

**Taller Estructuras de Datos en Kotlin**

El objetivo de este taller es que los aprendices sean capaces de comprender y utilizar las principales estructuras de datos en Kotlin, incluyendo arreglos, listas, conjuntos, mapas y pares.

Michael Nicolas Rodríguez Rodríguez ficha :2469283 git: https://github.com/Maicolrr/Ejercicios-en-kotlin

El aprendiz deberá realizar un informe donde se evidencien los siguientes puntos:

1. Introducción a las estructuras de datos en Kotlin
   1. ¿Qué son las estructuras de datos y para qué se utilizan?

R/: permite almacenar y organizar datos de un mismo tipo

* 1. Ventajas de utilizar estructuras de datos en Kotlin

R/: permite hacer un software ma eficiente ya que optimizamos recursos

* 1. Diferencias entre las estructuras de datos en Kotlin y Java

1. Arreglos en Kotlin
   1. ¿Qué es un arreglo?

R/: colección ordenada de datos que tienen el mismo tipo de dato. Todos los elementos son referenciados por un mismo nombre

* 1. Creación de arreglos en Kotlin

R/: var arreglo = arrayOf ("pepe", "juan","Maria","ana")

* 1. Accediendo a los elementos de un arreglo

R/:

var arreglo = arrayOf ("pepe", "juan","Maria","ana")

        println("tamaño es: ${arreglo.count()}")

    println("posicion 1: ${arreglo[0]}")

    println("posicion 2: ${arreglo[1]}")

* 1. Modificando los elementos de un arreglo
  2. Recorriendo un arreglo

1. R/:

for((i.value) in arreglo.withIndex()){

        print("posicion: $i = ${arreglo[i]}")

    }

* 1. Funciones útiles para trabajar con arreglos en Kotlin

R/:

fun main (){

    var arreglo = arrayOf ("pepe", "juan","Maria","ana")

    println("tamaño es: ${arreglo.count()}")

    println("posicion 1: ${arreglo[0]}")

    println("posicion 2: ${arreglo[1]}")

    for(i in arreglo){

        println(i)

    }

}

1. Listas en Kotlin
   1. ¿Qué es una lista?

R/: colección de elementos con un orden específico. Hay dos tipos de listas en Kotlin:

Lista de solo lectura: List no se puede modificar después de su creación.

Lista mutable: MutableList se puede modificar después de su creación.

* 1. Creación de listas en Kotlin

R/:

val lista = listOf(1, 2, 3, 4, 5) *//no se puede modificar*

val listaMutable = mutableListOf(1, 2, 3, 4, 5) *//se puede modificar*

* 1. Accediendo a los elementos de una lista

R/:

val listaNombreMutable = mutableListOf("juan","pepe", "ana", "luis", "camilo") *//se puede modificar* y también sin modificarse

println("tamaño de la lista: "+listaNombreMutable.count())

* 1. Modificando los elementos de una lista

R/:

val listaMutable = mutableListOf(1, 2, 3, 4, 5) *//se puede modificar*

    listaMutable.add(6)

    println(listaMutable)

val listaNombreMutable = mutableListOf("juan","pepe", "ana", "luis", "camilo") *//se puede modificar*

    println("llista  sin modificar"+listaNombreMutable)

    listaNombreMutable.removeAt(3)*//elimina la posicion*

    println("llista modificada"+listaNombreMutable)

* 1. Recorriendo una lista

R/:

val lista = listOf(1, 2, 3, 4, 5) *//no se puede modificar*

    println(lista)

    for(i in lista){

          println(i)

    }

* 1. Funciones útiles para trabajar con listas en Kotlin

R/: fun main(){

    val lista = listOf(1, 2, 3, 4, 5) *//no se puede modificar*

    println(lista)

    for(i in lista){

          println(i)

    }

    val listaMutable = mutableListOf(1, 2, 3, 4, 5) *//se puede modificar*

    println("lista sin modificar"+listaMutable)

    listaMutable.add(6)

    println("lista modificada"+listaMutable)

    for(i in listaMutable){

          println(i)

    }

}

1. Conjuntos en Kotlin
   1. ¿Qué es un conjunto?

R/: es una colección sin orden de elementos únicos, esto es, no puede tener ningún duplicado.

* 1. Creación de conjuntos en Kotlin

R/:

var conjunto : Set<Int> = setOf(1, 2, 3, 4)*//inmutable*

    var conjuntoMezclado = setOf(1, 2.45,"hola", "m")*//inmutable y con diferentes tipos de datos*

var conjuntoMezclado2 = mutableSetOf(1, 2.45,"hola", "m")*//mutable y con diferentes tipos de datos*

* 1. Accediendo a los elementos de un conjunto

R/:

var conjunto : Set<Int> = setOf(1, 2, 3, 4)*//inmutable*

    var conjuntoMezclado = setOf(1, 2.45,"hola", "m")*//inmutable y con diferentes tipos de datos*

    var conjuntoMezclado2 = mutableSetOf(1, 2.45,"hola", "m")*//mutable y con diferentes tipos de datos*

    println("conjunto: $conjunto")

    println("conjunto Mezclado: $conjuntoMezclado")

    println("conjunto Mezclado2: $conjuntoMezclado2")

* 1. Modificando los elementos de un conjunto

R/:

conjuntoMezclado2.add("juan")*//añade*

    conjuntoMezclado2.remove(1)*//elimina*

    conjuntoMezclado2.clear()*//limpia*

* 1. Recorriendo un conjunto

R/:

var conjunto : Set<Int> = setOf(1, 2, 3, 4)*//inmutable*

for (i in conjunto) {

        println(i)

