

ICOMDS-ME

Guía 3: Clases.

Ejercicio 0

Cree una clase `Persona`, que tenga como atributos `Nombre`, `Apellido`, `Mail`, y `Correo`. Luego, cree dos clases más: `Alumno` y `Profesor`, que además de los atributos anteriores tengan `Calificaciones` o `Cursos` según corresponda.

Ejercicio 1

A partir de la clase `vector2D`, cree una clase `Polígono` que pueda tomar n vértices y calcule su perímetro y área.

Nota: ¿cómo calculamos el área de un polígono arbitrario?

Ejercicio 2

Cree una clase `Sistema`, que permita cargar fuerzas y puntos de aplicación. Esta puede ser bidimensional o tridimensional. Luego, cree métodos para obtener la fuerza resultante y el torque resultante, desde el origen, o un punto introducido por el usuario.

Ejercicio 3

Cree una clase `Movimiento`, que represente la cinemática de una partícula en el tiempo. Esta debe tener N divisiones temporales, un espaciado entre tiempos dt , y un array de `vector2D` o `vector3D` para la posición, velocidad y tiempo. Luego, cree otras clases `MRU`, `MRUV` y `MCU` que hereden de la clase `Movimiento`, y permitan obtener, a partir de la información adecuada toda la información de las partículas en movimiento.

Ejercicio 4

Utilizando la clase `Sistema` del **Ej. 2**, implemente una clase `Viga` con la aproximación que considere adecuada. La misma requerirá no sólo fuerzas y puntos de aplicación, sino posiciones de apoyos, tipos, inclinación; además de tipo de perfil y dimensiones. Luego, escriba métodos que permitan:

- Calcular las reacciones de vínculos.
- Calcular el esfuerzo de corte y momento flector, discretizando la viga en N elementos.
- Definir qué altura de perfil es necesaria para resistir el esfuerzo.

Para ello necesitará usar tablas de alturas de perfiles.

Ejercicio 5

Escriba una clase `Transformador`. Este debe tener propiedades como `tensionPrimario`, `tensionSecundario`, `frecuencia`, `material`, etcétera. Además, la misma debe acceder a la información en las tablas de conductores, núcleos y carretes. Por último, escribir métodos que permitan dimensionar adecuadamente el núcleo, los conductores y sus pesos.

Se adjuntan las tablas necesarias.

Desafío 1

3. Cluster de Estrellas

En astronomía, la distancia entre estrellas cercanas es tan grande que un conjunto de estrellas puede ser modelado como un arreglo de masas puntuales definidas por sus coordenadas y sus masas.

Dado un conjunto de estrellas, se solicita implementar funciones para calcular el centro de masa y el momento de inercia respecto de un eje que pase por el centro de masas.

Las coordenadas del centro de masa se pueden calcular como:

$$CM_x = \frac{\sum_i m_i x_i}{\sum_i m_i} \quad CM_y = \frac{\sum_i m_i y_i}{\sum_i m_i} \quad CM_z = \frac{\sum_i m_i z_i}{\sum_i m_i}$$

Donde x_i, y_i, z_i son las coordenadas de la estrella i y m_i su masa.

Por otro lado, el momento de inercia respecto al eje Z que pasa por el centro de masa se define como $I = \sum_i m_i d_i^2$ donde d_i es el módulo del vector bidimensional $[x_i - CM_x, y_i - CM_y]$. Termine el diseño e implementación de los siguientes UDT para representar un punto en 3D, una estrella y un cluster de estrellas.

```
class Point3D {
    // TODO
};

class Star {
    // TODO
};

class StarCluster {
public:
    // TODO
    Point3D centroMasa(); // calcula el centro de masa del cluster
    double inerciaZ(); // calcula el momento de inercia respecto al eje Z
private:
    // TODO
};
```

Desafío 2

Escriba una clase `Circuito`. La misma debe tener una `Fuente` (en principio, alterna) y N `Componentes` conectados **en paralelo**. Estos deben ser Resistencias, Inductancias o Capacitancias.

Escriba un método que permita obtener la corriente y potencia para cada elemento, y la corriente y potencia total. Además, escriba un método que permita obtener el capacitor necesario para corregir el $\cos \varphi$.

Si se anima, dé la opción de que el circuito sea *monofásico* o *trifásico* (con cargas equilibradas).

Desafío 3

Escriba una clase `Estado` que represente un punto en el espacio termodinámico (p, v, T) . Luego, escriba una clase `Evolución`, que contenga un estado inicial y un estado final. A partir de estas clase, escriba cinco clases que la hereden: *Isotérmica*, *Adiabática*, etc. Después, escriba una clase `Ciclo`, que esté representada por un conjunto de evoluciones.

Según la información proporcionada, las clases que heredan de Evolución deben ser capaces de encontrar incógnitas en los datos de entrada. Luego, la clase Ciclo debe chequear si el ciclo es válido (i.e., empieza y termina en el mismo punto) y debe permitir calcular el Q_1 , Q_2 y W del ciclo.

In []:

1	
---	--