



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

TEMA:

Secuencias de iteración, control y funciones

PRESENTADO POR:

Medina Palacios Fernando

GRUPO:

9° B

MATERIA:

Desarrollo para dispositivos inteligentes

CARRERA:

TI. Desarrollo de software multiplataforma

Docente:

Ray Brunett Parra Galaviz

Tijuana, Baja California, 02 de octubre del 2024

Secuencias de iteración

Las secuencias en Kotlin sirven para optimizar las operaciones sobre colecciones, evitando la creación de objetos temporales entre los pasos de una cadena de computación.

las secuencias usadas son las siguientes:

- **Bucle for:** Es útil cuando se desea iterar sobre una colección o un rango de números. También se pueden definir rangos ascendentes y descendentes, y ajustar el tamaño de los pasos en las iteraciones.
- **Bucle while:** Este bucle repite un bloque de código mientras se cumpla una condición. Antes de cada iteración, verifica si la condición es verdadera.
- **Bucle do-while:** Similar a while, pero con la diferencia de que el código dentro del bucle se ejecuta al menos una vez, ya que la condición se evalúa después de cada iteración.

Secuencias de control

permiten alterar el flujo de ejecución del programa en función de ciertas condiciones. Kotlin ofrece varias estructuras de control como if y when.

- **Condicional if:** Esta estructura es utilizada para tomar decisiones basadas en una condición. A diferencia de otros lenguajes, en Kotlin if es una expresión, lo que significa que puede devolver un valor. Esto permite utilizarlo tanto para controlar el flujo como para asignar valores directamente en variables.
- **Expresión when:** permite manejar múltiples condiciones en un formato más conciso. Soporta no solo comparaciones simples con valores exactos, sino también condiciones más complejas como rangos, tipos o cualquier condición booleana. Además, puede utilizarse como una expresión que devuelve un valor.

funciones

se pueden definir para realizar tareas específicas. Kotlin permite definir funciones de diferentes maneras, incluidas funciones de una sola línea y funciones de orden superior.

- **Funciones estándar:** Se pueden definir con parámetros y un tipo de retorno. Si una función no necesita devolver un valor, se le asigna el tipo unit, que es el equivalente a void en otros lenguajes.
- **Funciones de una sola expresión:** Kotlin permite definir funciones en una sola línea cuando el cuerpo de la función es simple. Esto simplifica el código y lo hace más legible.

- **Parámetros con valores por defecto:** Kotlin permite asignar valores predeterminados a los parámetros de las funciones, lo que hace que ciertos argumentos sean opcionales al momento de llamar la función. Esto permite llamadas más flexibles sin necesidad de sobrecargar funciones.
- **Funciones de orden superior:** Kotlin soporta funciones que pueden aceptar otras funciones como parámetros o devolver funciones como resultado. Esto permite aplicar técnicas funcionales dentro de Kotlin y escribir código más modular y reutilizable.