

P6Aplicacion.java

```
1 package usuario;
2
3 import imagenes.ColorRGB;
4
5
6
7
8
9
10 /**
11  * Clase principal para la Práctica 6.
12  * El alumno debe realizar la siguiente secuencia de operaciones:
13  *   - Leer una imagen de disco y mostrarla por pantalla.
14  *   - Incluir un marco de anchura 10 píxeles.
15  *   - En una segunda ejecución cambie este valor por un número negativo y luego
16  *     por un ancho muy grande (por ejemplo 1000 píxeles) y compruebe qué sucede
17  *   - Transformar la imagen a escala de grises.
18  *   - Aumentar el brillo de la imagen en un 30%.
19  *   - En una segunda ejecución cambie este valor por 130% y compruebe qué sucede.
20  *   - Guardar imagen en disco.
21  *   - Deshacer el último efecto realizado.
22  *   - Intentar volver a deshacer el efecto.
23  * Para ello complete las líneas que se indican con la etiqueta "Rellene". En cada
24  * espacio designado con esta etiqueta deberá completar o introducir una única línea
25  * de código.
26  * También debe controlar los errores generados tal y como se recoge en el enunciado
27  * de la práctica.
28  *
29  *
30  */
31 public class P6Aplicacion {
32
33     public static void main(String[] args) throws EfectoException{
34         ImagenRGB imagen,imagen1;
35         Efecto efecto;
36         String fichero = "MicroHobby.jpg";
37         TrasformarColor tipo;
38         ColorRGB rojo = new ColorRGB(255,0,0);
39
40         try {
41             //Generación de la imagen original.
42             imagen = new ImagenRGB(fichero);
43             imagen1 = new ImagenRGB(fichero);
44             imagen.presentarImagen();
45
46
47             //Procesamiento para incluir en la imagen un marco de 10 píxeles de color rojo.
48             efecto = new EfectoMarco ("Foto", imagen,10,rojo);
49             EfectoMarco efecto1 = (EfectoMarco) efecto;
50             imagen = operarEfecto (efecto1,true);
51
52
53             //Procesamiento para incluir en la imagen un marco de ancho negativo (y
54             //posteriormente de ancho 1000 píxeles)
55             efecto = new EfectoMarco ("Foto", imagen,-10,rojo);
56             efecto = new EfectoMarco ("Foto", imagen,1000,rojo);
57
58             //Procesamiento a color gris.
59             tipo = new TrasformarColorEnGris();
60             efecto = new EfectoColor("Foto en GRIS", imagen1, tipo);
61
62             imagen = operarEfecto ( efecto, true );
63
64             //Procesamiento aumentando el brillo 30%.
65             tipo = new TrasformarColorEnBrillo(30);
66             efecto = new EfectoColor("Brillo 30", imagen1, tipo );
```

P6Aplicacion.java

```

67     imagen = operarEfecto ( efecto,true );
68
69     //Procesamiento aumentando el brillo 130%.
70     tipo = new TrasformarColorEnBrillo(130);
71     efecto = new EfectoColor("Brillo 130", imagen, tipo );
72
73     //Guarda resultado en fichero.
74     imagen.escribirImagen("ImagenConEfectos.bmp");
75     //Deshacer.
76     efecto.deshacer();
77     operarEfecto(efecto, false);
78
79     //Deshacer (2).
80     efecto.deshacer();
81     operarEfecto(efecto, false);
82
83     }catch(IllegalArgumentException iaE) {
84         System.out.println("Se ha producido una excepción IllegalArgumentException: "+
85         iaE.getMessage());
86     }catch(EfectoException e) {
87         System.out.println("Se ha producido una excepcion EfectoException: "+
88         e.getMessage());
89     }
90 }
91
92 /**
93  * Este método se encarga de aplicar el efecto indicado, devolviendo la imagen
94  * resultante y mostrándola por pantalla.
95  * El método permite aplicar el efecto o deshacer las operaciones previamente
96  * realizadas con el mismo.
97  * @param efecto Efecto a aplicar o deshacer.
98  * @param deshacer Con true deshace el tratamiento implementado con el efecto; false
99  *                 aplica el efecto.
100 */
101
102 private static ImagenRGB operarEfecto(Efecto efecto, boolean deshacer) {
103
104     ImagenRGB imagenResultado;
105
106     if(deshacer) {
107         efecto.aplicar();
108         imagenResultado = efecto.getImagen();
109         imagenResultado.presentarImagen();
110     }else {
111         imagenResultado = efecto.getImagen();
112         imagenResultado.presentarImagen();
113     }
114
115     return imagenResultado;
116 }
117
118 }
119 }
120

```