Programación 2024

Guía 9: Objetos

30 de octubre de 2024

Antes de comenzar los problemas genere un nuevo directorio `guia9` donde trabajará y guardará todos los programas y archivos que se producirán en este práctico.

Problema 1: Transformación de temperaturas (revisado)

(a) Realice una función que transforme de Celsius a Fahrenheit (reutilice el código de guías anteriores).

(b) Realice una funcion que transforme de Fahrenheit a Celsius (reutilice el código de guías anteriores).

(c) Implemente un programa que pregunte al usuario que transformacion desea y luego pregunte las temperaturas. Controle con `exception` cuando el usuario ingresa una temperatura que no corresponde (carácteres y temperatura fuera de rango).

Problema 2: Desarrolle una clase de objetos que trabaje con vectores de dimensión n (no use numpy pero si listas).

(a) Inicialice la clase definiendo el vector (la dimensión y el tipo).

(b) Implemente la función suma de vectores.

(c) Implemente la función producto interno.

(d) Implemente la función de la media .mean().

(e) Item implemente una función que determine si dos vectores son ortonormales usando la función del inciso anterior.

(f) Implemente la función rotación de vectores alrededor del eje z reutilizando la función desarrollada en la guia anterior.

Problema 3: Desarrolle una clase de objetos que trabaje con matrices cuadradas de dimensión n (no use numpy pero si listas de listas).

(a) Inicialice la clase definiendo la matriz.

(b) Implemente la función suma de matrices.

(c) Implemente la función que retorne una columna de la matriz.

(d) Implemente la función que retorne una fila de la matriz.

(e) Implemente la función de la media .mean().

(f) Implemente la función que realice la transpuesta (reutilizando lo realizado en la guia anterior).

Problema 4: Desarrolle una clase de objetos que trabaje con polinomios de grado n.

(a) Inicialice la clase definiendo el polinomio.

(b) Implemente el método suma de polinomios de grado n y m.

(c) Implemente el método que evalue el polinomio (por default).

(d) Desarrolle un método derivada del polinomio que devuelva la derivada.

(e) Implemente un método que grafique el polinomio y su derivada dado los puntos **x**.

(f) Implemente un método que imprima el polinomio con la forma usual.

Problema 5: Se desea implementar una clase que considere rectángulos. Un rectángulo es creado en una ubicación particular (x,y) especificando la esquina inferior izquierda del mismo; tiene un ancho y una altura.

a) Defina la clase *Rectangulo*, cuyos parámetros sean la ubicación del mismo, su ancho y su altura. Inicialice un objeto que represente un rectángulo en (27,45) de ancho 50 y altura 30.

b) Implemente un método que pertenezca a *Rectangulo* que calcule y devuelva el área del rectángulo.

c) Desarrolle un método que determine el perímetro del rectángulo.

Problema 6: Movimiento de una partícula.

a) Construya una clase llamada *AceleracionConstante* que permita calcular el movimiento en una dimensión con aceleración constante de una partícula con la ecuación de movimiento. El constructor guarda la posición, velocidad y aceleración iniciales. El llamado a la clase debe devolver la posición del objeto en un tiempo t, y un método llamado *velocidad* debe devolver la velocidad en un dado tiempo t.

b) Expanda la funcionalidad de la clase *AceleracionConstante* en una clase llamada *AceleracionLineal*, que pueda tratar también con casos en que la aceleración es un polinomio de primer orden de la forma

$$a(t) = a_0 + a_1\, t$$

donde j es el cambio en la aceleración por unidad de tiempo. Las ecuaciones de movimiento tendrán la forma

$$\begin{cases} x(t) = x_0 + v_0 t + \frac{1}{2} a_0 t^2 + \frac{1}{6} a_1\, t^3 \\ \\ v(t) = v_0 + a_0 t + \frac{1}{2} a_1\, t^2 \end{cases}$$

Implemente la clase *AceleracionLineal* que herede la funcionalidad de *AceleracionConstante* pero que tenga la habilidad extra de calcular la trayectoria cuando la aceleración sea lineal.