Práctico 1

Utilizaremos el lenguaje Python *(y/o algún otro que defina el docente)* para la resolución de los ejercicios a continuación. Considere escribir el código respetando la convención de escritura del lenguaje seleccionado. Por ejemplo, en el caso de usar Python, recuerde o revise la documentación PEP-8

Además, se debe considerar:

- Crear el código fuente y guardarlo en un archivo del tipo apellido-nombre-tp01ej01.py
- Comentar el código fuente a fin de dar una mayor legibilidad
- La salida por pantalla debe contener toda la información necesaria para el usuario, a fin de que sea entendible el programa
- Debe figurar en la parte superior de la pantalla el apellido y nombre del estudiante, antes de la ejecución del programa, como una cabecera con información del autor del programa.

Paradigma Orientado a Objetos

1. **Rectángulos.**
   - *a*) Crear una clase llamada *Rectangulo*.
   - *b*) La clase debe contener los atributos: base y altura.
   - *c*) Crear los métodos como: calculo_area y calculo_perimetro.
   - *d*) Se debe crear el objeto correspondiente, y se le debe pedir al operador que ingrese los datos necesarios para que el programa calcule el área y el perímetro.
2. **Círculos.**
   - *a*) Crear una clase llamada *Circulo*.
   - *b*) La clase debe contener el atributo: radio.
   - *c*) Crear los métodos como: calculo_area y calculo_circunferencia.
   - *d*) Se debe crear el objeto correspondiente, y se le debe pedir al operador que ingrese los datos necesarios para que el programa calcule el área y la circunferencia.
3. **Liquidación de Sueldos.**
   - *a*) Crear una clase llamada *Empleado*.
   - *b*) La clase debe contener los atributos: nombre, horas_trabajadas y tarifa_hora.
   - *c*) Crear el método: calcular_salario.
   - *d*) Se debe crear el objeto correspondiente, y se le debe pedir al operador que ingrese los datos necesarios para que el programa calcule el sueldo que le corresponde cobrar en función de las horas trabajadas y el valor de la hora trabajada.
4. **Cálculo del Indice de Masa Corporal (IMC).**
   - *a*) Crear una clase llamada *Persona*.

Lic. Carina Povarchik
Lic. Ariel Villar

b) La clase debe contener los atributos: nombre, peso y altura.

c) Crear el método: calcular_imc.

d) Se debe crear el objeto correspondiente, y se le debe pedir al operador que ingrese los datos necesarios para que el programa calcule el IMC *(cálculo de IMC investigar en fuentes confiables de salud).*

e) Nota: El IMC es una medida que se utiliza para determinar si una persona tiene un peso saludable en relación con su altura. Se calcula dividiendo el peso de una persona en kilogramos por el cuadrado de su altura en metros.

5. **Conversor de Distancias.** Se desea implementar un conversor de metros a kilómetros, y centímetros.

   a) Crear una clase llamada $ConversorMetros$.

   b) La clase debe contener el atributo: metros.

   c) Crear el método: a_km

   d) Crear el método: a_cm

   e) Se debe crear el objeto correspondiente, y se le debe pedir al operador que ingrese los datos necesarios para que el programa calcule la conversion de metros a kilometros y de metros a centimetros.

6. **Computo**. Se deberá crear una clase llamada *Computo*, con su constructor por defecto sin parámetros, que permitirá realizar una serie de cálculos sobre un número entero n. Deberá crear los siguientes métodos:

   a) Un método llamado $factorial$ que permita calcular el factorial de n. Deberá testear el método instanciando la clase como en todos los ejercicios.

   b) Un método llamado *suma* que calcule la suma de los primeros n números enteros.

   c) Un método llamado *es_primo* que permita testear si el número es primo o no.

   d) Un método *tabla_multiplicacion* que creará y mostrará la tabla de multiplicar de n.

   e) Un método *lista_divisores* que obtenga todos los divisores de n y los retorne en un arreglo o lista.

7. **Facturador.** Crear una clase llamada *Facturador* con los siguientes atributos: fechafacturacion, nombrecliente, detalle, preciounitario, cantidad, preciototal. Crear los eventos para facturar mercaderias varias y crear evento para mostrar la factura por pantalla, pedir que el usuario ingrese los datos. comentar el codigo y la salida por pantalla. Deberá crear los siguientes métodos:

   a) Un método llamado $facturar\_mercaderia$. El cual deberá realizar el proceso de facturacion, es decir, se deberá ingresar: fecha factura, nombre de cliente, detalle producto, cantidad, precio unitario y calcular precio total.

   b) Un método llamado $mostrar\_factura$ por pantalla.

8. **Mazo de Cartas.** Crear una clase llamada *MazoCartas*, que representará un mazo de 52 cartas de póker. Deberá usar internamente otra clase, que también deberá crear llamada *Carta* que modelará una carta de póker propiamente dicha.

   a) La clase *MazoCartas* tendrá un método *repartir* que robará una carta del mazo, eliminándola de éste.

   b) Un método llamado *barajar* que mezcle nuevamente las cartas y se asegure que estén las 52.

   c) La clase *Carta*, además, deberá tener los siguientes atributos: palo *(Corazones, Diamantes, Tréboles, Picas)* y valor. *(A,2,3,4,5,6,7,8,9,10,J,Q,K)*

9. **Heroe** Crear una clase llamada *Heroe*, que representará un guerrero de un juego arcade en 2D. Deberá usar internamente otra clase, que también deberá crear llamada *Inventario* que modelará la mochila o inventario del guerrero. En el programa principal habrá una Despensa, donde habrá una lista de elementos disponibles, puede considerar asignarles además un precio a cada elemento y agregar dinero al Heroe. El Heroe podrá tomar elementos de la Despensa, quitándolos de la lista disponible. También podría agregar interes a dicha lista considerando las cantidades disponibles por elemento y/o su precio. Elementos posibles podrían ser Pocion, Flechas, Guantes, Escudo, etc

   a) El Heroe podrá realizar las siguientes acciones: tomar un elemento de la despensa (o comprarlo según la versión que ud decida implementar) Podrá Usar un elemento que ya tiene, o dejar de usarlo y ponerlo en su mochila. Tendrá niveles de sangre o vida, y niveles de magia.

   b) El Heroe tendrá atributos como sangre o vida, nivel de magia, daño, daño crítico por ejemplo para determinar cuantas posibilidades de hacer mucho daño tiene en cada ataque. Hambre, que lo haría descansar por ejemplo. Nombre, Guilda. Puede agregar o modificar algunos de los atributos mencionados en pos de diseñar su Heroe.

10. Responda las siguientes preguntas éstas refieren a terminología básica de POO, marcando las que considere correctas o explicándolas:
    ☐ ¿Puedo tener más de una clase en un archivo?
    ☐ Que es una Instancia?
    ☐ ¿Clase e Instancia, son lo mismo?