

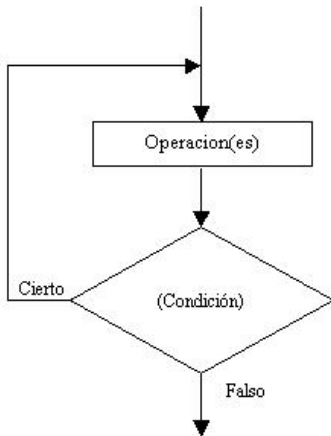
10 - Estructura repetitiva do/while

La estructura do/while es otra estructura repetitiva, la cual ejecuta al menos una vez su bloque repetitivo, a diferencia del while que podría no ejecutar el bloque.

Esta estructura repetitiva se utiliza cuando conocemos de antemano que por lo menos una vez se ejecutará el bloque repetitivo.

La condición de la estructura está abajo del bloque a repetir, a diferencia del while que está en la parte superior.

Representación gráfica:



El bloque de operaciones se repite MIENTRAS que la condición sea Verdadera.

Si la condición retorna Falso el ciclo se detiene.

Es importante analizar y ver que las operaciones se ejecutan como mínimo una vez.

Problema 1

Escribir un programa que solicite la carga de un número entre 0 y 999, y nos muestre un mensaje de cuántos dígitos tiene el mismo. Finalizar el programa cuando se cargue el valor 0.

Proyecto42 - Principal.kt

```
fun main(parametro: Array<String>) {
    do {
        print("Ingrese un valor comprendido entre 0 y 999:")
        val valor = readln().toInt()
        if (valor < 10)
            println("El valor ingresado tiene un dígito")
        else
            if (valor < 100)
                println("El valor ingresado tiene dos dígitos")
            else
                println("El valor ingresado tiene tres dígitos")
    } while (valor != 0)
}
```

En este problema por lo menos se carga un valor. Si se carga un valor menor a 10 se trata de un número de una cifra, si es menor a 100 se trata de un valor de dos dígitos, en caso contrario se trata de un valor de tres dígitos (no se controla que el operador cargue valores negativos o valores con más de tres dígitos). Este bloque se repite hasta que se ingresa en la variable valor el número 0 con lo que la condición de la estructura do while retorna falso y sale del bloque repetitivo finalizando el programa.

Problema 2

Escribir un programa que solicite la carga de números por teclado, obtener su promedio. Finalizar la carga de valores cuando se cargue el valor 0.

Cuando la finalización depende de algún valor ingresado por el operador conviene el empleo de la estructura do/while, por lo menos se cargará un valor (en el caso más extremo se carga 0, que indica la finalización de la carga de valores)

Proyecto43 - Principal.kt

```
fun main(parametro: Array<String>) {  
    var cant = 0  
    var suma = 0  
    do {  
        print("Ingrese un valor (0 para finalizar):")  
        val valor = readln().toInt()  
        if (valor != 0) {  
            suma += valor  
            cant++  
        }  
    } while (valor != 0)  
    if (cant != 0) {  
        val promedio = suma / cant  
        print("El promedio de los valores ingresados es: $promedio")  
    } else  
        print("No se ingresaron valores.")  
}
```

Acotaciones

Lo más común para incrementar en uno una variable Int es utilizar el operador ++:

```
cant++
```

Es lo mismo que escribir:

```
cant = cant + 1
```

También existe el operador -- que disminuye en uno la variable.

También en Kotlin se utiliza el operador += para la acumulación:

```
suma += valor
```

En lugar de escribir:

```
suma = suma + valor
```

En el problema 2 cada vez que se ingresa mediante un if verificamos si es distinto a cero para acumularlo y contarlo:

```
if (valor != 0) {
    suma += valor
    cant++
}
```

El ciclo do/while se repite mientras se ingrese un valor distinto a cero, cuando ingresamos el cero finaliza el ciclo y mediante un if verificamos si se cargó al menos un valor de información para obtener el promedio:

```
if (cant != 0) {
    val promedio = suma / cant
    print("El promedio de los valores ingresados es: $promedio")
} else
    print("No se ingresaron valores.")
```

Problema 3

Realizar un programa que permita ingresar el peso (en kilogramos) de piezas. El proceso termina cuando ingresamos el valor 0.

