

3 - Tipos de variables

Una variable es un depósito donde hay un valor. Consta de un nombre y pertenece a un tipo.

En el lenguaje Kotlin si necesitamos almacenar un valor numérico entero podemos definir una variable de tipo:

```
Byte
Short
Int
Long
```

Según el valor máximo a almacenar utilizaremos alguna de estos cuatro tipos de variables enteras. Por ejemplo en una variable de tipo Int podemos almacenar el valor máximo: 2147483647 y en general tenemos:

Tipo de variable	mínimo	máximo
Byte	-128	+127
Short	-32 768	+32 767
Int	-2 147 483 648	+2 147 483 647
Long	-9 223 372 036 854 775 808	+9 223 372 036 854 775 807

Si tenemos que almacenar un valor con parte decimal (es decir con coma como puede ser el 3.14) debemos utilizar una variable de tipo:

```
Double
Float
```

El tipo Double tiene mayor precisión que el tipo Float.

Y otro tipo de variables que utilizaremos en nuestros primeros ejercicios serán las variables de tipo String que permiten almacenar un conjunto de caracteres:

```
String
```

Una variable en Kotlin puede ser inmutable, esto significa que cuando le asignamos un valor no puede cambiar más a lo largo del programa, o puede ser mutable, es decir que puede cambiar el dato almacenado durante la ejecución del programa.

Para definir una variable en Kotlin inmutable utilizamos la palabra clave val, por ejemplo:

```
val edad: Int
edad = 48
val sueldo: Float
sueldo = 1200.55f
val total: Double
total = 7000.24
val titulo: String
titulo = "Sistema de Ventas"
```

Hemos definido cuatro variables y le hemos asignado sus respectivos valores.

Una vez que le asignamos un valor a una variable inmutable su contenido no se puede cambiar, si lo intentamos el compilador nos generará un error:

```
val edad: Int
edad = 48
edad = 78
```

Si compilamos aparece un error ya que estamos tratando de cambiar el contenido de la variable edad que tiene un 48. Como la definimos con la palabra clave val significa que no se cambiará durante toda la ejecución del programa.

En otras situaciones necesitamos que la variable pueda cambiar el valor almacenado, para esto utilizamos la palabra clave `var` para definir la variable:

```
var mes: Int
mes = 1
// algunas líneas más de código
mes = 2
```

La variable `mes` es de tipo `Int` y almacena un 1 y luego en cualquier otro momento del programa le podemos asignar otro valor.

En el lenguaje Kotlin si necesitamos almacenar un único caracter podemos definir una variable de tipo `Char`:

```
Char
```

Por ejemplo si queremos almacenar una variable mutable con el caracter 's':

```
var continua:Char='s'
```

Podemos hacer que lo infiera el compilador de Kotlin:

```
var continua='s'
```

Por último si queremos almacenar un valor de tipo lógico en Kotlin se necesita definir una variable de tipo `Boolean` que puede almacenar solo alguno de los dos valores (`true` o `false`):

```
var fin:Boolean=false
```

Problema

Crear un programa que defina dos variables inmutables de tipo `Int`. Luego definir una tercer variable mutable que almacene la suma de las dos primeras variables y las muestre. Seguidamente almacenar en la variable el producto de las dos primeras variables y mostrar el resultado.

Realizar los mismos pasos vistos anteriormente para crear un proyecto y crear el archivo `Principal.kt` donde codificar el programa respectivo (Si tenemos abierto el IntelliJ IDEA podemos crear un nuevo proyecto desde el menú de opciones: `New -> Project`)

Proyecto2 - Main.kt

```
fun main(parametro: Array<String>) {
    val valor1: Int
    val valor2: Int
    valor1 = 100
    valor2 = 400
    var resultado: Int
    resultado = valor1 + valor2
    println("La suma de $valor1 + $valor2 es $resultado")
    resultado = valor1 * valor2
    println("El producto de $valor1 * $valor2 es $resultado")
}
```

Definimos e inicializamos dos variables `Int` inmutables (utilizamos la palabra clave `val`):

```
val valor1: Int
val valor2: Int
valor1 = 100
valor2 = 400
```

