LLIURAMENT 2:REPRODUCTOR MULTIMÈDIA

Marc Cerrillo | Pau Bernabé

Universitat de Barcelona (Curs 2017-2018)

GRUP B00

PREGUNTES SOBRE EL LLIURAMENT 2

1. Expliqueu les classes implementades

En aquesta pràctica utilitzem 3 paquets diferents: vista, controlador i model. A continuació explicarem què fa cada classe dels diferents paquets.

VISTA

IniciadorAplicacioUB: Aquesta classe conté el main principal on s'executa el programa. Dins d'aquest main hem creat un objecte aplicacio de tipus AplicacioUB2 que crida al mètode gestioAplicacioUB.

AplicacioUB2: Dins aquesta classe hem implementat els menús que sortiran per pantalla i dins de cada opció hem col·locat diferents mètodes perquè recullin les dades introduïdes i vagin al controlador.

CONTROLADOR

Controlador: Aquesta classe fa de pont entre els paquets vista i model, més concretament entre la classe AplicacioUB2 (vista) i la classe Dades (model). Amb els valors obtinguts de la classe AplicacioUB2 farà les crides oportunes a la classe Dades

MODEL

Dades: Aquesta classe s'encarregarà de guardar totes les dades introduïdes per afegir-les a una biblioteca en forma de vídeo o àudio. També tindrà dos mètodes per guardar i recuperar les dades abans de sortir de l'aplicació.

BibliotecaFitxersMultimedia: Aquesta classe ens permet gestionar els fitxers de la biblioteca, com per exemple afegir o eliminar un fitxer multimèdia, mostrar la llista de tots els fitxers de la biblioteca, etc.

FitxerMultimedia: És una classe filla de la classe File que importem amb java.io.File. Al constructor de FitxerMultimedia declarem la informació del fitxer per cridar al constructor de File.

CarpetaFitxers: Aquesta classe representa una carpeta de fitxers a partir d'un ArrayList amb una capacitat màxima de 100 fitxers.

FitxerReproduible: És l'especialització de la classe FitxerMultimedia, ja que diferencia entre fitxers reproduïbles (vídeo i àudio) i els que no ho són.

Audio\Video: Aquestes classes seran les que a partir dels seus constructors crearan els seus respectius fitxers.

2. Expliqueu quants objectes s'han creat en l'execució d'aquest mètode *main* si estem a l'última línia:

```
public static void main(String[] args) {
// Creem un objecte de la vista
AplicacioUB2 aplicacio=new AplicacioUB2();
// Inicialitza l'execució de la vista
aplicacio.gestioAplicacioUB();
}
```

Es crea: aplicació ,controlador ,reproductor ,dades ,biblioteca ,carpeta , ArrayList

3. Expliqueu com heu implementat i on heu utilitzat el mètode *equals* heretat de la classe *Object*.

```
@Override
  public boolean equals(Object fitxerMultimedia){
     boolean esIgual=false;
     if (fitxerMultimedia instanceof FitxerMultimedia){
         FitxerMultimedia altre = (FitxerMultimedia)fitxerMultimedia;
         esIgual=this.getAbsolutePath().equals(altre.getAbsolutePath());
      }
   return esIgual;
  }
```

Com està indicat el mètode equals es sobreescriu. L'hem sobreescrit de manera que si un fitxer que es vulgui introduir a la nostra *BibliotecaFitxersMultimedia* té el mateix path, doncs els identificarem com a dos fitxers iguals. Hem escollit el Path total ja que conté tota la informació com pot ser el nom del fitxer i la seva extensió, d'aquesta manera creiem que no se'ns colarà cap fitxer igual.

L'hem utilitzat en la *BibliotecaFitxersMultimedia* a l'hora d'afegir un fitxer, si la condició és certa, el mètode llançarà una excepció que ens avisarà que aquell fitxer ja existeix a la nostra Biblioteca i ens permetrà continuar treballant amb el nostre programa.

