

# TAREA

## Nº 2

# UNIDAD

# 3

### PROGRAMACIÓN ORIENTADA A OBJETOS

**Asignatura:**

PROGRAMACIÓN ORIENTADA A OBJETOS

**UNIDAD 3:**

CLASES Y OBJETOS

**Objetivo de la Tarea**

Conocer los conceptos de clases, objetos y métodos y ponerlos en prácticas con el lenguaje de programación Python

**INSTRUCCIONES**

1. Realizar los ejercicios propuestos en el archivo adjunto.

**EJERCICIO 1**

Se solicita por teclado el ingreso de información de personas, definir su clase con mínimo 4 atributos entre los cuales se solicitará la fecha de nacimiento. Luego en una opción consultar ordene los objetos en la lista de personas de forma ascendente por edad la misma que será calculada a partir de la fecha de nacimiento antes ingresada. Luego muestre los datos por pantalla.

**Nota:** realizar las validaciones correspondientes en el ingreso de los datos.

**EJERCICIO 2**

Desarrollar un programa que conste de una clase padre Persona y tres subclases médico, enfermero y paciente. Definir los atributos nombres, apellidos, fecha nacimiento, edad (se deberá crear un método para calcularla a partir de la fecha de nacimiento) y un método para imprimir los datos en la clase Persona.

1. La clase médico heredaré los datos de la clase persona y tendrá cuatro atributos propios: especialidad, celular, código de registro profesional y sueldo mensual; mostrar el detalle de pago de impuestos según lo siguiente:
  - Si el sueldo anual es mayor a 68000 deberá hacerle una retención de impuestos del 15%.
  - Si el sueldo anual es entre 40000 y 68000 deberá hacerle una retención de impuestos del 8.5%.
  - Si el sueldo anual es menor a 40000 deberá hacerle una retención de impuestos del 3.2%.

Mostrar el valor de impuestos a pagar, según el descuento aplicado, además mostrar el total anual con y sin la reducción de impuestos; crear el método correspondiente para mostrar la información.

2. La clase enfermero heredará los datos de la clase persona y tendrá dos atributos propios: número de vacunas, experiencia profesional; además crear un método para mostrar la información.
3. La clase paciente también heredará los datos de la clase padre y se definirán los siguientes atributos propios: peso, talla, temperatura, tipo de sangre, motivo consulta y síntomas; además crear un método para mostrar la información.

Las clases deberán tener la capacidad de asignar y modificar sus valores iniciales por medio de los métodos getters y setters.

**Nota:** realizar las validaciones correspondientes en el ingreso de los datos.

### EJERCICIO 3

Construir una clase Estudiante la cual tendrá los siguientes atributos: Cédula, Nombres, Apellidos, Asignatura y además que pida por teclado la cantidad de notas a ingresar (validando que sea mayor a 3 ingresos), si esto no es así el programa no debe terminar sino seguir pidiendo la cantidad de notas a ingresar, pero antes mostrar un mensaje por pantalla alertando al usuario. Luego del ingreso correspondiente de la cantidad de notas, solicitar dichas notas para que permita calcular el promedio y muestre el mismo; además un mensaje según la siguiente tabla:

- Si el promedio es de 69.50 hasta 100 mostrar Aprobado
- Si el promedio es de 39.50 hasta menor a 69.50 mostrar Suspenso
- Si el promedio es menor a 39.50 mostrar Reprobado

Cabe indicar que la nota aceptada en el ingreso debe ser de entre 0 y 100; el programa sólo debe terminar cuando se ingresen la cantidad de notas solicitadas y se muestre su promedio con el mensaje de la tabla arriba descrita.

La clase deberá de tener la capacidad de asignar y modificar sus valores iniciales mediante los métodos GETTERS y SETTERS

**Nota:** realizar las validaciones correspondientes en el ingreso de los datos.

2. Adjuntar el archivo \*.py
3. Consultar los materiales de la plataforma (documento base de lectura e infografía, sobre el tema “CLASES Y OBJETOS”).
4. Adjuntar el archivo \*.py

5. Entrega el archivo de Python por medio del espacio de tareas en la plataforma en el tiempo establecido