



## **M03. Programació**

### **UF4. Programació orientada a objectes. Fonaments (RA1,RA2,RA3)**

#### **Aplicació en Java per gestionar la biblioteca “Terra Alta” (primera versió)**



## Descripció del projecte

S'ha de realitzar un prototipus d'aplicació d'escriptori basada en **JAVA** (versió 17 o posterior), que permeti millorar la gestió de les operacions més comuns que es realitzen en una petita **biblioteca** que dona servei als pobles de la comarca de la Terra Alta.

La gestió inclou Alta / Modificació / Eliminació / Llistat de tota la informació relacionada amb els clients, els llibres, els treballadors, etc. etc. de la biblioteca.

## Funcionalitats a implementar

- La biblioteca només pot prestar als seus clients llibres (llibres en paper i llibres en versió audiollibre) i una col·lecció de música clàssica en format de discos de vinil.
- La biblioteca considera que hi ha dos tipus de clients: client privat i client escola de música. El clients privats només poden demanar llibres, tant en paper com en audiollibre. Les escoles de música només poden demanar els discos de vinil.
- Tant els clients privats com les escoles de música només poden demanar un producte i fins que no facin la devolució, no en poden demanar un altre.
- La biblioteca té una web que li subministra els audiollibres, La biblioteca ha de emmagatzemar el nom de la web que subministra els audiollibres, el nom de la empresa propietària d'aquesta web, així com l'identificador de la web en el registre mercantil, el domicili social i el telèfon de contacte. Un web pot subministrar més d'un audiollibre, i un audiollibre és subministrat només per una web.
- La biblioteca necessita també emmagatzemar i gestionar informació dels seus treballadors.
- L'aplicació ha de tenir dos tipus de usuaris:
  - o l'administrador, pot dur a terme qualsevol tasca dins de l'aplicació (*usuari admin i contrasenya admin*)
  - o el treballador encarregat de gestionar els préstec de la biblioteca (*usuari encarregat i contrasenya encarregat*)
- Vosaltres mateixos heu de determinar quina informació heu de tenir en compte per desenvolupar l'aplicació. Per exemple, dels clients hauria de desar el dni, nom, cognom, email, ... ; dels llibres el títol, isbn, autors, la quantitat de llibres del mateix tipus que hi ha a la biblioteca, ... i dels llibres en paper, a més a més, la data d'impressió, la quantitat de fulles ... etc., etc.

- L'aplicació hauria de permetre la:
  - o Gestió (Alta / Modificació / Eliminació / Llistat) de tota la informació relacionada amb els llibres , els discos de vinil, els clients, els treballadors, etc. etc.
  - o Gestió i seguiment dels clientes, els seus préstecs/lloguers/devolucions, etc., et., així com la data d'inici i de devolució dels productes en préstec
  - o Cerca d'elements per diferents criteris. Per exemple, cerca de clients per DNI o pel telèfon o pel correu electrònic, .... cerca de llibres per autor o per editorial, ....
  - o Gestió i visualització en qualsevol moment de la capacitat d'allotjament de la biblioteca
  - o Creació d'una llista d'espera si el producte que demana el client encara està en préstec
  - o Valorar fer alguna penalització si fan la devolució fora de termini.
  - o ÉS MOLT IMPORTANT poder donar informació al client sobre el producte que demana, prestar-li un producte (un llibre, un audiolibre, un disco, etc) amb la seva data d'inici i devolució. També s'ha de controlar la quantitat de productes (llibres, discos, etc.) en préstec.
  - o Feu Menús per gestionar l'aplicació

## Requeriments tècnics del projecte

- Llenguatge de programació: **Java SE 11** o posterior.
- Cal utilitzar el paradigma de la **POO** .
- Creació de **menús**
- Diagrama **UML** amb el diagrama estàtic de classes.
- Definir jerarquies d'**herències** i **herències múltiples**
- Definir **relacions entre classes**
- Utilització de **polimorfisme**.
- Creació de **classes abstractes**.
- Creació **d'interfícies pròpies**
- Creació d'**enumeracions** amb atributs, mètodes, constructors, ...
- Les estructures de dades seran, o bé estàtiques (**array estàtic**), o bé (en cas de ser dinàmiques) un **ArrayList**.
- Utilitzar **Iterator** per recórrer les llistes
- Utilitzar altres formes per recórrer les llistes
- Implementació del **mètode compareTo()** de la interfície Comparable.
- Implementar el **mètode clone()** de la interfície Cloneable
- Implementació del **mètode equals()**
- Ús de la **classe Calendar** (és necessari treballar amb dates).
- **Redefinició de mètodes** de la classe Object.

