

Ampliación de Bases de Datos e Ingeniería de Software

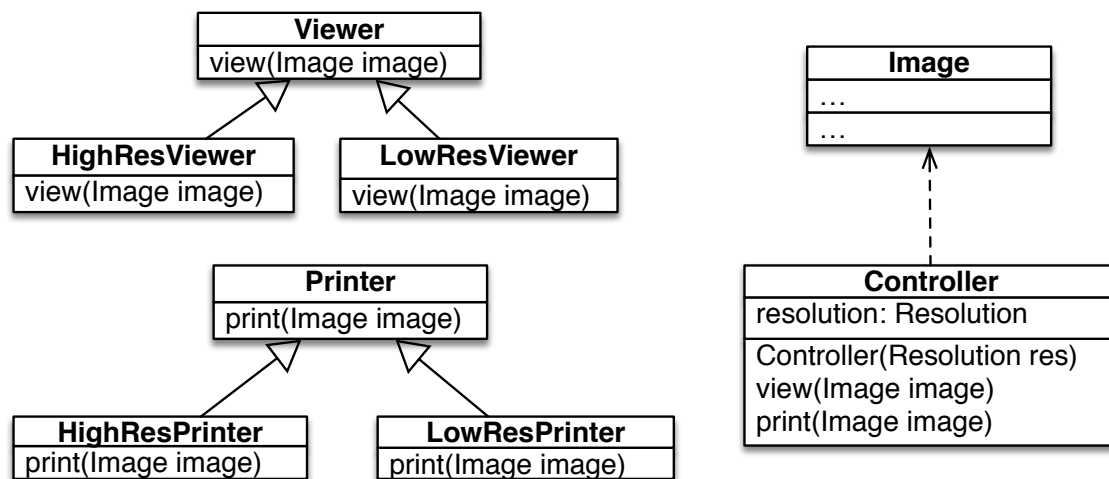
Escola Politècnica Superior
Universitat de Lleida

Recuperación del segundo parcial
23-24 de junio de 2020

Instrucciones generales

- La solución deberá entregarse en un fichero PDF cuyo nombre será **Apellidos, Nombre.pdf**
- La solución deberá ser mecanografiada, no escrita a mano.
- En caso de necesitar realizar algún diagrama, sí puede ser realizado a mano y, posteriormente, escaneado/fotografiado e incorporado al PDF.
- He configurado la actividad de manera que, durante el periodo de 24 horas, se pueda hacer un número ilimitado en entregas; os sugiero que, conforme vais realizando el examen, vayáis entregando parcialmente la solución.
- Si durante el examen detectáis alguna cosa que consideráis ambigua, explicitad la interpretación que hacéis y seguid con la solución.

En una aplicación se dispone de las siguientes clases que muestran e imprimen imágenes en diferentes resoluciones. En el diagrama no se muestran todas las dependencias ya que Controller depende de todas las otras clases.



El tipo **Resolution** està definido como:

```
1 public enum Resolution {
2     LOW, HIGH;
3 }
```

aunque es muy probable que aparezcan nuevas resoluciones (y nuestro diseño deberá ser capaz de adaptarse a ellas); eso sí, tal y como se verá en el código que mostramos a continuación, la resolución de **view** ha de ser la misma que la de **print**.

El método view, de la clase Controller, es el siguiente:

```
1 public void view(Image image) {
2     Viewer viewer;
3     switch (resolution) {
4         case Resolution.LOW : viewer = new LowResViewer();
5                               break;
6         case Resolution.HIGH : viewer = new HighResViewer();
7                               break;
8         default: throw new IllegalStateException(
9             "invalid resolution " + resolution);
10    }
11    viewer.view(image);
12 }
```

Y el método print es similar:

```
13 public void print(Image image) {
14     Printer printer;
15     switch (resolution) {
16         case Resolution.LOW : printer = new LowResPrinter();
17                               break;
18         case Resolution.HIGH : printer = new HighResPrinter();
19                               break;
20         default: throw new IllegalStateException(
21             "invalid resolution " + this.resolution);
22    }
23    printer.print(image);
24 }
```

Se pide:

- 1) ¿Qué problemas presenta el diseño de la clase Controller?
- 2) Diseñad e implementad **dos rediseños de la clase Controller** que solucionen los problemas mencionados en el apartado anterior. Indicad qué patrones aplicáis y por qué. Podéis añadir nuevas clases y, obviamente, modificar la clase Controller; las otras clases del diseño no pueden modificarse.