

Una obra de arte es la denominación que se da al producto de una creación en el campo del arte. Existen distintos tipos de obras de arte, las más conocidas son las pinturas y las esculturas.

Toda obra de arte tiene un titulo, un autor y año de creación.

Las pinturas se distinguen como características importantes las dimensiones de ancho y alto y el estilo de la pintura (acrílico, al agua, laca, fresco, etc.).

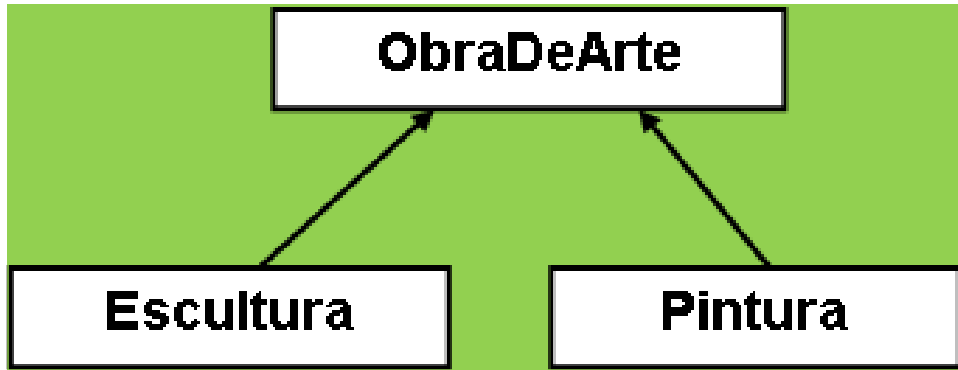
En las esculturas se sabe el material (madera, cera, yeso, hierro, cerámica, etc.), altura y peso.

Escultura: El David
Autor: Miguel Angel,
Se finalizó en 1504.
Altura: 5,16 metros.
Peso: 5.5 toneladas
Material: mármol



Pintura: La ultima cena,
Autor: Leonardo Da Vinci
Año: 1497 (se estima)
Estilo: Fresco.
Medidas: 460 x 880 cm.





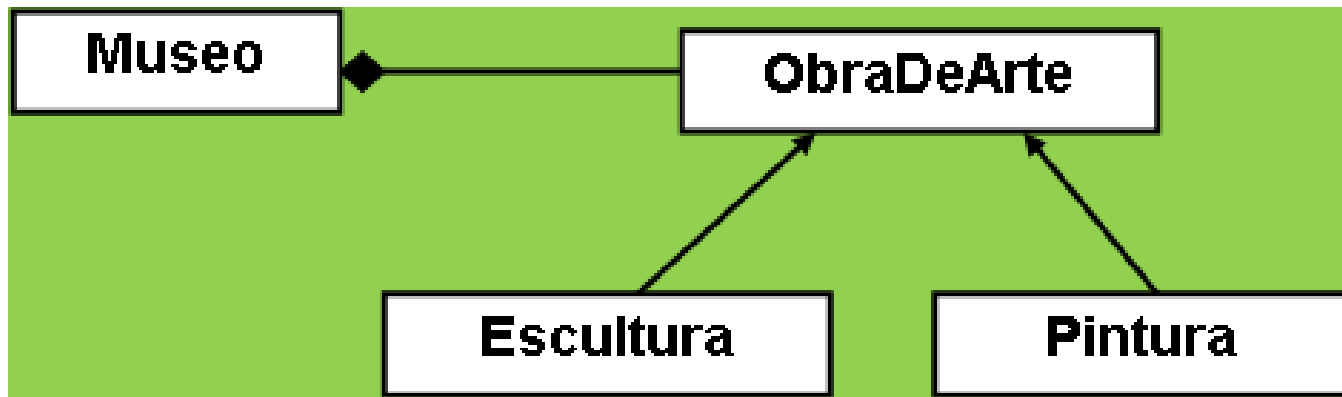
```
class ObraDeArte {  
    private String titulo, autor;  
    private int anio;  
    ObraDeArte(String titulo, String autor, int anio)  
    {   this.titulo = titulo;  
        this.autor = autor;  
        this.anio = anio;   }  
    void setTitulo(String titulo) { this.titulo=titulo;}  
    void setAutor(String autor) {this.titulo=titulo; }  
    void setAnio(int anio) { this.titulo=titulo;}  
    String getTitulo() { return this.titulo;}  
    String getAutor() {return this.autor;}  
    int getAnio() {return this.anio;}  
    void caracteristicas(){  
        System.out.println(this.getTitulo());  
        System.out.println(this.getAutor());  
        System.out.println(this.getAnio());  
    } }  
}
```

```
class Pintura extends ObraDeArte {
    private String estilo;
    private int ancho, alto;
    Pintura (...int ancho...)
    {   super( );
        this.ancho = ancho;

        ... .
    }
    void setAncho(int ancho) { this.ancho=ancho;}
    int getAncho() {return this.ancho();}

    void caracteristicas(){
        super();
        System.out.println(this.getAncho());
        ... .
    }
}
```

