Dia #6

**Programación Orientada a Objetos**

¿Qué es un Objeto?

Se puede clasificar un objeto como cualquier cosa, también como una variable.

Los objetos tienes propiedades, características, métodos, clases. Que los hace únicos

Clases, Propiedades y Objetos:

class Frutas (color: String , sabor:String , precio:Int ){

var color:String = ""

var sabor:String = ""

var precio:Int = 0

init{

this.color=color

this.sabor=sabor

this.precio=precio

}

}

fun main() {

var Manzana = Frutas("Rojo", "Dulce", 4)

println(Manzana.color)

}

Métodos (Funciones en POO) :

fun podrirse(){

println("La fruta se está pudriendo!!!")

frescura=frescura-1

}

fun FrescuraTotal(){

println("La frescura total de la Fruta es de $frescura %")

}

Estos métodos van dentro de la clase

Constructores:

// Propiedades de la clase Frutas

var color: String = ""

var sabor: String = ""

var precio: Int = 0

var frescura: Int = 0

// Constructor primario

init {

this.color = color

this.sabor = sabor

this.precio = precio

this.frescura = frescura

}

// Propiedad adicional

var numeroHojas: Int = 0

// Constructor secundario

constructor(color: String, sabor: String, precio: Int, frescura: Int, numeroHojas: Int) : this(color, sabor, precio, frescura) {

this.numeroHojas = numeroHojas

}

Herencia en Kotlin:

Primero, para entender la herencia en Kotlin, debemos comprender que tenemos 2 tipos de clases: Las “Open” y las “Final”, si se crea una clase como lo hemos visto con la palabra reservada “class” estamos creando una clase “Final”, y estas no pueden heredar su contenido a otras.

Para crear una clase open debemos poner las palabras reservadas “open class” para así tener la clase tipo “Open” y que pueda heredar su contenido a otras clases

class FrutasConGrasa(color: String, sabor: String, precio: Int, frescura: Int, numeroHojas: Int, cantidadGrasa: Int) : Frutas(color, sabor, precio, frescura) {

var cantidadGrasa: Int = 0

init {

this.cantidadGrasa = cantidadGrasa

}

}

**Código Completo de POO:**

open class Frutas(color: String, sabor: String, precio: Int, frescura: Int) {

var color: String = ""

var sabor: String = ""

var precio: Int = 0

var frescura: Int = 0

init {

this.color = color

this.sabor = sabor

this.precio = precio

this.frescura = frescura

}

var numeroHojas: Int = 0

constructor(color: String, sabor: String, precio: Int, frescura: Int, numeroHojas: Int) : this(color, sabor, precio, frescura) {

this.numeroHojas = numeroHojas

}

fun podrirse() {

println("La fruta se está pudriendo!!!")

frescura -= 1

}

fun frescuraTotal() {

println("La frescura total de la fruta es de $frescura %")

}

}

class FrutasConGrasa(color: String, sabor: String, precio: Int, frescura: Int, numeroHojas: Int, cantidadGrasa: Int) : Frutas(color, sabor, precio, frescura) {

var cantidadGrasa: Int = 0

init {

this.cantidadGrasa = cantidadGrasa

}

}

fun main() {

// Crear una instancia de Frutas (Manzana)

val manzana = Frutas("Rojo", "Dulce", 4, 100, 0)

// Imprimir propiedades de la fruta (Manzana)

println("Color de la fruta: ${manzana.color}")

println("Número de hojas de la fruta: ${manzana.numeroHojas}")

// Simular deterioro de la fruta (Manzana)

manzana.podrirse()

manzana.frescuraTotal()

// Crear una nueva instancia con la clase heredada

val aguacate = FrutasConGrasa("Verde", "Neutro", 20 , 100, 0, 15)

// Imprimir la cantidad de grasa del aguacate

println("Cantidad de grasa del aguacate: ${aguacate.cantidadGrasa}")

}