

Programación OO con Java

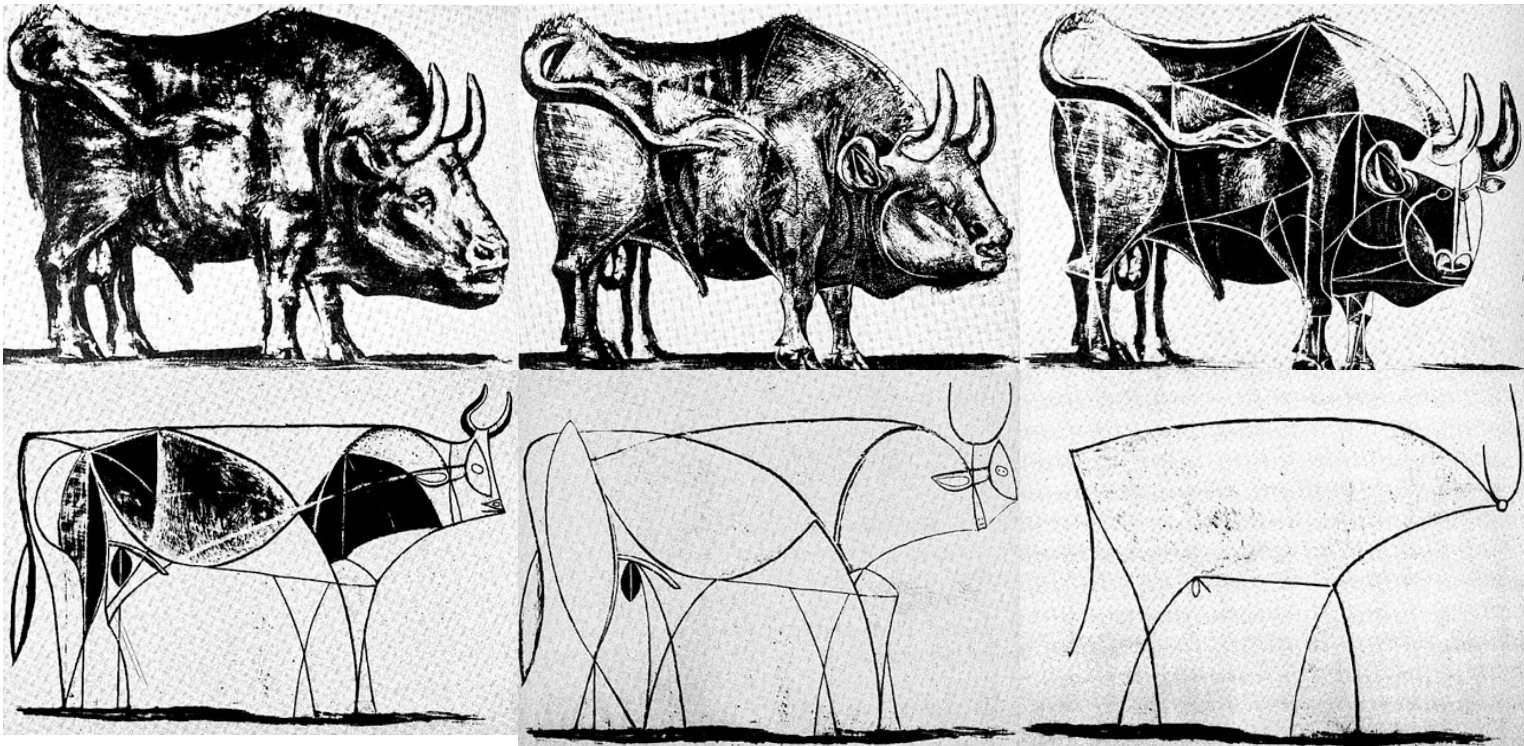
- Que es la Orientación a Objetos?
 - Consiste en el concepto de encapsulamiento.
 - Yo sé que es lo que hace una clase, pero no sé de que manera lo hace.
 - En términos más sencillos, yo sé que el método `ordenarArreglos(int [] arreglo)`, ordena un arreglo de enteros, pero no sé que técnica usa para esto (`quickSort`, `bubbleSort`, etc..).

Programación OO con Java

- Qué son los Objetos?
 - Un objeto contiene en sí mismo la información y los métodos o funciones necesarios para manipular esa información, restringiéndose solo a la información mas relevante de la entidad real o virtual que se desea modelar.