- Los museos son instituciones dedicadas al estudio, conservación y exposición de obras de arte.
- Es decir un museo es una colección de obras de arte.
- Los museos reciben visitantes, para lo cual es muy adecuado proporcionar información (listados) de las obras de arte que dispone y sus características.



```
import java.util.Vector;

class Museo {
    String nombre, domicilio, localidad
    Vector obras;
    Museo(....)
        { // inicialización atributos
            obras = new Vector(); }

    // métodos set/get de atributos

    void agregarObra(ObraDeArte o)
        { obras.addElement(o); }
```

Codificar el método mostrarObrasDeArte, que muestra todas las características de cada obra de arte, y si es pintura o escritura.

```
class Museo {
```

```
....
```

```
void mostrarObrasDeArte()    {  
    ObraDeArte o;  
    for (int i=0; i<obras.size(); i++)  
        o = (ObraDeArte) obras.elementAt(i);  
        o.getCaracteristicas();  
    }  
}
```

```
}
```

```
}
```

¿qué instanceof se debe hacer?
¿ qué cast se debe hacer?
¿a qué clase corresponde características?



¿ es polimórfico?

Codificar el método `MostrarPinturasPorAutor` que muestra todas las características de las pinturas del autor pasado por parámetro que están en el museo.

```
void mostrarPinturasPorAutor(String autor)
{
    Pintura p;
    for (int i=0; i<obras.size(); i++)
        if ( obras.elementAt(i) instanceof Pintura)
        {
            p = (Pintura) obras.elementAt(i);
            if (p.getAutor().equals(autor))
                (p.getCaracteristicas());
        }
}
```

```
void mostrarPinturasPorAutor(String autor)
{
    ObraDeArte o;
    for (int i=0; i<obras.size(); i++)
        if (obras.elementAt(i) instanceof
            Pintura)
        {
            o=(ObraDeArte)obras.elementAt(i);
            if (o.getAutor().equals(autor))
                (o.getCaracteristicas());
        }
}
```

RECOMENDACIONES

- En relaciones de herencia, verificar la correcta aplicación de **“es un”**
- En clases POJOS, **no leer datos** (console..etc), los datos vienen por parámetros
- En clases POJOS, **no desplegar mensajes** (SOP), los datos se devuelven por métodos “get”.
- En gestores **no leer datos** (console..etc), los datos vienen por parámetros
- En gestores, los métodos que calculan valores, **deben devolver valores**. NO usar void + SOP.
- Solo desplegar mensajes en métodos **listadores**, pero use opciones alternativas (vectores o cadena)
- Usar identificadores adecuados para clases/atributos/métodos
- Para diferenciar distintos tipos de objetos en un jerarquía de herencia esta **prohibido** usar atributos “tipo”. **NO HAY POLIMOFISMO**