Programació de serveis i processos - Sebastian Ciscar

None

None

Copyright - Sebastian Ciscar - Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

Table of contents

1. Exercicis de Kotlin variables i tipus de dades 4
   - 1.1 1. Variables i Tipus de Dades 4
   - 1.2 2. Declaració explícita de tipus 4
   - 1.3 3. Inferència de tipus 5
   - 1.4 4. Concatenació de cadenes 5
   - 1.5 5. Plantilles de cadena 6
   - 1.6 6. Null Safety 7
   - 1.7 7. Conversió de tipus 7
   - 1.8 8. Treballant amb el tipus `Char` 8
   - 1.9 9. Operacions aritmètiques 9
2. Control de flux 10
   - 2.1 Exercici 1: Conversió de temperatures 10
   - 2.2 Exercici 2: Número més gran 10
   - 2.3 Exercici 3: Generador de contrasenyes aleatòries 10
   - 2.4 Exercici 4: Calculadora de l'IMC 11
   - 2.5 Exercici 5: Taula de multiplicar 11
   - 2.6 Exercici 6: Adivina el número 11
   - 2.7 Exercici 7: Comptador de vocals 12
   - 2.8 Exercici 8: Nombre perfecte 12
   - 2.9 Exercici 9: Conversió d'hores a minuts i segons 12
   - 2.10 Exercici 10: FizzBuzz personalitzat 13
3. Funcions 14
   - 3.1 Exercici 1: Funció de suma amb valor per defecte 14
   - 3.2 Exercici 2: Funció que retorna múltiples valors 14
   - 3.3 Exercici 3: Filtratge de llista amb lambda 14
   - 3.4 Exercici 4: Concatenació de cadenes 14
   - 3.5 Exercici 5: Funció d'ordre superior 15
   - 3.6 Exercici 6: Lambda amb llista de preus 15
   - 3.7 Exercici 7: Funció anònima 15
   - 3.8 Exercici 8: Filtres múltiples amb lambda 16
   - 3.9 Exercici 9: Conversió de temperatures amb funcions d'extensió 16
   - 3.10 Exercici 10: Funció recursiva 16
4. POO 17
   - 4.1 Exercici 1: Creació d'una classe Persona 17
   - 4.2 Exercici 2: Constructor secundari 17

Copyright - Sebastian-CC23 - Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

4.3 Exercici 3: Herència en classes 17
4.4 Exercici 4: Getters i setters personalitzats 18
4.5 Exercici 5: Objecte Singleton 18
4.6 Exercici 6: Classes abstractes 19
4.7 Exercici 7: Interfícies 19
4.8 Exercici 8: Polimorfisme 19
4.9 Exercici 9: Funcions d'àmbit 20
4.10 Exercici 10: Companion object 20
4.11 Exercici 1: Creació d'un array de números enters 21
4.12 Exercici 2: Accés a elements d'un array 21
4.13 Exercici 3: Llista immutable de números 21
4.14 Exercici 4: Llista mutable de cadenes 21
4.15 Exercici 5: Funcions sobre llistes 21
4.16 Exercici 6: Conjunt immutable 22
4.17 Exercici 7: Mapa immutable 22
4.18 Exercici 8: Recorregut d'un array amb índexs 22
4.19 Exercici 9: Operacions amb llistes mutables 22
4.20 Exercici 10: Mapa mutable 23
5. Solucions 23

Copyright - Sebastian-Cano -- Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

1. Exercicis de Kotlin variables i tipus de dades

1.1 1. Variables i Tipus de Dades

1.1.1 Exercici 1.1:

Enunciat: Declara una variable `var` que continga el número enter 10 i, després, canvia el seu valor a 20.

Solució

```
var numero = 10
numero = 20
println(numero)
```

1.1.2 Exercici 1.2:

Enunciat: Declara una variable `val` amb el valor de text "Hola" i intenta canviar el seu valor a "Adéu". Observa l'error que es genera.

