

## INTRODUCCION POO

### **Actividad 01: Atributos de clase. Población.**

- Crear una clase Individuo con atributos **DNI** (de tipo cadena), **nombre** y **año de nacimiento**.
- Incorpora un constructor general que reciba como parámetros los 3 atributos anteriores.
- Incorporar métodos **getter()** y **setter()** a través de la utilidad que tendrán los IDE 's como Eclipse o Netbeans para automatizar dicho proceso (investiga para ver cómo hacerlo).
- Incorporar un *atributo estático* o de clase que controle la población (cantidad de individuos) y que llamaremos **poblacion**.
- Posteriormente, en el método principal, instanciaremos una serie de elementos Individuo, comprobando que el valor del atributo de clase poblacion se actualiza, efectivamente, con el número de individuos instanciados.

## Actividad 02: Clase Racional.

Comprueba que la clase siguiente funciona correctamente. Sin embargo, es mejorable en su diseño: no dispone de un constructor explícito, el método main está codificado en el mismo fichero fuente, no se especifican modificadores de acceso a los miembros de la clase, etc.. Modifica la aplicación subsanando los inconvenientes antes indicados.

```
class CRacional
{
    int numerador;
    int denominador;
    void asignarDatos(int num, int den)
    {
        numerador = num;
        if (den == 0) den = 1; // el denominador no puede ser cero
        denominador = den;
    }
    void visualizarRacional()
    {
        System.out.println(numerador + "/" + denominador);
    }

    public static void main (String[] args)
    {
        // Punto de entrada a la aplicación
        CRacional r1 = new CRacional(); // crear un objeto CRacional

        r1.asignarDatos(2, 5);
        r1.visualizarRacional();
    }
}
```

⇒ Esto implica realizarlo en dos archivos

### **Actividad 03: Clase Circulo.**

Crea una clase denominada Circulo con los siguientes datos:

Miembros dato:

- El radio.

Métodos:

- Dos constructores, uno que reciba el radio como argumento; y otro, que por defecto le asigne al radio el valor 2.
- Un método de instancia u objeto denominado calcularSuperficie que devuelva el área del círculo. Sabemos que el área es igual a PI multiplicado por el radio al cuadrado.  $PI = 3.141592$ .
- Utiliza el valor constante PI de la librería Math

Construye otro fichero denominado TestCirculo donde se arranque la aplicación. Crea un círculo de radio 10 llamado circulo1 y visualiza su área. A continuación, instancia otro círculo circulo2 cuyo radio sea el asignado por defecto y muestra igualmente su área.

## Actividad 04: Tarea UT05: Clase Rectangulo.

Diseña una clase denominada Rectangulo teniendo en cuenta la información siguiente:

Atributo	Significado
Base	Base del rectángulo.
Altura	Base del rectángulo.
Método	Significado
Rectangulo	Es el constructor de la clase. Inicia los miembros dato. Utiliza un constructor con argumentos, otro por defecto y un tercero que haga una copia del objeto
setBase	Permite asignar la base.
getBase	Retorna la base.
setAltura	Asigna el número de altura.
getAltura	Retorna la altura.
getArea	Retorna el área.
getPerimetro	Retorna el perímetro.

Incorpora en el proyecto una clase TestRectangulo con un método main() que permita instanciar un objeto de la clase Rectangulo para calcular su perímetro y área.

## Actividad 05: Menú mediante clases ya creadas.

Crea un proyecto que te permita manejar las clases [Rectangulo](#) y [Circulo](#), cada una con un archivo separado en un mismo paquete. Tendrás que añadir otro que sea el principal.

Circulo  
Rectángulo  
Salir

Al seleccionar la opción de Circulo, introducirá su radio por teclado y mostrará por pantalla su área y perímetro.

Lo mismo al introducir los lados de un rectángulo.