- Creació de **paquets**.
- Implementació de **menús** per gestionar l'aplicació
- Documentació de totes les classes amb **Javadoc**.
- NO hi ha persistència de dades ( les dades no s'emmagatzemen ni en BDD, ni en fitxers, ...)

**ÉS OBLIGATORI** que totes les funcionalitats demanades s'hagin implementat i funcionin correctament o sigui, el projecte no és vàlid si l'aplicació no té un *main* que permeti la seva execució i ens permeti comprovar el seu correcte funcionament.

## Etapes del desenvolupament i lliurament

**Primer lliurament: Lliurar d'una primera versió dels punts 1, 2 i 3, el Dilluns 29 de novembre de 2021**

1. Realitzeu un **anàlisi dels requeriments** funcionals de l'aplicació fent una llista amb els teus requeriments
2. Dissenyeu les classes implicades, amb els seus atributs i les seves operacions; les jerarquies de classes necessàries, i les associacions i relacions entre classes, interfícies, etc. i representeu-les amb un **diagrama UML**.
3. **Implementa en Java totes les classes** segons diagrama UML que has dissenyat a l'apartat anterior.

**Segon lliurament: Lliurar TOT EL PROJECTE**

4. Els punts 1, 2, i 3 novament amb les modificacions que hi hagin fet posterior al primer lliurament
5. **Implementeu** la solució. Cal utilitzar el llenguatge Java SE 11 o posterior.
6. Heu de tenir preparat un **joc de dades per provar el funcionament de la aplicació**.
7. Documenteu les classes utilitzant **Javadoc**.
8. **Presenteu l'aplicació funcionat correctament** controlant els errors. Per exemple, els errors que es poden produir a l'entrada de dades, etc.

## Indicacions pel el PRIMER lliurament del projecte

**Indicacions pel el PRIMER lliurament del projecte (DILLUNS 29 DE NOVEMBRE DE 2021)**

- o Cal lliurar el projecte al moodle, abans de la data especificada al grup aula de LaNET.
- o S'ha d'entregar en fitxer zip o tar, amb el format  
*NomCognom1\_NomCognom2\_NomCognom3\_M03UF4\_PROJECTE.zip*

- Aquest fitxer comprimit contindrà:
  - Un document PDF amb els requeriments funcionals de l'aplicació.
  - Una imatge PNG/JPG/PDF amb el diagrama de classes (UML).
  - El codi font
  - Un fitxer README.txt amb els comentaris addicionals que considereu rellevants.
  - Un document PDF amb captures de pantalla descrivint el funcionament de l'aplicació i provant cadascun dels requeriments de la mateixa.

### Indicacions pel el SEGON lliurament del projecte

- Lliurament via Github: Dipòsit públic de nom daw2\_m03uf4\_projecte\_grupXX OR dam2\_m03uf4\_projecte\_grupXX (a on XX és el vostre número de grup pel projecte) dins del compte d'usuari de qualsevol dels integrants del grup.
- S'haurà d'enviar l'adreça URL del dipòsit públic de Github a on es troba el projecte a estela.simon@fje.edu. Dins del correu només s'ha d'indicar l'adreça URL del dipòsit i els noms dels membres del grup.

