

Programación en Lenguaje Java

Problema 2.3. Practicar con expresiones



Michael González Harbour

Mario Aldea Rivas

Departamento de Matemáticas,
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Problema 2.3. Practicar con expresiones

Datos personales

Apellidos:

Nombre:

1 *Expresión aritmética que incluye un cambio de unidades*

Objetivos

Practicar con expresiones.

Descripción

Escribir en Java un método que realiza el cálculo de las siguientes expresiones matemáticas que calculan un valor para el atributo x , que representa la posición en un movimiento oscilatorio amortiguado. La letra e representa las constantes del mismo nombre.

$$\omega = \left[\frac{k}{m} - \left(\frac{b}{2m} \right)^2 \right]^{1/2}$$
$$x(t) = Ae^{-\frac{b}{2m}t} \cos(\omega t + \delta)$$

Suponer que los siguientes atributos están declarados en la clase y ya tienen valores asignados:

double x , k , m , b , amplitud

Al atributo amplitud se corresponde con el valor A de la fórmula. El tiempo t (en segundos) y el ángulo δ (en grados) se pasarán al método como parámetros. El valor ω se calculará con la fórmula indicada, con la que se obtienen rad/s, y se guardará en una variable local para usarlo en el cálculo de x .

Se pide:

El código Java de un método que calcula y almacena el nuevo valor para x usando las expresiones de arriba. *Nota:* observar que ω está en rad/s pero δ está en grados.

Respuesta:

<Poner aquí el código Java del método>

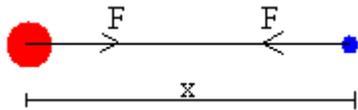
Programación en Java

2 Métodos que retornan expresiones

Objetivos

Practicar con un método que devuelve el valor de una expresión.

Descripción



La separación entre los átomos de una molécula diatómica se puede calcular como

$$x = x_0 + A \cdot \text{sen}(\omega_0 t + \phi)$$

siendo ω_0 expresada en radianes:

$$\omega_0 = \sqrt{\frac{k}{\mu}} = \sqrt{k \left(\frac{1}{m_1} + \frac{1}{m_2} \right)}$$

donde k es la constante de interacción, m_1 y m_2 son las masas de los átomos, A es la amplitud del movimiento, ϕ es la fase inicial y t es el tiempo.

Se dispone de la siguiente clase que almacena los atributos necesarios para representar este sistema y dispone de un constructor que da valor inicial a estos atributos.

```
/**
 * Clase que almacena los datos de una molécula diatómica
 *
 */
public class MoléculaDiatómica
{
    private double m1, m2; // masas de los átomos, en kg
    private double k; // constante de interacción, en N/m
    private double a; // amplitud del movimiento, en m
    private double fi; // fase inicial del movimiento, en grados

    /**
     * Constructor al que se le pasan m1, m2, k, A en unidades del sistema
     * internacional, y la fase fi en grados
     */
    public MoléculaDiatómica(double m1, double m2, double k, double a, double fi)
    {
        this.m1=m1;
        this.m2=m2;
        this.k=k;
        this.a=a;
        this.fi=fi;
    }
}
```

Programación en Java

Lo que se pide es escribir un método para esa clase que calcule y retorne la expresión del valor de la separación entre los átomos, x , dada la posición inicial x_0 y el tiempo t como argumentos.

Fuente: <http://www.sc.ehu.es/sbweb/fisica/oscilaciones/diatomica/diatomica.htm>

Respuesta que se pide

<Poner aquí el código Java del método>

3 Método que calcula una expresión no aritmética

Objetivo

Practicar métodos y expresiones no aritméticas

Descripción

Se desea saber si un carácter es una letra minúscula o mayúscula, o simplemente si es una letra.

Lo haremos comprobando si el carácter está entre la a y la z (mayúscula o minúscula), usando por ejemplo el operador de comparación $<$.

Se pide:

- un método al que se le pasa el carácter como parámetro y retorna un booleano que indica si el carácter es una letra minúscula o no.
- un método al que se le pasa el carácter como parámetro y retorna un booleano que indica si el carácter es una letra mayúscula o no.
- un método al que se le pasa el carácter como parámetro y retorna un booleano que indica si el carácter es una letra (minúscula o mayúscula) o no.

Respuesta:

<Poner aquí el código Java de los tres métodos>

4 Conversiones de tipos

Objetivos

Programación en Java

Practicar con las conversiones de tipos

Descripción

Dada esta clase Java que describimos con su diagrama de clase

Dibujo
+Dibujo (int alto, int ancho) +void pintaLinea(int x1, int y1, int x2, int y2)

Al constructor se le pasa el tamaño de la ventana en pixels. El método pintaLinea() pinta una línea del punto (x1,y1) al punto (x2,y2).

Escribir un método en una clase aparte, con la cabecera mostrada más abajo, que llamando a pintaLinea() dibuje en el objeto dib una línea que saliendo del punto A, cuyas coordenadas (ax, ay) se pasan a éste método, tenga una longitud lon, y un ángulo con la horizontal igual a ang (que está en grados). Las coordenadas del punto de destino serán (ax+lon*cos(ang), ay+lon*sen(ang)).

```
public void pinta(Dibujo dib, double ax, double ay, double lon, double ang)
```

Observar que serán necesarias conversiones de tipo de double a int y que las funciones trigonométricas, en Java, trabajan en radianes.

Respuesta que se pide

<Poner aquí el código Java del método>