```
fun cuadrado(x: Int): Int {

    return x * x

}
```

```
fun main() {
    println(cuadrado(3))
}
```

}