Solució

```
val salutacio = "Hola"
// salutacio = "Adéu" // Error: Val cannot be reassigned
println(salutacio)
```

1.1.3 Exercici 1.3:

Enunciat: Declara una variable `var` amb el valor "Bon dia" i canvia-la a "Bona vesprada".

Solució

```
var frase = "Bon dia"
frase = "Bona vesprada"
println(frase)
```

1.2 2. Declaració explícita de tipus

1.2.1 Exercici 2.1:

Enunciat: Declara una variable de tipus `Int` amb el valor 25 explícitament.

Solució

```
val edat: Int = 25
println(edat)
```

1.2.2 Exercici 2.2:

Enunciat: Declara una variable de tipus `Double` amb el valor 3.14 explícitament.

Copyright - Sebastian-Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

Solució

```
val pi: Double = 3.14
println(pi)
```

1.2.3 Exercici 2.3:

Enunciat: Declara una variable de tipus `Boolean` amb el valor true.

Solució

```
val esVeritat: Boolean = true
println(esVeritat)
```

1.3 3. Inferència de tipus

1.3.1 Exercici 3.1:

Enunciat: Declara una variable amb el valor 100 sense especificar el tipus. Què inferix Kotlin?

Solució

```
val numero = 100
println(numero::class) // Class kotlin.Int
```

1.3.2 Exercici 3.2:

Enunciat: Declara una variable amb el valor 2.5 sense especificar el tipus. Què inferix Kotlin?

Solució

```
val decimal = 2.5
println(decimal::class) // Class kotlin.Double
```

1.3.3 Exercici 3.3:

Enunciat: Declara una variable amb el text "Kotlin" sense especificar el tipus. Què inferix Kotlin?

Solució

```
val llenguatge = "Kotlin"
println(llenguatge::class) // Class kotlin.String
```

1.4 4. Concatenació de cadenes

1.4.1 Exercici 4.1:

Enunciat: Declara dues variables amb els valors "Hola" i "Món" i concatena-les utilitzant l'operador +.

Copyright - Sebastian Cisca - Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

Solució

```
val part1 = "Hola"
val part2 = "Món"
val resultat = part1 + " " + part2
println(resultat) // Hola Món
```

1.4.2 Exercici 4.2:

Enunciat: Crea dues variables que contindran un nom i una edat, i concatena-les en una frase.

Solució

```
val nom = "Joan"
val edat = 30
val frase = "El meu nom és " + nom + " i tinc " + edat + " anys."
println(frase)
```

1.4.3 Exercici 4.3:

Enunciat: Declara una variable amb una frase i concatena una altra cadena que l'usuari introduïsca per teclat.

Solució

```
val fraseInicial = "Benvingut, "
print("Introduïx el teu nom: ")
val nom = readLine()
val fraseFinal = fraseInicial + nom
println(fraseFinal)
```

1.5 5. Plantilles de cadena

1.5.1 Exercici 5.1:

Enunciat: Declara una variable amb el nom "Maria" i utilitza-la dins d'una plantilla de cadena per a mostrar-la.

Solució

```
val nom = "Maria"
println("Hola, $nom!")
```

1.5.2 Exercici 5.2:

Enunciat: Declara dues variables numèriques, suma-les, i mostra el resultat utilitzant plantilles de cadena.

Solució

```
val num1 = 5
val num2 = 3
println("La suma de $num1 i $num2 és ${num1 + num2}.")
```


Copyright - Sebastian Gil -- Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

1.5.3 Exercici 5.3:

Enunciat: Utilitza una plantilla de cadena per a mostrar el valor d'una variable dins d'una frase.

Solució

```
val ciutat = "València"
println("Estic visitant $ciutat aquest cap de setmana.")
```

1.6 6. Null Safety

1.6.1 Exercici 6.1:

Enunciat: Declara una variable `String?` nullable i comprova si té un valor abans d'imprimir-la.

Solució

```
val nom: String? = null
if (nom != null) {
    println(nom)
} else {
    println("La variable és nul·la.")
}
```

1.6.2 Exercici 6.2:

Enunciat: Utilitza l'operador segur (`?.`) per a accedir a les propietats d'una variable nullable.

