

TP4 Tecnología Multimedia

1) La programación orientada a objetos es una forma especial de programar, más cercana a cómo expresaremos las cosas en la vida real que otros tipos de programación. Al programar orientado a objetos tenemos que aprender a pensar cómo resolver los problemas de una manera distinta a como se realizaba anteriormente, en la programación estructurada. Ahora tendremos que escribir nuestros programas en términos de clases, objetos, propiedades, métodos y otras cosas que permita soltarnos un poco con los conceptos de este tipo de programación.

La programación orientada a objetos se define como un paradigma de la programación, una manera de programar específica, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.

2) una **clase** es una construcción que se utiliza como un modelo para crear objetos de ese tipo. El modelo describe el estado y contiene el comportamiento que todos los objetos creados a partir de esa clase tendrán. Un objeto creado a partir de una determinada clase se denomina una instancia de esa clase.

La clase consta de: clase, constructor y método. El nombre de la clase y el del constructor deben ser el mismo. El método es una función que se aplica a los objetos y que puede tomar parámetros externos a la clase.

ejemplo:

```
class Burbuja {
```

```
    //Declaramos la clase 'Burbuja' con tres propiedades
```

```
    float x;
```

```
    float y;
```

```
    color c;
```

```
    Burbuja(float burbuja_x, float burbuja_y, color burbuja_c) {
```

```
        //Creamos un constructor con el mismo nombre de la clase y le pasamos sus parámetros
```

```
        x = burbuja_x;
```

```
        y = burbuja_y;
```

```
        c = burbuja_c;
```

```
    }
```

```
    void subir(int n) {
```

```
        //Creamos un método 'caer' para mover los objetos 'gota'.
```

```
        fill(c, 50);
```

```
        stroke(50);
```

```
        y = y-n;
```

```

    if (y < 0) {
    y = height+20;
    }
    ellipse(x, y, 20, 20);
}
}
//Y después la otra parte que le sigue.

```

3)

class: es una construcción que se utiliza como un modelo (o plantilla) para crear objetos de ese tipo. El modelo describe el estado y contiene el comportamiento que todos los objetos creados a partir de esa clase tendrán.

Objetos: es simplemente una instancia de la clase

Instancia: es un objeto creado a partir de una determinada clase. Una de las utilidades de usar clases, objetos y funciones es la de mantener nuestro código organizado y reutilizar partes del mismo.

Ejemplo de class e instancia:

```

class Burbuja {
  //Declaramos la clase 'Burbuja' con tres propiedades
  float x;
  float y;
  color c;

```

```

  Burbuja(float burbuja_x, float burbuja_y, color burbuja_c) {
    //Creamos un constructor con el mismo nombre de la clase y le pasamos sus
    parámetros
    x = burbuja_x;
    y = burbuja_y;
    c = burbuja_c;
  }

```

```

void subir(int n) {
  //Creamos un método 'caer' para mover los objetos 'gota'.
  fill(c, 50);
  stroke(50);
  y = y-n;
  if (y < 0) {
    y = height+20;
  }
}

```

```
    ellipse(x, y, 20, 20);  
  }  
}
```

variables de una clase: Una variable de clase es creada cuando su clase o interfaz es preparada y es inicializada a un valor por defecto

Constructor: un constructor es una subrutina cuya misión es inicializar un objeto de una clase. En el constructor se asignan los valores iniciales del nuevo objeto.

Ejemplo: `public SSuma() {}`

Método: El método es una función que se aplica a los objetos y que puede tomar parámetros externos a la clase.

Ejemplo:

```
class Nombre {  
    // Variables miembro (habitualmente privadas)  
    miembro_1; //lista de miembros  
    miembro_2;  
    miembro_3;  
  
    // Funciones o métodos (habitualmente públicas)  
    funcion_miembro_1( ); // funciones miembro conocidas  
    funcion_miembro_2 ( ); // funciones como métodos  
  
    // Propiedades (habitualmente públicas)  
    propiedad_1;  
    propiedad_2;  
    propiedad_3;  
    propiedad_4;  
}
```

4) no me quedó del todo claro profe...