Se debe informar:

a) Cuántas piezas tienen un peso entre 9.8 Kg. y 10.2 Kg.?, cuántas con más de 10.2 Kg.? y cuántas con menos de 9.8 Kg.?

b) La cantidad total de piezas procesadas.

Proyecto44 - Principal.kt

```
fun main(parametro: Array<String>) {
    var cant1 = 0
    var cant2 = 0
    var cant3 = 0
    do {
        print("Ingrese el peso de la pieza (0 para finalizar):")
        val peso = readln().toDouble()
        if (peso > 10.2)
            cant1++
        else
            if (peso >= 9.8)
                cant2++
            else
                if (peso > 0)
                    cant3++
    } while(peso != 0.0)
    println("Piezas aptas: $cant2")
    println("Piezas con un peso superior a 10.2: $cant1")
    println("Piezas con un peso inferior a 9.8: $cant3");
    val suma = cant1 + cant2 + cant3
    println("Cantidad total de piezas procesadas: $suma")
}
```

Los tres contadores cant1, cant2, y cant3 se inicializan en 0 antes de entrar a la estructura repetitiva.

A la variable suma no se la inicializa en 0 porque no es un acumulador, sino que guarda la suma del contenido de las variables cont1, cont2 y cont3.

La estructura se repite mientras no se ingrese el valor 0 en la variable peso. Este valor no se lo considera un peso menor a 9.8 Kg., sino que indica que ha finalizado la carga de valores por teclado.

Debemos en la condición del do/while comparar el contenido de la variable peso que es de tipo Double con el valor 0.0 (de esta forma indicamos que el valor es Double, no podemos poner solo 0)

Problemas propuestos

- Realizar un programa que acumule (sume) valores ingresados por teclado hasta ingresar el 9999 (no sumar dicho valor, indica que ha finalizado la carga). Imprimir el valor acumulado e informar si dicho valor es cero, mayor a cero o menor a cero.
- En un banco se procesan datos de las cuentas corrientes de sus clientes. De cada cuenta corriente se conoce: número de cuenta y saldo actual. El ingreso de datos debe finalizar al ingresar un valor negativo en el número de cuenta.

Se pide confeccionar un programa que lea los datos de las cuentas corrientes e informe:

a) De cada cuenta: número de cuenta y estado de la cuenta según su saldo, sabiendo que:

```
Estado de la cuenta      'Acreeedor' si el saldo es >0.  
                          'Deudor' si el saldo es <0.  
                          'Nulo' si el saldo es =0.
```

b) La suma total de los saldos acreedores.

Proyecto45

```
fun main(parametro: Array<String>) {
    var suma = 0
    do {
        print("Ingrese un valor (finalizar con 9999):")
        val valor = readln().toInt()
        if (valor != 9999)
            suma += valor
    } while (valor != 9999)
    println("El valor acumulado es $suma")
    if (suma==0)
        println("El valor acumulado es cero.")
    else
        if (suma>0)
            println("El valor acumulado es positivo.")
        else
            println("El valor acumulado es negativo")
}
```

Proyecto46

```
fun main(parametro: Array<String>) {
    var suma = 0.0
    do {
        print("Ingrese número de cuenta:")
        val cuenta = readln().toInt()
        if ( cuenta >= 0) {
            print("Ingrese saldo:")
            val saldo = readln().toDouble()
            if (saldo > 0) {
                println("Saldo Acreedor.")
                suma += saldo
            } else
                if (saldo < 0)
                    println("Saldo Deudor.")
                else
                    println("Saldo Nulo.")
        }
    } while(cuenta>=0)
    println("Total de saldos Acreedores: $suma")
}
```