Solució

```
val nom: String? = "Joan"
println(nom?.length) // Mostra la longitud només si no és null
```

1.6.3 Exercici 6.3:

Enunciat: Proporciona un valor predeterminat per a una variable nullable utilitzant l'operador Elvis (`?:`).

Solució

```
val nom: String? = null
val valorPredeterminat = nom ?: "Anònim"
println(valorPredeterminat) // Mostra "Anònim"
```

1.7 7. Conversió de tipus

1.7.1 Exercici 7.1:

Enunciat: Declara una variable `Int` amb el valor 50, converteix-la a `String` i concatena-la amb un text.

Copyright - Sebastian-CV23 - Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

Solució

```
val numero = 50
val text = numero.toString() + " és un número gran."
println(text)
```

1.7.2 Exercici 7.2:

Enunciat: Declara una variable `Double` amb el valor 8.7, converteix-la a `Int`, i mostra el resultat abans i després.

Solució

```
val decimal = 8.7
println("Valor original: $decimal")
val enter = decimal.toInt()
println("Convertit a enter: $enter")
```

1.7.3 Exercici 7.3:

Enunciat: Declara una variable `Char` amb el valor 'A' i converteix-la a número utilitzant `toInt()`.

Solució

```
val lletra = 'A'
val valorNumeric = lletra.toInt()
println(valorNumeric) // Mostra el valor numèric corresponent al caràcter 'A'
```

1.8 8. Treballant amb el tipus `Char`

1.8.1 Exercici 8.1:

Enunciat: Declara una variable `Char` amb el valor 'B' i mostra el seu valor numèric utilitzant `toInt()`.

Solució

```
val lletra = 'B'
println(lletra.toInt()) // Mostra el valor numèric de 'B'
```

1.8.2 Exercici 8.2:

Enunciat: Crea una variable `Char` a partir d'un valor numèric utilitzant `toChar()`.

Solució

```
val numero = 67
val lletra = numero.toChar()
println(lletra) // Mostra el caràcter corresponent al número 67
```

1.8.3 Exercici 8.3:

Enunciat: Declara una variable de tipus `Char` i converteix-la a majúscula utilitzant una funció integrada.

Solució

```
val lletra = 'b'
println(lletra.uppercaseChar()) // Converteix a majúscula
```

1.9 9. Operacions aritmètiques

1.9.1 Exercici 9.1:

Enunciat: Declara dues variables `var` de tipus `Int`, realitza una suma i mostra el resultat.

Solució

```
var num1 = 7
var num2 = 5
val suma = num1 + num2
println("La suma és $suma")
```

1.9.2 Exercici 9.2:

Enunciat: Declara dues variables `Double`, realitza una divisió i mostra el resultat.

Solució

```
val num1 = 9.0
val num2 = 3.0
val divisio = num1 / num2
println("La divisió és $divisio")
```

1.9.3 Exercici 9.3:

Enunciat: Declara dues variables, realitza una resta i una multiplicació, i mostra els resultats.

Solució

```
val num1 = 10
val num2 = 4
val resta = num1 - num2
val multiplicacio = num1 * num2
println("La resta és $resta")
println("La multiplicació és $multiplicacio")
```


Copyright - Sebastian-Gil -- Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

2. Control de flux

2.1 Exercici 1: Conversió de temperatures

Enunciat: Escriu un programa que demane a l'usuari una temperatura en graus Celsius i la convertisca a Fahrenheit. La fórmula per a la conversió és: `F = C * 9/5 + 32`.

Solució:

Solució

```
fun main() {
    print("Introdueix la temperatura en graus Celsius: ")
    val celsius = readLine()!!.toDouble()
    val fahrenheit = (celsius * 9/5) + 32
    println("La temperatura en Fahrenheit és: $fahrenheit")
}
```

2.2 Exercici 2: Número més gran

Enunciat: Crea un programa que demane a l'usuari tres números enters i determine quin és el més gran.

