

## Tecnología Multimedial 1

### Trabajo Práctico nº4

**Profesor: David Bedoian**  
**Alumno: Elian Rodriguez**  
**Comisión 3**

#### A) Fundamento teórico

- 1) Explica en un párrafo a qué se refiere la Programación Orientada a Objetos.
  - 2) Describí la sintaxis de declaración y contenido de una clase (class) en Processing
  - 3) Desarrolla la definición y un ejemplo de 6 términos que consideres relevantes o relativos al tema. Por ejemplo class, instancia, constructor, método, etc...
  - 4) Explicar el concepto de conocimiento entre objetos.
- 

#### A)

- 1) La Programación Orientada a Objetos habla de un “objeto” al que se le puede asignar propiedades, dichas propiedades son lo que ya conocemos como variables y funciones. A partir de este punto se puede decir que la **clase** (class) es un modelo (plantilla) que contendrá un grupo de propiedades que serán asignables a un objeto; dentro de estas propiedades se encuentran los **campos**, lo que se conoce como variables, serán aquellas que indiquen cómo se vé, cuánto pesa, qué tan grande es, entre otras opciones y, por otro lado, existen los **métodos**, que son cómo el objeto se comportará. Por medio de estos campos y métodos será posible representar características y acciones de un determinado objeto u objetos.
- 2) Contenido desarrollado dentro de el tp4.pde
- 3) Las **class** son un grupo de campos y métodos. Se podría decir que son un conjunto de propiedades, un modelo, que se le asigna a un objeto. Para este caso se podría llamar a la class “**Pelota**”. El **objeto** de esta class es quien va a adquirir estos campos y métodos.

Los **campos** son variables, atributos que se le asignan a los objetos. A nuestra pelota se le asignan, por ejemplo: color, peso, tamaño, etc.

Los **métodos** son funciones, acciones que este objeto puede realizar. Es la forma en la que actuará y se comportará un determinado objeto. Entonces, la pelota podrá, por ejemplo: rodar() rebotar() etc.

**Instancia** se le denomina a un objeto nuevo a partir del modelo ya establecido de una clase. EJ: PelotaTenis, PelotaVolley, PelotaBasket. En

este caso todas son pelotas y todas se comportan de acuerdo a la class "Pelota"

Dentro de una class, el **constructor**, es aquella parte del código que será activada cuando el objeto es creado. Cada clase tiene al menos un constructor. EJ: Pelota = new Pelota();

**Herencia** es lo que se describe como la capacidad de que una clase pueda compartir sus **campos** con otra clase. Por ejemplo: Es posible tener una clase llamada "Globo" que, en este caso, comparte el atributo del color o tamaño de la clase "Pelota".

**Polimorfismo**, dentro de lo que a este concepto se refiere, se puede decir que diferentes objetos pueden compartir **métodos**. Se puede hablar de un conjunto de pelotas, la class entonces se llamará "pelota" y lo que se le puede asignar a estas diferentes pelota sería, por ejemplo, la opción de rebotar().

- 4) Se le suele llamar **conocimiento entre objetos** a aquellas conexiones que se realizan entre las clases y los objetos para poder desarrollarse como una aplicación funcional. Por ejemplo, la herencia.