Programación OO con Java

- Abstracción



Programación OO con Java

- Qué son las Clases?
 - Las clases son las matrices de las que luego se pueden crear múltiples objetos del mismo tipo.
 - La clase define las variables y los métodos comunes a los objetos de ese tipo, pero luego, cada objeto tendrá sus propios valores y compartirán las mismas funciones.

Programación OO con Java

- Qué son los mensaje?
 - Los mensajes son simples llamadas a las funciones o métodos del objeto con el se quiere comunicar para decirle que haga cualquier cosa.

Programación OO con Java

- Cómo crear un Objeto?
 - Primero hay que declarar el Objeto.
 - Date fecha;
 - String nombre;
 - Integer i;
 - Después hay que inicializarlo.
 - fecha = new Date(2004,08,29);
 - nombre = new String("María");
 - i = new Integer(1024);
 - En general se tiene:
 - Clase variable = new Clase(Parametros constructor);

Programación OO con Java

- Constructores
 - Un objeto se crea usando constructores.

```
public class Rectangulo {  
    float x1,y1,x2,y2;  
    public Rectangulo(float ex1, float ey1,  
        float ex2, float ey2) {  
        x1 = ex1;  
        x2 = ex2;  
        y1 = ey1;  
        y2 = ey2;  
    }  
}
```

Programación OO con Java

- Ejercicio
 - Cree una clase Caja que posea
 - Alto
 - Ancho
 - Y un método para calcular su área
 - Ver: `Caja.java`, `Cajita.java` y `usa_caja.java`

Programación OO con Java

- Que es la herencia?
 - Permite crear una clase a través de una clase existente, y esta clase tendrá todas las variables y los métodos de su "superclase", y además se le podrán añadir otras variables y métodos propios.

Programación OO con Java

- Existen superclases y subclases.
- Las subclases heredan los métodos y variables de las superclases.
- Las superclases se transforman en padres y abuelos de las subclases.
- En Java la superclase más alta en el árbol jerárquico es la clase Object.
- Todas las clases heredan los métodos de la clase Object.

Programación OO con Java

- Para referirse a una variable de la clase superior se utiliza `super.variable`.
- Para referirse a un método de las clases superiores se utiliza la sentencia `super.metodo()`.
- Para referirse a una variable propia de la clase se utiliza la sentencia `this.(variable|metodo())`.

Programación OO con Java

- Ejemplo:
 - Nuestra superclase es Integer
 - Integer implementa el método suma().
 - Nuestra subclase será IntegerMensaje que extiende Integer.
 - IntegerMensaje debe entregar el mensaje “Estoy sumando” antes de efectuar la suma y enseguida sumar.

Programación OO con Java

- Entonces el método de suma de IntegerMensaje es distinto que el método suma de Integer
- ```
public IntegerMensaje suma() {
 System.out.println("Estoy sumando");
 super.suma();
}
```
- Vemos que aprovechamos que la suma se efectúa en la superclase y la reutilizamos para hacer el cálculo.

# Programación OO con Java

- Cómo es el constructor de un hijo?

```
- public class RecV extends Rectangulo {
- public String s;
- public RecV(float ex1, float ey1,
float ex2, float ey2, String cadena) {
- super(ex1,ey1,ex2,ey2);
- s = cadena;
- }
- }
```

# Programación OO con Java

- Se deben seguir reglas:
  - Los constructores NO se heredan, a diferencia de los métodos.
  - Toda clase debe tener un constructor. Si no esta declarado explícitamente, por defecto tiene el constructor con 0-parámetros.
  - Si una clase tiene definidos constructores, esos son los **únicos que** tiene, específicamente deja de existir el constructor con 0-argumentos.

# Programación OO con Java

- Se deben seguir reglas:
  - El cuerpo de un constructor **debe** llamar una vez al constructor de la clase padre, que debe preceder a cualquier otra acción en el cuerpo del constructor. Si el cuerpo de un constructor no comienza con una sentencia llamando explícitamente al constructor de la clase padre, entonces la sentencia
    - `super() ;`
    - se asume (i.e. el constructor con 0-argumentos).



# Programación OO con Java

- Ejercicio
  - Cree una clase hijo, Cajita que incluya un nuevo parámetro: profundidad y un nuevo método: volumen
  - Ver: `Cajita.java` y `usa_caja.java`