Solució:

Solució

```
fun main() {
    print("Introdueix tres números: ")
    val num1 = readLine()!!.toInt()
    val num2 = readLine()!!.toInt()
    val num3 = readLine()!!.toInt()
    val major = if (num1 > num2 && num1 > num3) num1 else if (num2 > num3) num2 else num3
    println("El número més gran és: $major")
}
```

2.3 Exercici 3: Generador de contrasenyes aleatòries

Enunciat: Escriu un programa que genere una contrasenya aleatòria de 8 caràcters, que continga lletres majúscules, minúscules i números.

Solució:

Solució

```
import kotlin.random.Random

fun main() {
    val charset = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"
    val password = (1..8)
        .map { charset.random() }
        .joinToString("")
    println("Contrasenya generada: $password")
}
```


Copyright - Sebastian Tena - Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

2.4 Exercici 4: Calculadora de l'IMC

Enunciat: Crea un programa que calcule l'Índex de Massa Corporal (IMC) en base al pes i l'altura introduïts per l'usuari. La fórmula és: `IMC = pes / altura^2`.

Solució:

Solució

```
fun main() {
    print("Introdueix el pes (kg): ")
    val pes = readLine()!!.toDouble()
    print("Introdueix l'altura (m): ")
    val altura = readLine()!!.toDouble()
    val imc = pes / (altura * altura)
    println("El teu IMC és: %.2f".format(imc))
}
```

2.5 Exercici 5: Taula de multiplicar

Enunciat: Fes un programa que demane a l'usuari un número i mostre la seua taula de multiplicar fins al 10.

Solució:

Solució

```
fun main() {
    print("Introdueix un número: ")
    val num = readLine()!!.toInt()
    for (i in 1..10) {
        println("$num x $i = ${num * i}")
    }
}
```

2.6 Exercici 6: Adivina el número

Enunciat: Crea un joc on l'ordinador trie un número aleatori entre 1 i 100, i l'usuari haja d'adivinar-lo. L'ordinador donarà pistes si el número introduït és massa alt o massa baix.

Solució:

Solució

```
import kotlin.random.Random

fun main() {
    val secret = Random.nextInt(1, 101)
    var guessed = false
    while (!guessed) {
        print("Adivina el número (1-100): ")
        val guess = readLine()!!.toInt()
        if (guess == secret) {
            println("Enhorabona! Has encertat.")
            guessed = true
        } else if (guess < secret) {
            println("Massa baix!")
        } else {
            println("Massa alt!")
        }
    }
}
```


Copyright - Sebastian Lizana - Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

2.7 Exercici 7: Comptador de vocals

Enunciat: Escriu un programa que demane una frase a l'usuari i compte quantes vocals conté.

Solució:

Solució

```
fun main() {
    print("Introdueix una frase: ")
    val frase = readLine()!!.toLowerCase()
    val vocals = "aeiou"
    var comptador = 0
    for (caracter in frase) {
        if (vocals.contains(caracter)) comptador++
    }
    println("La frase conté $comptador vocals.")
}
```

2.8 Exercici 8: Nombre perfecte

Enunciat: Fes un programa que determine si un número donat per l'usuari és un nombre perfecte (un nombre perfecte és aquell la suma dels seus divisors, excepte ell mateix, és igual al mateix nombre).

Solució:

Solució

```
fun main() {
    print("Introdueix un número: ")
    val num = readLine()!!.toInt()
    var suma = 0
    for (i in 1 until num) {
        if (num % i == 0) suma += i
    }
    if (suma == num) {
        println("$num és un nombre perfecte.")
    } else {
        println("$num no és un nombre perfecte.")
    }
}
```

2.9 Exercici 9: Conversió d'hores a minuts i segons

Enunciat: Crea un programa que convertisca una quantitat d'hores donada per l'usuari en minuts i segons.

