

Miniguía de Processing orientado a objetos



Por: *Pedro Ruiz Fernández*
Versión 24/07/2017

Licencia



Programación orientada a objetos.

Processing es un lenguaje orientado a objetos, eso significa que debemos entender los conceptos de clase, objetos, constructores, métodos, etc.

Conceptos

Vamos a definir varios conceptos básicos esenciales para empezar a trabajar con objetos y no fracasar en el intento:

- **Objeto:** es un elemento que vamos a programar que tiene entidad propia, con características o parámetros propios, y con métodos que van a ser acciones que se realizan sobre él. Por ejemplo podemos programar una pelota y para ello tenemos que definirla con varios parámetros como son posición en X, posición en Y, diámetro, color, etc. Además podemos definir acciones que podemos realizar sobre la misma, como: dibujarla, moverla, acelerarla, etc.
- **Clase:** en programación a veces nos toca trabajar con entidades y objetos que se comportan de manera idéntica, ese modelo de tipo de objetos con características idénticas la llamamos clase. Por ejemplo puedo establecer una clase llamada pelota con sus parámetros (características) y métodos (acciones), y más tarde crear objetos llamados pelota1, pelota2, etc, con los parámetros y métodos de la clase pelota a la que pertenecen.

Programa ejemplo nº 1

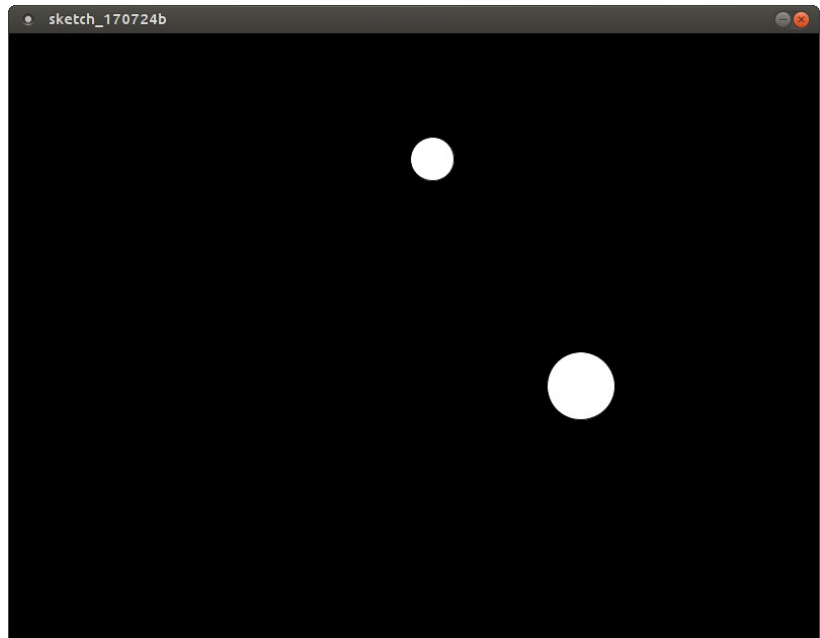
El movimiento lo demostramos andando para ello vamos a realizar un programa de ejemplo para explicar los conceptos anteriores.

```
//definimos la clase del objeto en nuestro caso
pelota
class pelota {
  /*dentro de la clase definimos los cinco parámetros
  que tendrán estos objetos, en nuestro caso son float
  porque van a ser asignadas más tarde con valores
  aleatorios (random)*/
  float posX,posY,diametro,velX,velY;
  /*a continuación ponemos el constructor en nuestro
  caso "pelota", un constructor sirve para asignar a un
  objeto real los parámetros que va a tener*/
  pelota (float x, float y) {
    posX=x;
    posY=y;
    diametro=random(100);
    velX=random (10);
    velY=random (10);
  }
  /* Ahora pasamos a definir los procedimientos o
  métodos que posee la clase de objetos pelota, que
  serán las acciones que se pueden hacer con ese tipo
  de objetos. En nuestro caso vamos a definir dos, uno
  que dibuja la pelota y otro que la mueve */
```

```
/* este procedimiento dibuja la pelota usando tres parámetros pasados de los cinco, estos parámetros se pasarán al
objeto en su construcción, además del resto*/
void dibujapelota() {
  ellipse (posX,posY,diametro,diametro);
}
```

```
/* Con este procedimiento vamos a cambiar las posiciones del centro de la pelota en función de dos velocidades X e Y,
y también haremos que la pelota rebote al tocar un borde */
void muevepelota(){
  posX=posX+velX;
  posY=posY+velY;
  if (posX<(diametro/2) || posX >(width-(diametro/2))) {velX=-velX;}
  if (posY<(diametro/2) || posY >(height-(diametro/2))) {velY=-velY;}
}
}
```

```
/*declaro las dos variables tipo pelota*/
pelota pelota1,pelota2;
```



```

/*en el setup creo las dos pelotas con el constructor definido anteriormente*/
void setup() {
  pelota1= new pelota (random(width),random(height));
  pelota2= new pelota (random(width),random(height));
  background(0);
  size (800,600);
}