    }

* 1. Funciones útiles para trabajar con conjuntos en Kotlin

R/:

fun main (){

    var conjunto : Set<Int> = setOf(1, 2, 3, 4)*//inmutable*

    var conjuntoMezclado = setOf(1, 2.45,"hola", "m")*//inmutable y con diferentes tipos de datos*

    var conjuntoMezclado2 = mutableSetOf(1, 2.45,"hola", "m")*//mutable y con diferentes tipos de datos*

    println("conjunto: $conjunto")

    println("conjunto Mezclado: $conjuntoMezclado")

    println("conjunto Mezclado2: $conjuntoMezclado2")

    conjuntoMezclado2.add("juan")*//añade*

    println("conjunto Mezclado2 Actualizado: $conjuntoMezclado2")

    conjuntoMezclado2.remove(1)*//elimina*

    println("conjunto Mezclado2 Actualizado: $conjuntoMezclado2")

    conjuntoMezclado2.clear()*//limpia*

    println("conjunto Mezclado2 Actualizado: $conjuntoMezclado2")

*///////////////////////////////////////////////////////////////////*

    var suma : Int = 0

    for (i in conjunto) {

        println(i)

        suma+=i

    }

    println("la suma de los elementos del conjunto es: $suma")

}

1. Mapas en Kotlin
   1. ¿Qué es un mapa?

R/: Un mapa es una colección que almacena sus elementos (entradas) en forma de pares clave-valor. Esto quiere decir que a cada clave le corresponde un solo valor y será única como si se tratase de un identificador.

* 1. Creación de mapas en Kotlin

R/:

println("nombres: $mapaNombres1")

println("nombres: $mapaNombres2")

println("claves: ${mapaNombres2.keys}")//tre las claves

println("valores: ${mapaNombres2.values}")//tre los valores

println("la clave 2 tiene como valor: "+mapaNombres2[2])//tre los valores de una clave determinada

* 1. Accediendo a los elementos de un mapa

R/:

println( "nombres: $mapaNombres1")

println( "nombres: $mapaNombres2")

println( "claves: ${mapaNombres2.keys}")//tre las claves

println( "valores: ${mapaNombres2.values}")//tre los valores

* 1. Modificando los elementos de un mapa

R/:

mapaNombres2[4]="carlos"//agregar claves y valores

mapaNombres2.put(5,"andres")//agregar claves y valores

mapaNombres2.set(6,"yara")//agregar claves y valores

* 1. Recorriendo un mapa

R/:

val mapaNombres2 = mutableMapOf(1 to "juan", 2 to "ana", 3 to "pepe")//modificable

for(dato in mapaNombres2){

        println("Clave${dato.key}: ${dato.value}")

    }

* 1. Funciones útiles para trabajar con mapas en Kotlin

R/:

fun main() {

    val mapaNombres1 = mapOf("clave1" to "juan", "clave2" to "ana", "clave3" to "pepe")

    val mapaNombres2 = mutableMapOf(1 to "juan", 2 to "ana", 3 to "pepe")//modificable

    println("nombres: $mapaNombres1")

    println("nombres: $mapaNombres2")

    println("claves: ${mapaNombres2.keys}")//tre las claves

    println("valores: ${mapaNombres2.values}")//tre los valores

    println("la clave 2 tiene como valor: "+mapaNombres2[2])//tre los valores de una clave determinada

    mapaNombres2[4]="carlos"//agregar claves y valores

    mapaNombres2.put(5,"andres")//agregar claves y valores

    mapaNombres2.set(6,"yara")//agregar claves y valores

    println( "nombres: $mapaNombres2")

    for(dato in mapaNombres2){

        println("Clave${dato.key}: ${dato.value}")

    }

}

1. Pares en Kotlin
   1. ¿Qué es un par?

R/: estructura que permite guardar dos valores. Esto en Android es muy útil ya que podemos emplear los Pair Para guardar una pareja de valores, por ejemplo un usuario y contraseña.

* 1. Creación de pares en Kotlin

R/:

var pair = Pair("Kotlin",2)

    var pair2 = "Kotlin" to 2

    var (user, password) =Pair("pepe\_123","pepito12345")

* 1. Accediendo a los elementos de un par

R/:

var pair = Pair("Kotlin",2)

println(pair.first )*//imprimo el primer caracter*

    println(pair.second)*//imprimo el segundo carácter*

*//////////////////////////////////////////////////////////////*

var (user, password) =Pair("pepe\_123","pepito12345")

println("el nombre de usuario es: $user")

    println("la contraseña es: $password")

* 1. Modificando los elementos de un par
  2. Recorriendo un par
  3. Funciones útiles para trabajar con pares en Kotlin

1. Prácticas de estructuras de datos en Kotlin
   1. Ejercicios prácticos para aplicar los conceptos aprendidos
   2. Solución a los ejercicios prácticos

Recursos adicionales:

Taller Estructuras de Datos en Kotlin 1

Documentación oficial de Kotlin: [**https://kotlinlang.org/docs/reference/**](https://kotlinlang.org/docs/reference/)

**Entrega.**

Se deberá realizar la entrega de un informe con la solución de los puntos anteriores, el aprendiz acompañará la investigación con ejemplos practicos de cada estructura y deberá publicar el código fuente en un repositorio en GitHub.

Taller Estructuras de Datos en Kotlin 2