Solució:

Solució

```
fun main() {
    print("Introdueix el número d'hores: ")
    val hores = readLine()!!.toInt()
    val minuts = hores * 60
    val segons = minuts * 60
    println("$hores hores són $minuts minuts i $segons segons.")
}
```


Copyright - Sebastian Romero Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

2.10 Exercici 10: FizzBuzz personalitzat

Enunciat: Escriu un programa que recórrega els números de l'1 al 50 i mostre "Fizz" si el número és múltiple de 3, "Buzz" si és múltiple de 5, i "FizzBuzz" si és múltiple de tots dos. A més, personalitza el programa perquè mostre un altre missatge per als múltiples de 7.

Solució:

Solució

```
fun main() {
    for (i in 1..50) {
        when {
            i % 3 == 0 && i % 5 == 0 -> println("FizzBuzz")
            i % 3 == 0 -> println("Fizz")
            i % 5 == 0 -> println("Buzz")
            i % 7 == 0 -> println("Boom")
            else -> println(i)
        }
    }
}
```


Copyright - Sebastian Tirado - Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

3. Funcions

3.1 Exercici 1: Funció de suma amb valor per defecte

Enunciat: Escriu una funció que accepte dos números i retorne la seua suma. Fes que un dels números tinga un valor per defecte.

Solució

```
fun suma(a: Int = 5, b: Int): Int {
    return a + b
}

fun main() {
    println(suma(b = 10)) // Output: 15
}
```

3.2 Exercici 2: Funció que retorna múltiples valors

Enunciat: Crea una funció que retorne un parell d'elements: el quocient i el residu de dividir dos números.

Solució

```
fun divisio(dividend: Int, divisor: Int): Pair<Int, Int> {
    return Pair(dividend / divisor, dividend % divisor)
}

fun main() {
    val (quocient, residu) = divisio(10, 3)
    println("Quocient: $quocient, Residu: $residu")
}
```

3.3 Exercici 3: Filtratge de llista amb lambda

Enunciat: Escriu una funció que filtre una llista d'enters i retorne només els parells, utilitzant una expressió lambda.

Solució

```
fun filtraParells(nums: List<Int>): List<Int> {
    return nums.filter { it % 2 == 0 }
}

fun main() {
    println(filtraParells(listOf(1, 2, 3, 4, 5, 6))) // Output: [2, 4, 6]
}
```

3.4 Exercici 4: Concatenació de cadenes

Enunciat: Crea una funció que accepte una quantitat variable de cadenes i les concatene en una sola cadena.

Copyright - Sebastian Tena Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

Solució

```
fun concatena(vararg cadenes: String): String {
    return cadenes.joinToString(" ")
}

fun main() {
    println(concatena("Hola", "a", "tots!")) // Output: "Hola a tots!"
}
```

3.5 Exercici 5: Funció d'ordre superior

Enunciat: Escriu una funció que accepte un altre funció com a paràmetre i aplique aquesta funció a dos nombres enters.

Solució

```
fun operacio(x: Int, y: Int, funcio: (Int, Int) -> Int): Int {
    return funcio(x, y)
}

fun main() {
    val suma = operacio(3, 4, { a, b -> a + b })
    println(suma) // Output: 7
}
```

3.6 Exercici 6: Lambda amb llista de preus

Enunciat: Escriu una funció que, donada una llista de preus, aplique un descompte del 10% a cadascun dels elements, utilitzant una expressió lambda.

Solució

```
fun aplicarDescompte(preus: List<Double>): List<Double> {
    return preus.map { it * 0.9 }
}

fun main() {
    println(aplicarDescompte(listOf(100.0, 200.0, 300.0))) // Output: [90.0, 180.0, 270.0]
}
```

3.7 Exercici 7: Funció anònima

Enunciat: Crea una funció que trie el número més gran entre tres valors, utilitzant una funció anònima.

Solució

```
fun majorDeTres(a: Int, b: Int, c: Int): Int {
    return fun(): Int {
        return if (a > b && a > c) a else if (b > c) b else c
    }()
}

fun main() {
    println(majorDeTres(3, 7, 5)) // Output: 7
}
```


Copyright - Sebastian Gil Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

3.8 Exercici 8: Filtres múltiples amb lambda

Enunciat: Escriu una funció que filtre una llista de cadenes, retornant només aquelles que tinguen més de 5 caràcters, utilitzant expressions lambda.

