



Universidad Nacional de Ingeniería

INGENIERÍA EN COMPUTACIÓN

 **Kotlin**
Desarrollo móvil con Android

Laboratorio 2. Lenguaje de Programación KOTLIN

Facilitador: MSc. Glenda Barrios Aguirre

Objetivos:

1. Conocer el Lenguaje y realizar ejercicios de programación básica que incluyan variables y sentencias de control
2. Explorar las ventajas de la Programación Orientada a Objetos.

Ejercicios:

1. Cree una función en Kotlin para calcular el área de un triángulo, la función debe retornar el área y recibir como parámetro la base y la altura. Impleméntelo usando variables de tipo val y funciones de conversión de entero a float.
2. Cree una función en Kotlin para calcular el área de un círculo, la función debe recibir como parámetro el radio y retornar el área calculada, el valor de PI debe estar definido como una constante. Y el parámetro debe solicitarse al usuario, además debe validarse el nulo con el operador elvis . Abajo un ejemplo de cómo solicitar un parámetro.

```
val num:Int?  
  
print("Ingrese un número: ")  
num= readLine()?.toIntOrNull()  
println(num)
```

3. Escribe un programa que calcule el salario semanal de un colaborador en base a las horas trabajadas, a razón de \$12 la hora. En la medida de lo posible haga uso de los métodos asociados a la variable.

4. Haciendo uso de los Strings y los caracteres de tabulación escriba un programa que muestre en pantalla 10 palabras en inglés con su respectiva traducción en español, deben estar distribuidas en dos columnas y alineadas a la izquierda. Pruebe con los caracteres de tabulación
5. Haciendo uso de las sentencias de control condicionales escriba un programa que pida una hora por teclado y que muestre como resultado Buenos días, Buenas tardes o Buenas noches, según la hora. Tome en cuenta que de 06 a 12 es mañana, de las 13 a las 20 es tarde y fuera de los rangos anteriores, noche.
6. Escribe un programa que dado un número del 1 a 7 escriba el correspondiente nombre del día de la semana.
7. Escribe un programa que ordene tres números enteros especificados como parámetro a la función
8. Escribe un programa que calcule el salario semanal de un trabajador teniendo en cuenta que las horas ordinarias (40 primeras horas de trabajo) se pagan a \$12 dólares la hora. A partir de la hora 41, se pagan a 16. El programa debe recibir como parámetro el total de horas trabajadas a la semana.
9. Muestra los números múltiplos de 5 de 5 a 100 utilizando un bucle do-while.
10. Muestra los números del 320 al 160, contando de 20 en 20 hacia atrás utilizando un bucle for.
11. Muestra la tabla de multiplicar de un número especificado como parámetro
12. Escribe un programa que lea un número **n** e imprima una pirámide de números con **n** filas como en la siguiente figura:
1
121
12321
1234321
13. Escribe un programa en el que se declaren las variables enteras x e y. Asígnale los valores 144 y 999 respectivamente. A continuación, muestra por pantalla el valor de cada variable, la suma, la resta, la división y la multiplicación haciendo uso de los métodos de cada variable.
14. Crea las variables nombre, dirección y teléfono y asígnale los valores correspondientes. Muestra los valores de esas variables de tal forma que el resultado del programa sea visible en una sola línea

15. Escribe un programa que calcule el promedio y la suma de un arreglo de 10 números enteros.
16. Haciendo uso de lambdas, escriba una función que calcule la suma, el producto y el promedio de tres números.
17. Escriba un programa que determine si dos números son amigos. Dos números enteros positivos se consideran amigos si la suma de los divisores de uno es igual al otro número y viceversa. Por ejemplo, los números 220 y 284 son amigos. Los divisores del número 220 son: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, y suman 284. Los divisores de 284 son: 1, 2, 4, 71 y 142, que suman 220.
18. Escriba un programa que, dado un array de números enteros, determine cuáles son los elementos que se encuentran duplicados. Ejemplo de la salida esperada.

```
Elementos del array
Elemento [0] = 1
Elemento [1] = 2
Elemento [2] = 3
Elemento [3] = 3
Elemento [4] = 4
Elemento [5] = 4
Elemento [6] = 5
Elemento [7] = 2
Elemento duplicado: 2
Elemento duplicado: 3
Elemento duplicado: 4
```

Programación Orientada a Objetos

19. Imprima en pantalla los valores del objeto Persona desde un constructor. Los atributos de la clase **Persona** deben tener: Nombres, Apellidos, Cedula, Sexo y AnioNacimiento.
20. Cree una clase abstracta denominada Figura, la cual deberá tener 4 subclases denominadas: Triangulo, Cuadrado, Circulo y Rectángulo. La clase Figura debe tener la propiedad Nombre, la cual debe asignarse a través de las subclases, y el método área (), éste será abstracto y su implementación deberá hacerse desde las clases hijas, con la siguiente especificación:
 - a. Área del triángulo = $\text{base} * \text{altura} / 2$
 - b. Área del circulo = $\text{PI} * r \text{ al cuadrado}$
 - c. Área del rectángulo = $\text{Base} * \text{altura}$
 - d. Área del cuadrado = $\text{lado} * \text{lado}$

Mande a imprimir los valores, especificando como datos de entrada lo especificado debajo.

figura1: Círculo
radio = 2

figura2: Rectángulo
base = 1 altura = 2

figura3: Cuadrado
lado = 3

figura4: TriánguloRectángulo
base = 3 altura = 5

21. Creen una clase llamada **Insecto**, que cuente con los siguientes atributos o características: longitud, #alas, #patas, color, #antenas, tipoinsecto. El primer atributo debe ser un float, el color y el tipoinsecto de tipo **string** y el resto valores enteros. Escriba las líneas de código que permitan especificar valores a la clase de manera privada e imprimir dichos valores en pantalla.
22. Cree una clase de tipo DataClass denominada **Clase**, los datos que requiere manejar deben ser: profesor, materia, laboratorios, Conferencias. Realice las pruebas necesarias que le permitan asignarle valores e imprimir los resultados en pantalla.