Definimos una tercer variable mutable también de tipo Int:

```
var resultado: Int
```

Primero en la variable resultado almacenamos la suma de los contenidos de las variables valor1 y valor2:

```
var resultado: Int
resultado = valor1 + valor2
```

Para mostrar por la Consola el contenido de la variable \$resultado utilizamos la función println y dentro del String que muestra donde queremos que aparezca el contenido de la variable le antecedimos el caracter \$:

```
println("La suma de $valor1 + $valor2 es $resultado")
```

Es decir en la Consola aparece:

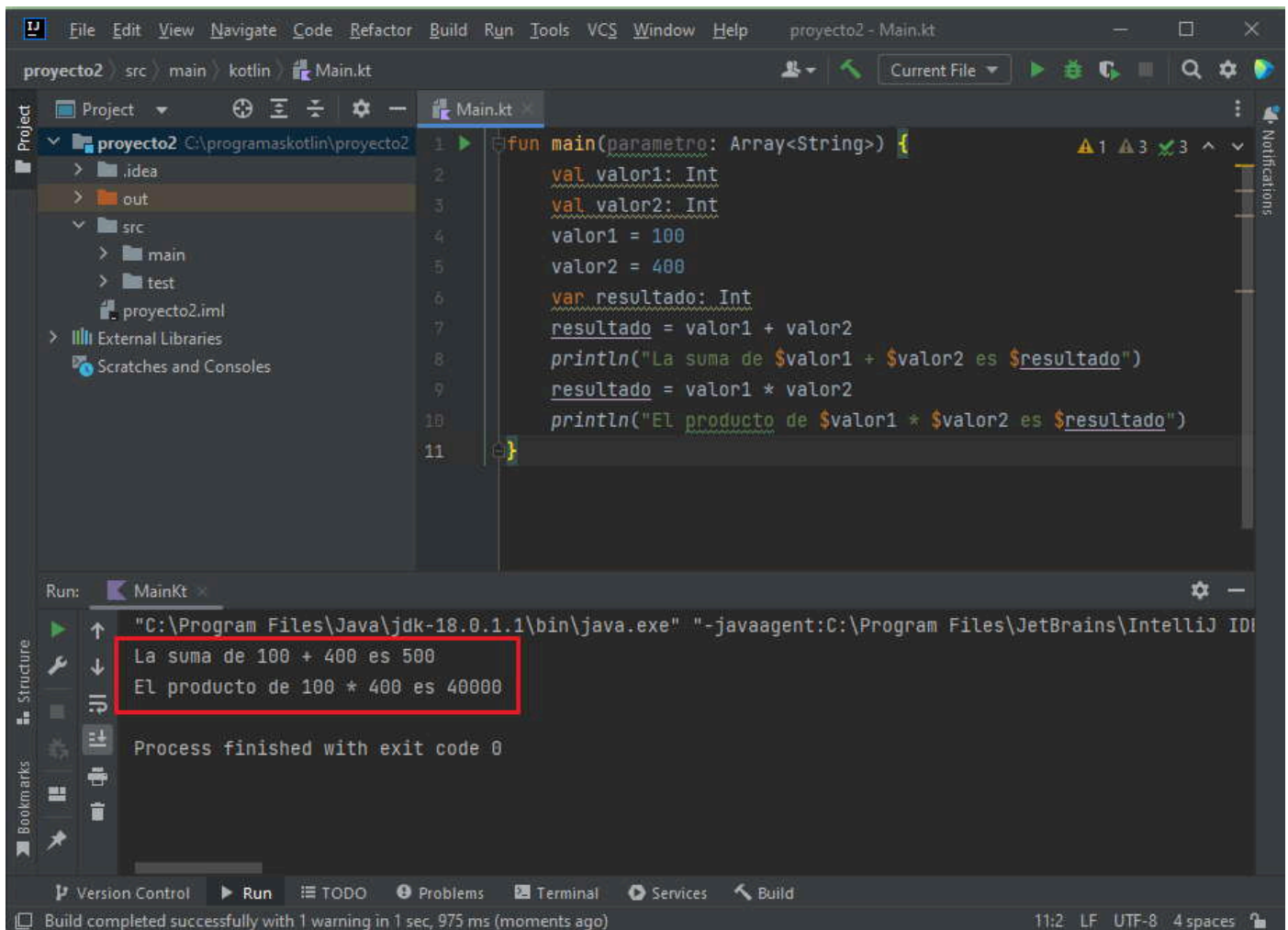
```
La suma de 100 + 400 es 500
```

Como la variable resultado es mutable podemos ahora almacenar el producto de las dos primeras variables:

```
resultado = valor1 * valor2
println("El producto de $valor1 * $valor2 es $resultado")
```

kotlin sustituye todas las variables por su contenido en un String.