Solució

```
fun filtraCadenes(llista: List<String>): List<String> {
    return llista.filter { it.length > 5 }
}

fun main() {
    println(filtraCadenes(listOf("Hola", "Programació", "Kotlin", "Lambda"))) // Output: ["Programació", "Lambda"]
}
```

3.9 Exercici 9: Conversió de temperatures amb funcions d'extensió

Enunciat: Crea una funció d'extensió per a la classe `Double` que convertisca temperatures de Celsius a Fahrenheit.

Solució

```
fun Double.aFahrenheit(): Double {
    return this * 9 / 5 + 32
}

fun main() {
    println(25.0.aFahrenheit()) // Output: 77.0
}
```

3.10 Exercici 10: Funció recursiva

Enunciat: Escriu una funció recursiva que calcule el factorial d'un número donat.

Solució

```
fun factorial(n: Int): Int {
    return if (n == 0) 1 else n * factorial(n - 1)
}

fun main() {
    println(factorial(5)) // Output: 120
}
```


Copyright - Sebastian Tena Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

4. POO

4.1 Exercici 1: Creació d'una classe Persona

Enunciat: Escriu una classe `Persona` amb propietats per al nom, l'edat i el gènere. Afig un mètode per a mostrar la informació de la persona.

Solució

```
class Persona(val nom: String, var edat: Int, val genere: String) {
    fun mostrarInfo() {
        println("Nom: $nom, Edat: $edat, Gènere: $genere")
    }
}

fun main() {
    val persona = Persona("Anna", 25, "Femení")
    persona.mostrarInfo()
}
```

4.2 Exercici 2: Constructor secundari

Enunciat: Crea una classe `Cotxe` amb un constructor primari que accepte el model i l'any. Afig un constructor secundari que permeta crear un cotxe sense especificar l'any.

Solució

```
class Cotxe(val model: String, val any: Int) {
    constructor(model: String) : this(model, 2024)
    fun mostrarInfo() {
        println("Model: $model, Any: $any")
    }
}

fun main() {
    val cotxe1 = Cotxe("Toyota", 2020)
    val cotxe2 = Cotxe("Honda")
    cotxe1.mostrarInfo()
    cotxe2.mostrarInfo()
}
```

4.3 Exercici 3: Herència en classes

Enunciat: Crea una classe base `Animal` amb un mètode que faça un soroll. Crea dues subclasses, `Gos` i `Gat`, que sobreescriguen el mètode per a fer el seu propi soroll.

Copyright - Sebastian Lloret Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

Solució

```
open class Animal {
    open fun soroll() {
        println("L'animal fa soroll")
    }
}

class Gos : Animal() {
    override fun soroll() {
        println("El gos borda")
    }
}

class Gat : Animal() {
    override fun soroll() {
        println("El gat miola")
    }
}

fun main() {
    val gos = Gos()
    val gat = Gat()
    gos.soroll()
    gat.soroll()
}
```

4.4 Exercici 4: Getters i setters personalitzats

Enunciat: Crea una classe `CompteBancari` amb un saldo que no pot ser negatiu. Utilitza getters i setters personalitzats per a assegurar-te que no es pot retirar més diners dels que hi ha.

Solució

```
class CompteBancari(private var saldo: Double) {
    var saldoDisponible: Double
        get() = saldo
        set(value) {
            if (value >= 0) saldo = value else println("No pots tenir saldo negatiu!")
        }

    fun retirar(diners: Double) {
        if (diners <= saldo) saldo -= diners else println("No tens prou diners!")
    }
}

fun main() {
    val compte = CompteBancari(100.0)
    compte.retirar(50.0)
    println("Saldo actual: ${compte.saldoDisponible}")
    compte.saldoDisponible = -10.0
}
```

4.5 Exercici 5: Objecte Singleton

Enunciat: Crea un objecte `Calculadora` amb funcions estàtiques per a sumar, restar, multiplicar i dividir.

