Marc Cerrillo - NIUB: 20081552

Pau Bernabé - NIUB: 20081736

Programació II

22/03/2018

Memòria del Lliurament 1

Índex:

1. INTRODUCCIÓ

En aquesta pràctica volem organitzar fitxers multimèdia dins una carpeta de fitxers. Cada fitxer multimèdia quedarà representat pel seu fitxer, guardant el camí on es troba, el nom, l'extensió, la data última de modificació i la descripció del fitxer.

Tractarem com utilitzar l'herència, la crida a llibreries, *getters* i *setters…*

2. ANÀLISI

Haurem de crear un projecte amb diferents paquets i classes. Començarem amb IniciadorAplicacioUB des d'on s'executarà el programa. Seguidament implementarem la classe AplicacioUB1 on hi haurà el menú principal on escollir les diferents opcions. Al paquet model tindrem les classes FitxerMultimedia i CarpetaFitxers on guardarem la informació dels fitxers i implementarem els diferents mètodes.

3. DESENVOLUPAMENT

Al nostre projecte hem definit dos paquets: edu.ub.prog2.CerrilloMarcBernabePau.model i edu.ub.prog2.CerrilloMarcBernabePau.vista. Dins aquests paquets hem implementat diferents classes que explicarem a continuació:

IniciadorAplicacioUB: és simplement la classe que executa el programa i hem creat un objecte de classe AplicacioUB1.

AplicacioUB1: és la classe del menú principal. Utilitzem els atributs OpcionsMenuPrincipal i descMenuPrincipal que ens mostra les opcions per escollir i el Scanner sc llegeix la opció introduïda. A partir d'aquí entra a un bucle do...while on tenim un switch amb cada opció que fa la seva funció pertinent.

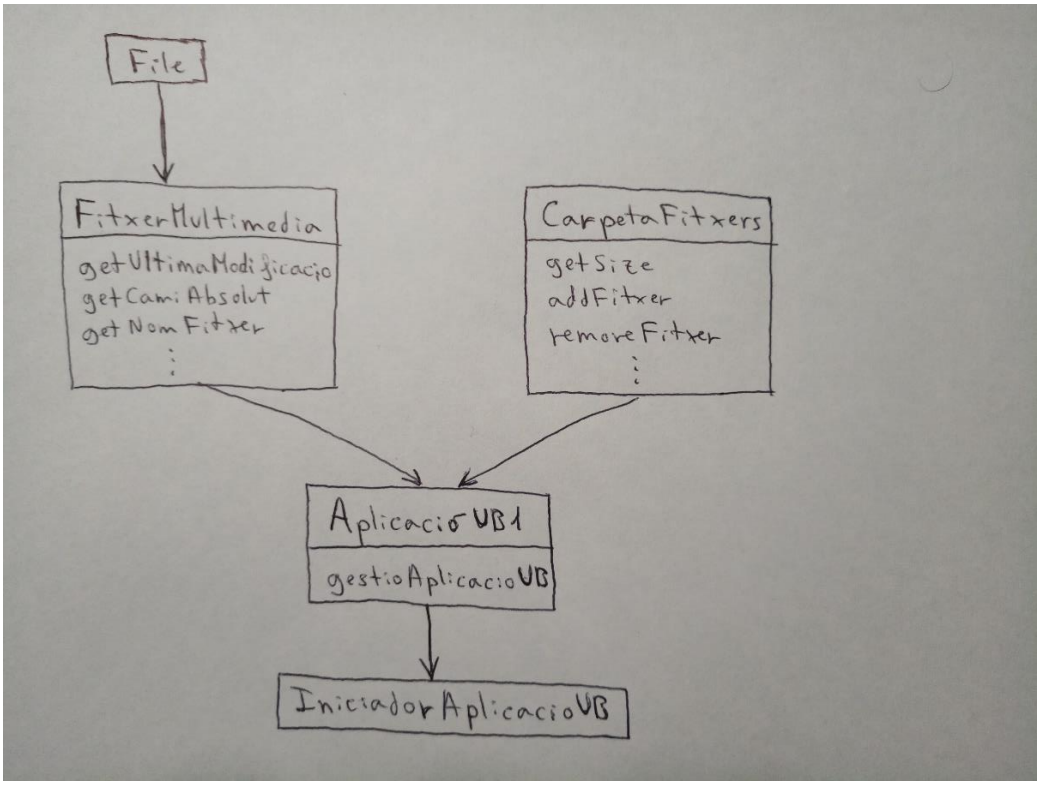
FitxerMultimedia: és una classe filla de la classe File que importem amb java.io.File. Al constructor de FitxerMultimedia declarem el camí al fitxer per cridar al constructor de File. Els diferents mètodes que tenim són getUltimaModificacio() que ens dona la data d'úlima modificació, getCamiAbsolut() que en dona el camí on està el fitxer, el getNomFitxer() que ens retorna el nom del fitxer, el getExtensio() que en dona la extensió, el equals(Object fitxerMultimedia) que ens diu si hi ha dos fitxers que es diuen igual i String toString() ens mostra el resum de tots els atributs del fitxer multimèdia en forma de cadena de text.

CarpetaFitxers: aquesta classe representa una carpeta de fitxers i hi implementem els mètodes getSize() que ens retorna el nombre d'elements de la carpeta, addFitxer() afegeix un nou fitxer a la carpeta, removeFitxer() elimina un fitxer, getAt() retorna el fitxer a la posició indicada de la carpeta, clear() elimina tots els elements de la carpeta i isFull() indica si la carpeta està plena. Per implementar aquest mètodes utilitzem la classe ArrayList importada de java.util.ArrayList.

4. PREGUNTES PLANTEJADES

1. Explicar breument les classes implementades.
   Les classes implementades ja han estat explicades a l'apartat de desenvolupament.

2. Dibuixar el diagrama de relacions entre les classes que heu utilitzat a la vostra pràctica. Incloure tant les classes implementades com les que pertanyen a la llibreria de Java i UtilsProg2. No cal incloure la llista d'atributs i mètodes.



3. Explicar com heu implementat i on heu utilitzat el mètode ***isFull***.
   El mètode isFull() de la classe CarpetaFitxers ens indica si la carpeta està plena (100 fitxers). A partir de l'array utilitzem el .size() per saber les posicions que ocupa i si són > o = a 100 llavors la carpeta està plena.

4. Segons la implementació de la classe ***CarpetaFitxers***, si tenim dos fitxers multimèdia corresponents al mateix fitxer, quan cridem al mètode per eliminar un d'aquests fitxers eliminarà l'altre també o no?
   Aquesta situació no passarà ja que tens un mètode implementat que et permet evitar dos fitxers amb el mateix nom.