

Problema 2: Tipos, clases y métodos, expresiones

Datos personales	
Apellidos:	Prieels
Nombre:	Cedric

1 Crear una clase sencilla

Objetivos

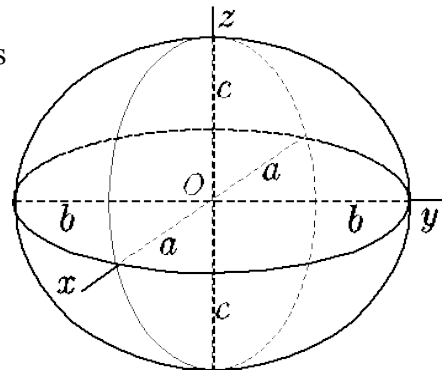
Practicar con la creación de una clase sencilla que tenga un método que da valor a los atributos y un método simple que retorna un dato.

Descripción

Escribir una clase denominada Elipsoide que permita almacenar las medidas de un elipsoide como el de la figura en los atributos a, b, y c, del tipo double y que tenga las siguientes operaciones:

- método constructor al que se le pasan como argumentos las tres medidas, y las almacena en los atributos
- operación que retorna el volumen del elipsoide, de acuerdo con la siguiente expresión:

$$\frac{4}{3}\pi abc$$



Respuesta:

<Poner aquí el código Java de la clase:>

2 Método con argumentos

Objetivos

Practicar con un método al que se pasa un argumento.

Descripción

Se dispone de la siguiente clase en Java que representa un polinomio de tercer grado, almacenando en 4 atributos sus coeficientes:

```
public class Polinomio3G {  
    private double a,b,c,d;  
}
```

Se pide añadir a la clase una operación que retorne el valor de la expresión ax^3+bx^2+cx+d , dado el valor de x que se pasa como argumento.

Programación, Curso 2015-2016

Nota:

Elevar al cuadrado se puede hacer multiplicando el dato por si mismo. a^2 : $a*a$

Elevar al cubo igual a^3 : $a*a*a$

Respuesta:

```
/**
 * Método que calcula el valor de un polinomio dado en el punto x. a, b, c y d son los
 * coeficientes del polinomio y tienen que ser definidos dentro de una clase.
 */

public double valorPoli (double x) {

    return (a*x*x*x + b*x*x + c*x + d);

}
```

3 *Compatibilidad de tipos*

Objetivo

Practicar con la compatibilidad de tipos.

Descripción

En Java no es posible asignar directamente un valor de un tipo a una variable de otro tipo si puede haber pérdida de información (al revés, sí). Por ejemplo:

- Un número real no se puede guardar en un entero
- Un entero largo (del tipo long) no cabe en un entero normal (del tipo int)
- Un número real de doble precisión (double) no cabe en una variable real del tipo float

Se parte de estas declaraciones de variables:

```
int i;
long n;
double x;
float g,f;
```

A continuación se muestra un fragmento de programa Java con instrucciones de asignación. Indicar breve y **razonadamente** cuáles de ellas son válidas y cuáles no.

```
n=2_000_000;
i=n;
n=i;
g=12.0+i;
x=12.0+i;
x=g;
f=1.23E20;
x=12.45f;
```

Respuesta:

- `n=2_000_000`; si el caracter '_' es valido, entonces funciona porque un long tiene un tamaño más grande que el simple int de la derecha
- `i=n`; no funciona, no se puede guardar un long en un int
- `n=i`; si que funciona, porque un entero normal cabe en un entero largo
- `g=12.0 + i`; no funciona, la suma de un double con un int no cabe en un float
- `x=12.0 + i`; funciona, el miembro de la derecha devuelve un double que cabe en x
- `x=g`; funciona porque el tamaño de un double es más grande que el tamaño de un float
- `f=1.23E20`; No cabe, un double (derecha) no cabe en un float
- `x=12.45f`; funciona porque tamaño double > tamaño float

4 Expresiones

Objetivo

Familiarizarse con expresiones trigonométricas en Java

Descripción

Dados los ángulos alfa (α) y beta (β) expresados en grados, así como las variables x, w y t:

```
double alfa, beta; // ángulos en grados
double w; frecuencia angular en radianes/segundo
double t; // tiempo en s
double x; // sin unidades
```

Escribir en Java las siguientes expresiones

$$w = \arctan(\alpha/\beta) \cdot \frac{2}{t}$$

$$\beta = a \sin(wt)$$

$$x = \sqrt{e^{t/2} \cos(wt)}$$

Respuesta:

- `w = Math.atan((Math.toRadians(alfa)/ Math.toRadians(beta)) * (2/t);` // aunque no creo que // sea necesario en este caso de cambiar alfa y beta a radianes, porque se dividen.
- `beta = alfa * Math.sin(w * t);` // No hay que cambiar w*t porque ya tiene unidades de rad
- `x = Math.sqrt(Math.exp(t/2) * Math.cos(w * t));`