Solució

```
object Calculadora {
    fun sumar(a: Int, b: Int) = a + b
    fun restar(a: Int, b: Int) = a - b
    fun multiplicar(a: Int, b: Int) = a * b
    fun dividir(a: Int, b: Int) = a / b
}

fun main() {
    println(Calculadora.sumar(10, 5))
    println(Calculadora.dividir(10, 2))
}
```


Copyright - Sebastian Lis - Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

4.6 Exercici 6: Classes abstractes

Enunciat: Defineix una classe abstracta `Forma` amb un mètode abstracte `area()`. Crea subclasses `Cercle` i `Quadrat` que implementen aquest mètode.

Solució

```
abstract class Forma {
    abstract fun area(): Double
}

class Cercle(val radi: Double) : Forma() {
    override fun area(): Double = Math.PI * radi * radi
}

class Quadrat(val costat: Double) : Forma() {
    override fun area(): Double = costat * costat
}

fun main() {
    val cercle = Cercle(5.0)
    val quadrat = Quadrat(4.0)
    println("Àrea del cercle: ${cercle.area()}")
    println("Àrea del quadrat: ${quadrat.area()}")
}
```

4.7 Exercici 7: Interfícies

Enunciat: Crea una interfície `Volar` amb un mètode `vola()`. Fes que una classe `Avió` i una classe `Ocell` implementen aquesta interfície.

Solució

```
interface Volar {
    fun vola()
}

class Avio : Volar {
    override fun vola() {
        println("L'avió vola alt!")
    }
}

class Ocell : Volar {
    override fun vola() {
        println("L'ocell vola pel cel!")
    }
}

fun main() {
    val avio = Avio()
    val ocell = Ocell()
    avio.vola()
    ocell.vola()
}
```

4.8 Exercici 8: Polimorfisme

Enunciat: Crea una classe `Vehicle` amb un mètode `moure()`. Crea subclasses `Bicicleta` i `Cotxe` que sobreescriguen aquest mètode. Crea una funció que accepte un `Vehicle` i cride al seu mètode `moure()`.

Copyright - Sebastian Lisa - Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

Solució

```
open class Vehicle {
    open fun moure() {
        println("El vehicle es mou")
    }
}

class Bicicleta : Vehicle() {
    override fun moure() {
        println("La bicicleta roda")
    }
}

class Cotxe : Vehicle() {
    override fun moure() {
        println("El cotxe circula")
    }
}

fun movimentVehicle(vehicle: Vehicle) {
    vehicle.moure()
}

fun main() {
    val bicicleta = Bicicleta()
    val cotxe = Cotxe()
    movimentVehicle(bicicleta)
    movimentVehicle(cotxe)
}
```

4.9 Exercici 9: Funcions d'àmbit

Enunciat: Crea una classe `Llibre` amb propietats títol i autor. Utilitza funcions d'àmbit com `apply` i `run` per a inicialitzar un llibre i mostrar la seua informació.

Solució

```
class Llibre(val titol: String, val autor: String)

fun main() {
    val llibre = Llibre("El Quixot", "Cervantes").apply {
        println("Llibre: $titol, Autor: $autor")
    }

    llibre.run {
        println("Lectura del llibre: $titol per $autor")
    }
}
```

4.10 Exercici 10: Companion object

Enunciat: Crea una classe `Matemàtiques` amb un `companion object` que continga una funció per a calcular el factorial d'un número.

Solució

```
class Matematiques {
    companion object {
        fun factorial(n: Int): Int {
            return if (n == 0) 1 else n * factorial(n - 1)
        }
    }
}

fun main() {
    println(Matematiques.factorial(5)) // Output: 120
}
```


Copyright - Sebastian Zamora Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

4.11 Exercici 1: Creació d'un array de números enters

Enunciat: Escriu un programa que cree un array de 5 números enters i després mostre la suma de tots els elements.