4. Expliqueu com heu utilitzat la classe ***AplicacioException*** al vostre codi.

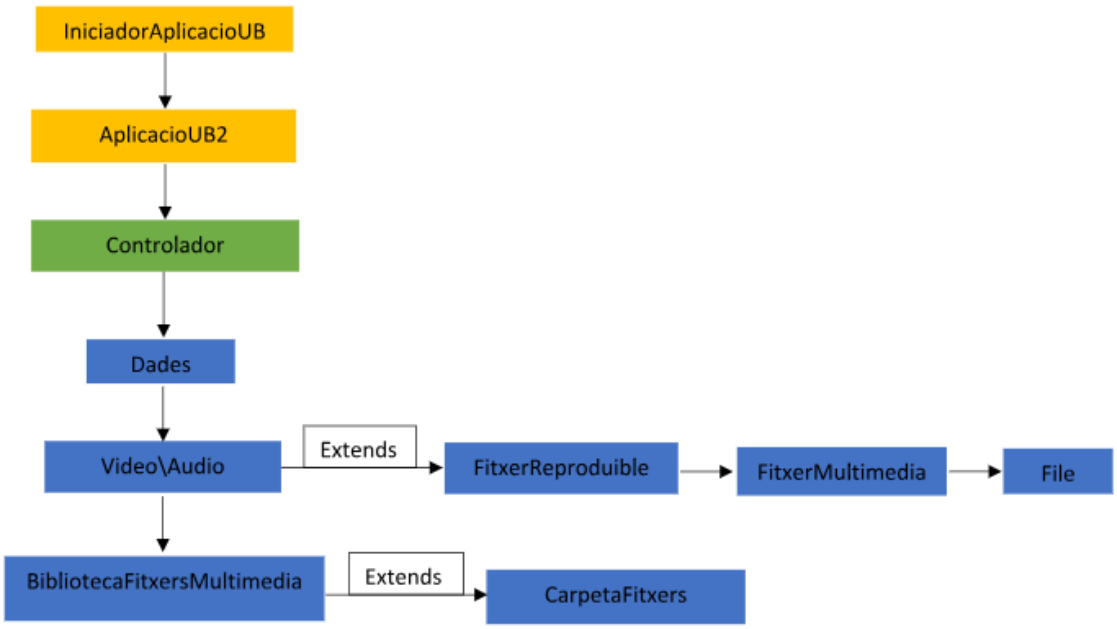
La classe AplicacióException la podíem trobar als fitxers Utils. Aquesta ens ha servit per a controlar el nostre codi en cas que hi haguessin excepcions. Un exemple podria ser quan tenim

la biblioteca buida, si l'usuari vol imprimir el seu contingut, l'excepció serà llençada a l'usuari, avisant-lo que no hi tenim cap fitxer.

5. Expliqueu si es podria fer servir la sobrecàrrega de mètodes a la classe **Controlador** i, en cas afirmatiu, detalleu com.

Podriem utilitzar la sobrecàrrega, però els arguments de tals mètodes haurien de tenir diferents arguments que els hi passaríem, ja que sinó, tindríem un error.