### Rúbrica d'avaluació

L'alumne ha de mostrar que la seva aplicació està en funcionament correctament (condició obligatòria)

	Rúbrica d'avaluació
1	Requeriments, més de 20 correctament --- <b>0,30 punts</b> Requeriments, menys de 20 correctament --- entre 0 i 0,30 punts No hi ha requeriments o estén tots malament --- 0 punts
2	UML correctament... fins a <b>0,40 punts</b>
3	Implementar en Java totes les classes segons el seu diagrama UML, utilitzant la POO correctament --- <b>0,60 punts</b> Implementa algunes correctament --- entre 0 i 0,60 punts No implementa cap ---0 punts
4	Més de tres herència i al menys una herència múltiple correctament --- <b>1 punt</b> Més de tres herència correctament i no fa herència múltiple --- entre 0,30 i 1 punt Només fa dos herència i una herència múltiple correctament --- entre 0,20 i 0,30 punts Només fa dos herència correctament i no fa herència múltiple --- entre 0,10 i 0,20 punts

	Només fa una herència i una herència múltiple correctament --- entre 0,05 i 0,10 punts Només fa una herència correctament --- entre 0 i 0,05 punts No implementa cap --- 0 punts correctament
5	Utilitza Polimorfisme correctament --- <b>0,30 punts</b> . No utilitza Polimorfisme --- 0 punts
6	Més d'una classes abstractes correctament --- <b>0,30 punts</b> Una classe abstracta correctament --- 0,25 punts Cap classe abstracta --- 0 punts
7	Més d'una Interfície pròpia correctament --- <b>0,30 punts</b> Una Interfície pròpia correctament --- 0,25 punts
8	Enumeracions amb atributs, mètodes, constructors, ... correctament. ---- <b>0,30 punts</b> Enumeracions sense atributs, mètodes, constructors, ... correctament. ---- 0,20 punts Cap enumeracions – 0 punts
9	<i>Implementa molts ArrayList correctament i al menys un array estàtic correctament. ---</i> <b>0,60 punts</b> <i>Implementa molts ArrayList correctament i cap array estàtic --- 0,50 punts</i> <i>Implementa molts array estàtics correctament i cap ArrayList --- 0,40 punts</i> <i>Implementa un ArrayList correctament i/o un array estàtic correctament --- 0,10 punts</i> <i>Cap array --- 0 punts</i>
10	Utilitza els mètodes de ArrayList per treballar amb les llista correctament --- <b>0,20 punts</b> No ho utilitza --- 0 punts
11	Utilitzar <i>Iterator</i> per recórrer les llistes correctament – <b>0,20 punts</b> Recorre les llistes correctament però sense iterators ---0,15 punts No recorre les llistes de cap manera --- 0 punts
11	Mètode <i>compareTo()</i> correctament --- <b>0,10 punts</b> No ho utilitza --- 0 punts
12	Mètode <i>clone()</i> --- <b>0,10 punts</b> No ho utilitza --- 0 punts

13	Mètode <i>equals()</i> --- <b>0,10 punts</b> No ho utilitza --- 0 punts
14	Classe Calendar --- <b>0,10 punts</b> No ho utilitza --- 0 punts
15	Implementació de menús per gestionar l'aplicació correctament --- <b>0,50 punts</b> Només fa un menú correctament --- 0,30 No fa menú --- 0 punts
16	Fa relacions entre classes (tenint en compte dues classes i tenint en compte tres classes) correctament --- <b>1,5 punts</b> Fa relacions entre classes (tenint en compte dues classes) correctament --- 1 punt No fa cap tipus de relació entre classes --- 0 punts
17	Redefinició de mètodes de la classe Object correctament --- <b>0,20 punts</b> No ho fa --- 0 punts.
18	Creació de paquets correctament --- <b>0,20 punts.</b> No ho fa --- 0 punts
19	Utilitza Javadocs correctament --- <b>0,20 punts</b> No ho fa --- 0 punts
20	Es fa un programa principal (main) a on es comprova que s'executen correctament TOTES les funcionalitats del projecte --- <b>2 punts</b> Es fa un programa principal (main) a on es comprova que s'executa correctament només una part de les funcionalitats del projecte --- entre 0 i 2 punts No hi ha programa principal o no s'executa correctament --- 0 punts
21	Un document PDF amb la llista de requeriments, captures de pantalla descrivint el funcionament de l'aplicació i provant cadascun dels requeriments de la mateixa --- <b>0,50 punts</b>