

P5Aplicacion.java

```
1 package usuario;
2
3 import imagenes.ColorRGB;
4
5 /**
6  * Clase principal para la Práctica 5.
7  * El alumno debe realizar la siguiente secuencia de operaciones:
8  *   - Leer una imagen de disco y mostrarla por pantalla.
9  *   - Realizar una operación de espejo vertical y mostrar el resultado por pantalla.
10 *   - Realizar una operación de espejo horizontal y mostrar el resultado por pantalla.
11 *   - Deshacer espejado total y mostrar el resultado por pantalla.
12 *   - Crear un marco foto a la imagen anterior y mostrar el resultado por pantalla.
13 *   - Guardar imagen procesada en disco.
14 */
15 @version 1.0
16 */
17 public class P5Aplicacion {
18
19     public static void main(String[] args) {
20         ImagenRGB imagen;
21         Efecto efecto;
22         String fichero = "MicroHobby.jpg";
23         ColorRGB amarillo = new ColorRGB(190,50,230);
24
25         //Generación de la imagen original.
26         imagen = new ImagenRGB(fichero);
27         imagen.presentarImagen();
28
29         //up casting//
30         efecto = new EfectoMarcoFoto("Foto", imagen, amarillo);
31
32         //Down casting//
33         EfectoEspejo efecto1 = (EfectoEspejo) efecto;
34
35         //Procesamiento con espejos - Vertical.
36         efecto1.setTipo(true);
37         imagen = OperarEfecto (efecto1,true);
38
39         //Procesamiento con espejos - Horizontal.
40         efecto1.setTipo(false);
41         imagen = OperarEfecto (efecto1 ,true);
42
43         //Procesamiento con espejos - Deshacer espejado total.
44         imagen = OperarEfecto (efecto1, false);
45
46         //Establecimiento de un marco estilo foto.
47         efecto = new EfectoMarcoFoto ("Foto", imagen, amarillo);
48         imagen = OperarEfecto (efecto,true);
49
50         //Guarda resultado en fichero.
51         imagen.escribirImagen("ImagenConEfectos.bmp");
52     }
53 }
```

P5Aplicacion.java

```
67     }
68
69
70     /**
71      * <p>Este método se encarga de aplicar un procesado sobre la imagen. Devuelve la imagen
72      * resultante y la muestra por pantalla.
73      * El método permite aplicar el efecto indicado o deshacer las operaciones previamente
74      * realizadas con el mismo.</p>
75      * Precondiciones:<ol>
76      * <li>{@code "efecto" debe ser un objeto de clase Efecto (o derivada) válidamente
77      creado.</li>
78      * </ol>
79      * @param efecto Efecto a aplicar o deshacer.
80      * @param deshacer Con true deshace el tratamiento previamente implementado con el
81      * efecto; Con false aplica el efecto.
82      */
83     private static ImagenRGB OperarEfecto(Efecto efecto, boolean deshacer) {
84         ImagenRGB imagenResultado;
85
86         if(!deshacer) {
87             efecto.deshacer();
88         }
89         else {
90             efecto.aplicar();
91         }
92         imagenResultado = efecto.getImagen();
93         imagenResultado.presentarImagen();
94
95         return imagenResultado;
96     }
97 }
98
```