```

```

/*en el draw nuevo y pinto las pelotas*/
void draw() {
  background(0); //para que borre las pelotas anteriores
  pelota1.muevepelota();
  pelota1.dibujapelota();
  pelota2.muevepelota();
  pelota2.dibujapelota();
}

```

Programa ejemplo nº 2

Ahora vamos a realizar una variante del programa anterior añadiendo un array (lista de elementos) de pelotas, en nuestro caso 10. En el código todo lo relativo a la clase es exactamente igual, pero tendremos cambios en las definiciones de variables del programa, en *setup* y *draw*, vamos al código:

```

//definimos la clase del objeto en nuestro caso
pelota

```

```

class pelota {
  //dentro de la clase definimos los cinco parámetros
  que tendrán estos objetos, en nuestro caso son float
  porque van a ser asignadas más tarde con valores
  aleatorios (random)

```

```

float posX,posY,diametro,velX,velY;
  /*a continuación ponemos el constructor en nuestro
  caso "pelota", un constructor sirve para asignar a un
  objeto real los parámetros que va a tener*/

```

```

  pelota (float x, float y) {
    posX=x;
    posY=y;
    diametro=random(100);
    velX=random (10);
    velY=random (10);
  }

```

```

  /* Ahora pasamos a definir los procedimientos o
  métodos que posee la clase de objetos pelota, que
  serán las acciones que se pueden hacer con ese tipo
  de objetos. En nuestro caso vamos a definir dos, uno que dibuja la pelota y otro que la mueve */

```

```

  /* este procedimiento dibuja la pelota usando tres parámetros pasados de los cinco, estos parámetros se pasarán al
  objeto en su construcción, además del resto*/

```

```

void dibujapelota() {
  ellipse (posX,posY,diametro,diametro);
}

```

```

  /* Con este procedimiento vamos a cambiar las posiciones del centro de la pelota en función de dos velocidades X e Y,
  y también haremos que la pelota rebote al tocar un borde */

```

```

void muevepelota(){
  posX=posX+velX;
  posY=posY+velY;
  if (posX<(diametro/2) || posX >(width-(diametro/2))) {velX=-velX;}
  if (posY<(diametro/2) || posY >(height-(diametro/2))) {velY=-velY;}
}
}

```

```

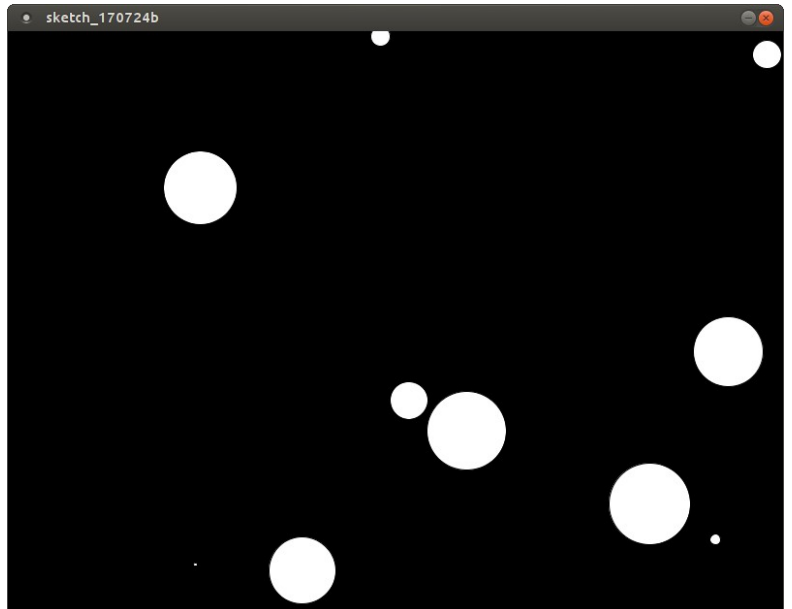
/*aquí defines un conjunto de elementos pelota (array) llamado pelotas, con 10 elementos*/
pelota [] pelotas=new pelota[10];

```

```

void setup() {
  /* creo con un for las 10 pelotas*/
  for (int n=0;n<10;n++) {

```



```
pelotas[n]=new pelota (random(width),random(height));
}
background(0);
size (800,600);
}

/*en el draw muevo y pinto las pelotas*/
void draw() {
background(0); //para que borre las pelotas anteriores
for (int n=0;n<10;n++) {
pelotas[n].muevepelota();
pelotas[n].dibujapelota();
}
}
```