????? info "Solució"

```
fun main() {
    val numeros = arrayOf(1, 2, 3, 4, 5)
    val suma = numeros.sum()
    println("La suma és: $suma")
}
```

4.12 Exercici 2: Accés a elements d'un array

Enunciat: Crea un array de cadenes amb els noms de la setmana. Mostra el nom del tercer dia utilitzant tant l'operador `[]` com la funció `get()`.

????? info "Solució"

```
fun main() {
    val dies = arrayOf("Dilluns", "Dimarts", "Dimecres", "Dijous", "Divendres")
    println(dies[2]) // Amb []
    println(dies.get(2)) // Amb get()
}
```

4.13 Exercici 3: Llista immutable de números

Enunciat: Crea una llista immutable de 5 números enters. Imprimeix la suma dels elements i indica si conté el número 10.

????? info "Solució"

```
fun main() {
    val numeros = listOf(1, 2, 3, 4, 5)
    val suma = numeros.sum()
    println("Suma: $suma")
    println("Conté 10: ${numeros.contains(10)}")
}
```

4.14 Exercici 4: Llista mutable de cadenes

Enunciat: Crea una llista mutable de tres cadenes. Afig una quarta cadena i imprimeix el contingut de la llista.

????? info "Solució"

```
fun main() {
    val llista = mutableListOf("Cadena 1", "Cadena 2", "Cadena 3")
    llista.add("Cadena 4")
    println(llista)
}
```

4.15 Exercici 5: Funcions sobre llistes

Enunciat: Escriu un programa que filtre els números parells d'una llista immutable de 10 elements.

Copyright - Sebastian Lisca - Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

????? info "Solució"

```
fun main() {
    val numeros = listOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    val parells = numeros.filter { it % 2 == 0 }
    println(parells)
}
```

4.16 Exercici 6: Conjunt immutable

Enunciat: Crea un conjunt immutable que continga alguns números. Mostra si el conjunt conté el número 5.

????? info "Solució"

```
fun main() {
    val conjunt = setOf(1, 2, 3, 4, 5)
    println("Conté 5: ${conjunt.contains(5)}")
}
```

4.17 Exercici 7: Mapa immutable

Enunciat: Crea un mapa immutable amb tres parells clau-valor on les claus siguen números i els valors cadenes. Imprimeix el valor associat a la clau 2.

????? info "Solució"

```
fun main() {
    val mapa = mapOf(1 to "Un", 2 to "Dos", 3 to "Tres")
    println("Valor per a clau 2: ${mapa[2]}")
}
```

4.18 Exercici 8: Recorregut d'un array amb índexs

Enunciat: Escriu un programa que recórrega un array d'enters utilitzant `withIndex()` i mostre tant l'índex com el valor de cada element.

????? info "Solució"

```
fun main() {
    val numeros = arrayOf(10, 20, 30, 40)
    for ((index, valor) in numeros.withIndex()) {
        println("Índex: $index, Valor: $valor")
    }
}
```

4.19 Exercici 9: Operacions amb llistes mutables

Enunciat: Crea una llista mutable de números. Elimina l'últim element de la llista i imprimeix la nova llista.

????? info "Solució"

```
fun main() {
    val llista = mutableListOf(1, 2, 3, 4, 5)
    llista.removeAt(llista.size - 1)
    println(llista)
}
```


Copyright - Sebastian Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)

4.20 Exercici 10: Mapa mutable

Enunciat: Crea un mapa mutable amb tres parells clau-valor. Afig un nou element al mapa i imprimeix tot el contingut.

????? info "Solució"

```
fun main() {
    val mapa = mutableMapOf(1 to "Un", 2 to "Dos", 3 to "Tres")
    mapa[4] = "Quatre"
    println(mapa)
}
```

5. Solucions

????? info "Solucions" Les solucions a tots els exercicis es troben a continuació.

Copyright - Sebastian Professor Departament d'Informàtica IES Jaume II el Just (Tavernes de la Valldigna)