



Paradigmas de Programación (EIF-400) FP Kotlin Parte 03

CARLOS LORÍA-SÁENZ LORACARLOS@GMAIL.COM

SETIEMBRE-OCTUBRE 2017

EIF/UNA

Objetivos

2

- ▶ Estudiar el paradigma de OOP-FP en el caso de Kotlin
- ▶ Contrastar con JS/Java8
- ▶ Clases/Objetos continuación

Material

3

- ▶ En el sitio
- ▶ [Kotlin referencia](#)

Básico

4

► Cubrir basics

Clases Resumen

5

- ▶ Clases
 - ▶ Constructores y propiedades
 - ▶ Bloque(s) inicializador(es) (`init`)
 - ▶ Constructores múltiples
 - ▶ Creación: No hay operador `new`
 - ▶ `public` por defecto
- ▶ Herencia (no `extends`)
 - ▶ Closed por defecto
 - ▶ `open, override`
- ▶ Funciones de extensión

Propiedades en detalle

6

- ▶ “Propiedad” (property) generaliza “campo” (field)
- ▶ Es NORMALMENTE un campo más un accesor más un setter (opcional). Ese campo se le llama “backing field”
- ▶ Pueden haber propiedades sin backing-field
- ▶ Pueden ser read-only: sólo se puede leer
- ▶ Modificable: se puede además mutar
- ▶ En Kotlin `val` declara una propiedad readonly
- ▶ Con `var` se declara modificable

Ejercicios

7

- ▶ Considere el demo `oop.kt` (en work)
- ▶ Logre los siguiente
- ▶ Haga una clase `Triangle` derivada de `Shape2d` y un caso de prueba similar al de `rect`. Debe tener `Color` también
- ▶ ¿Tiene su solución redundancia? Corrija si es así
- ▶ Agruegue `Square` derivada de `Rectangle` y su correspondiente caso de prueba.
- ▶ Convierta la función `project` en un método de extensión de `Rectangle`. Modifique el caso de prueba para que use ahora ese método
- ▶ Modifique `project` para que retorne un `object`.

Continuará...

- ▶ Más sobre interfaces
- ▶ Lambdas
- ▶ FP en colecciones