6. Feu un diagrama de les classes.

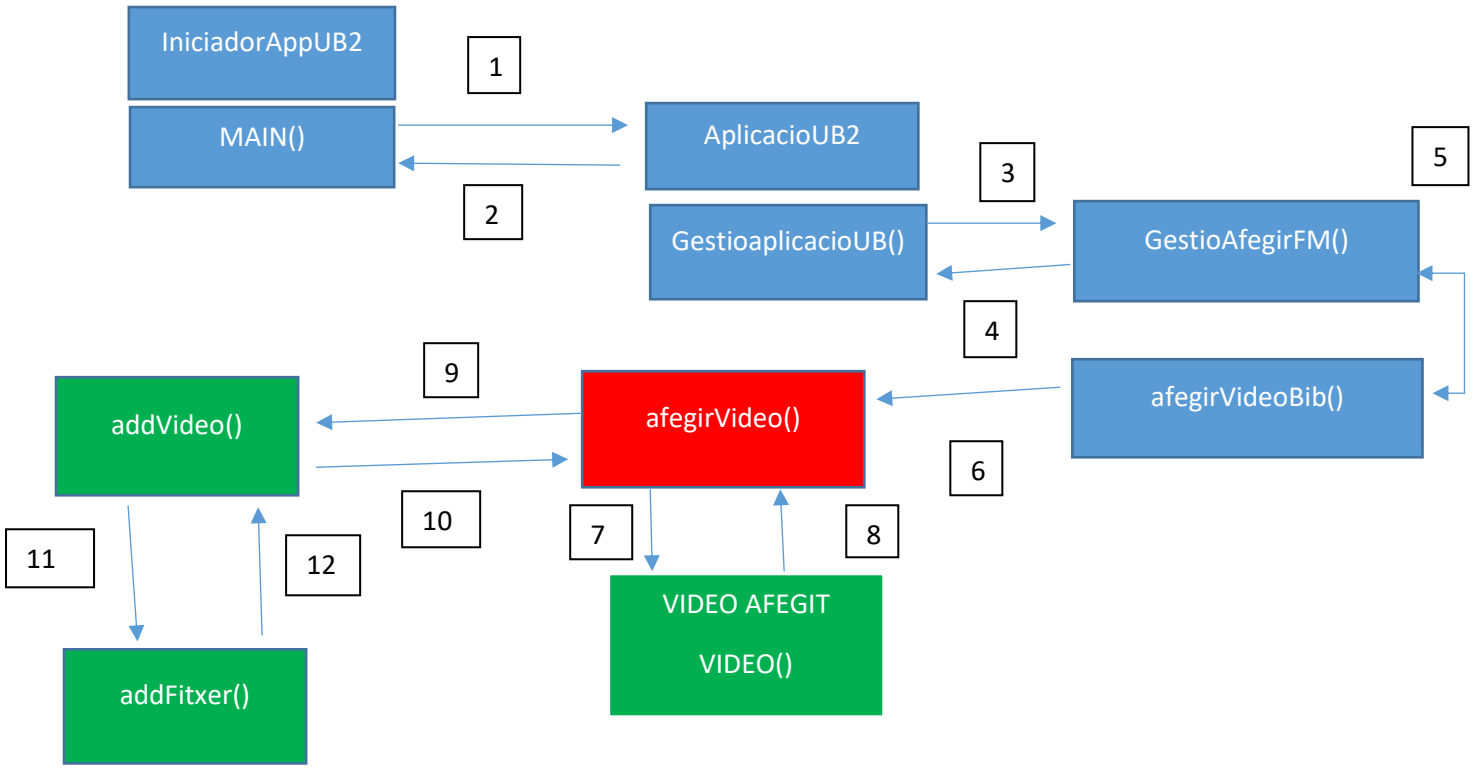


7. Feu un diagrama de flux per mostrar el recorregut que fa el vostre programa quan s'executa l'opció d'afegir un fitxer multimèdia a la biblioteca. Especificar els mètodes que es criden en cadascuna de les classes. Feu servir fletxes i números per indicar l'ordre de les crides.

Blau= VISTA

Vermell= Dades

Verd = Controlador



8. Expliqueu quins canvis hauríeu de fer al codi per tal d'aconseguir que al cridar al mètode reproduir de la següent manera:

FitxerReproduible fr = new FitxerReproduible(repro);
fr.reproduir();

Hauriem de fer que FitxerReproduible no fos abstracta, una classe abstracta no pot ser instanciada.

9. Expliqueu les proves realitzades per comprovar el correcte funcionament de la pràctica, resultats obtinguts i accions derivades.

Hem comprovat la majoria de casos que es poden donar en el nostre programa. Per exemple una de les condicions que ens donaven, dos fitxers no podien ser iguals i un fitxer havia d'existir per a poder ser afegir a la nostra Biblioteca. El que fèiem era intentar inserir fitxers inventats o algun que fos igual a algun prèviament afegit i en els dos casos esmentats, donava l'error i no afegia el fitxer, cosa que per alguns problemes que vam tenir, si ho feia.

Quan guardàvem les dades en un .dat també ens va sorgir algun error ja que, si no recordo malament no posàvem bé el directori.

Quan afegiem els fitxers, primerament, ens els afegia a la carpetaFitxers. Però el que vam fer va ser sobreescriure els mètodes per a que els fitxers s'afegissin a la biblioteca.

10. Observacions generals

En aquest lliurament no hem tingut massa dificultat a que tot funcionés a la seva totalitat, algun problema que hem tingut és al mostrar la biblioteca, ja que el LIST<String> ens venia una mica de nou, però al final hem sapigut com resoldre-ho. També al iniciar l'aplicació surten unes lletres en vermell, que no fan que el programa funcioni de manera errònia.

Aquests dos lliuraments no són pas difícils, sinó que hem de ser ordenats i tenir una idea clara de com estructurarem el codi.