El resultado de la ejecución de este programa será:



Conciso

Si entramos a la página oficial de Kotlin (<https://kotlinlang.org/>) podemos ver que una de sus premisas es que un programa en Kotlin sea "CONCISO" (es decir que se exprese un algoritmo en la forma más breve posible)

Haremos un primer cambio al Proyecto2 para que sea más conciso:

```
fun main(parametro: Array<String>) {  
    val valor1: Int = 100  
    val valor2: Int = 400  
    var resultado: Int = valor1 + valor2  
    println("La suma de $valor1 + $valor2 es $resultado")  
    resultado = valor1 * valor2  
    println("El producto de $valor1 * $valor2 es $resultado")  
}
```

En este primer cambio podemos observar que en Kotlin podemos definir la variable e inmediatamente asignar su valor. Podemos asignar un valor literal como el 100:

```
val valor1: Int = 100
```

o el contenido de otras variables:

```
var resultado: Int = valor1 + valor2
```

Otra paso que podemos dar en Kotlin para que nuestro programa sea más conciso es no indicar el tipo de dato de la variable y hacer que el compilador de Kotlin lo infiera:

```
fun main(parametro: Array<String>) {  
    val valor1 = 100  
    val valor2 = 400  
    var resultado = valor1 + valor2  
    println("La suma de $valor1 + $valor2 es $resultado")  
    resultado = valor1 * valor2  
    println("El producto de $valor1 * $valor2 es $resultado")  
}
```

El resultado de compilar este programa es lo mismo que los anteriores. El compilador de Kotlin cuando hacemos:

```
val valor1 = 100
```

deduce que queremos definir una variable de tipo Int

Si en la variable valor1 almacenamos el número 5000000000, luego el compilador de Kotlin puede inferir que se debe definir una variable de tipo Long

```
val valor1 = 5000000000
```

Para trabajar con los valores decimales por inferencia debemos utilizar la siguiente sintaxis:

```
var peso = 4122.23 // infiere que es Double  
val altura = 10.42f // debemos agregarle la f o F al final para que sea un Float y no un Double
```

Muy fácil es para definir un String:

```
val titulo = "Sistema de Facturación"
```

Utilizaremos mucho esta sintaxis a lo largo del tutorial.

En Kotlin 1.3 o superior si no vamos a utilizar el parámetro que llega a la función 'main' podemos obviar el mismo:

```
fun main() {  
    val valor1 = 100  
    val valor2 = 400  
    var resultado = valor1 + valor2  
    println("La suma de $valor1 + $valor2 es $resultado")  
    resultado = valor1 * valor2  
    println("El producto de $valor1 * $valor2 es $resultado")  
}
```

Problemas propuestos

- Definir una variable inmutable con el valor 50 que representa el lado de un cuadrado, en otras dos variables inmutables almacenar la superficie y el perímetro del cuadrado.
Mostrar la superficie y el perímetro por la Consola.

- Definir tres variables inmutables y cargar por asignación los pesos de tres personas con valores Float. Calcular el promedio de pesos de las personas y mostrarlo.

Proyecto3

```
fun main(argumento: Array<String>){
    val lado = 50
    val perimetro = lado * 4
    val superficie = lado * lado
    print("El perímetro de un cuadrado de lado $lado es $perimetro y su superficie es $superficie")
}
```

Proyecto4

```
fun main(argumento: Array<String>) {
    val peso1 = 89.4f
    val peso2 = 67f
    val peso3 = 87.45f
    val promedio = (peso1 + peso2 + peso3) / 3
    println("El promedio de los tres pesos de personas es $promedio")
}
```