



Kotlin - Programación orientada a objetos

—

Diego Alexander Araque Molina
Stefania Arguelles Reyes

—

kotlin es un lenguaje enfocado a la programación orientada a objetos

Fue nombrado lenguaje del mes en la edición de enero de 2012 de la revista Dr. Boob.

Está diseñado para interoperar con código java

En Kotlin el manejo de datos se realiza de manera similar a los demás lenguajes, se ingresa por teclado y retorna por consola

Ingreso por teclado:

```
readLine()  
readLine()!!
```

Salida por consola:

```
print("HolaMundo")  
println("HolaMundo")
```

Los operadores pueden ser:

Aritméticos: Son operaciones matemáticas básicas

Lógicos: Para establecer una condición lógica

Relacionales: Para evaluar si se cumple una relación

Aritméticos:

Suma: $a + b$

Resta: $a - b$

Multiplicación: $a * b$

División: a / b

Módulo: $a \% b$

Incremento: $a++$

Decremento: $b--$

Relacionales:

($>$, $<$, $>=$, $<=$, $==$, $!=$)

Lógicos:

($\&\&$, $||$)

Son palabras reservadas que ayudan a establecer condiciones

as, is, in

```
if (x is Int) { }
```

```
val text = other as String
```

```
if (x in 0..10) { }
```

if- else:

```
if (valor1 > valor2) {  
    print("El mayor valor es $valor1")  
}  
else {  
    print("El mayor valor es $valor2")  
}
```

Puede ser utilizada como expresión

```
val resultado = if (valor % 2 == 0) {  
    print("Cuadrado:")  
    valor * valor  
} else {  
    print("Cubo:")  
    valor * valor * valor  
}
```

When:

```
when {
```

```
    x > 0 && y > 0 -> println("Primer cuadrante")
```

```
    x < 0 && y > 0 -> println("Segundo cuadrante")
```

```
    x < 0 && y < 0 -> println("Tercer cuadrante")
```

```
    x > 0 && y < 0 -> println("Cuarto cuadrante")
```

```
    else -> println("El punto se encuentra en un eje")
```

```
}
```

while:

```
var x = 1
```

```
while (x <= 100) {
```

```
    println(x)
```

```
    x = x + 1
```

```
}
```

For:

```
for(i in 1..10 step 2)  
    println(i)
```

```
for(i in 10 downTo 1 step 2)  
    println(i)
```

En kotlin la solución de un problema complejo se logra descomponiendolo en funciones que solucionan pequeñas partes del problema

```
fun mostrarSuperficie(lado: Int) {  
    val superficie = lado * lado  
    println("La superficie es $superficie")  
}
```


Programación Orientada a Objetos

Kotlin

Kotlin es un lenguaje enfocado en la programación orientada a objetos, se destaca su facilidad para crear objetos e instanciar clases

Clase: Plantilla para definir propiedades y métodos

Método: Entidad independiente con sus propios datos y programación

Definir un método:

```
fun imprimir() {  
    println("Nombre: $nombre y tiene  
una edad de $edad")  
}
```

Método constructor:

```
class Persona (var nombre: String, var  
edad: Int)
```

Objeto nombrado:

Uso de la palabra clave 'object' para definir un objeto sin necesidad de declarar una clase

```
object Matematica {  
    val PI = 3.1416  
  
    fun aleatorio(minimo: Int, maximo: Int)  
= ((Math.random() * (maximo + 1 -  
minimo)) + minimo).toInt()  
}
```

Acceder a un método desde otro método:

Directamente al método:

sumar()

restar()

Desde la clase:

```
val operaciones1 = Operaciones()
```

```
    operaciones1.cargar()
```

En la programación orientada a objetos las clases interactúan y colaboran entre sí

```
class Dado (var valor: Int){  
    fun tirar() { // Sentencias}  
class JuegoDeDados {  
    val dado1 = Dado(1)  
    fun jugar() { //Sentencias    }  
}
```


Encapsulamiento:

Para acceder a un método desde una clase determinada se modifica el acceso de acuerdo a las sentencias `public`, `protected`, `private`

```
private fun sumar() {  
    //Sentencias  
}
```

`private`

`protected`

`public`

Data Class:

Una clase que solo guarda datos

```
data class Artículo(var codigo: Int, var  
descripcion: String, var precio: Float)
```

Enum Class:

Se utiliza para definir un conjunto de constantes

```
enum class TipoCarta{  
    DIAMANTE,  
    TREBOL,  
    CORAZON,  
    PICA  
}
```

Herencia:

Es una relación entre clases que genera la creación de una clase a partir de otra

Auto "es un" Vehiculo

Circulo "es una" Figura

Mouse "es un" DispositivoEntrada

Suma "es una" Operacion

```
open class Persona(val nombre: String, val edad: Int) {  
    open fun imprimir() { //Sentencia }  
}  
  
class Empleado(nombre: String, edad: Int, val sueldo:  
Double): Persona(nombre, edad) {  
    override fun imprimir() { //Sentencia }  
  
    fun pagalmpuestos() { //sentencia }  
}
```

Clase abstracta:

Una clase que posee métodos y propiedades que son útiles a distintas clases

```
abstract class Operacion(val valor1: Int, val valor2:
Int) {
    protected var resultado: Int = 0

    abstract fun operar()

    fun imprimir() {
        println("Resultado: $resultado")
    }
}
```

Interfaces:

Declara una serie de métodos y propiedades que deben ser implementados por las clases

Utilizada para crear múltiples herencias


```
interface Punto { fun imprimir() }
class PuntoPlano(val x: Int, val y: Int): Punto {
    override fun imprimir() { //Sentencia }
}
class PuntoEspacio(val x: Int, val y: Int, val z: Int): Punto {
    override fun imprimir() { //Sentencia }
}
fun main(parametro: Array<String>) {
    val puntoPlano1 = PuntoPlano(10, 4)
    puntoPlano1.imprimir()
    val puntoEspacio1 = PuntoEspacio(20, 50, 60)
    puntoEspacio1.imprimir()
}
```

<https://www.tutorialesprogramacionya.com/kotlinya/index.php?inicio=0>

<https://fabiomsr.github.io/from-java-to-kotlin/index.html>

<https://www.toptal.com/software/introducci%C3%B3n-a-kotlin-programaci%C3%B3n-de-android-para-seres-humanos/es>

kotlin para android <https://youtu.be/P9ZFhTw2hGA>

curso de kotlin

<https://devcode.la/cursos/kotlin/#clases/programacion-orientada-a-objetos/distribucion-del-codigo-fuente-en-kotlin>

<https://kotlinlang.org/docs/reference/classes.html>

ide

online <https://try.kotlinlang.org/#/Examples/Hello.%20world!/Reading%20a%20name%20from%20the%20command%20line/Reading%20a%20name%20from%20the%20command%20line.kt>

tutorial <https://www.tutorialspoint.com/kotlin/index.htm>

programación orientada a objetos con kotlin <https://www.youtube.com/watch?v=zR92kdl21Ww>

ejemplos github <https://github.com/DevPicon/kotlin-examples/tree/master/src/la/devcode/kotlin>

tutoriales básicos <https://devcode.la/cursos/kotlin/#clases>



GRACIAS