Escola del Clot
CFGS Desenvolupament d'Aplicacions Web
CFGS Desenvolupament d'Aplicacions Multiplataforma
M03. Programació
Prof. Estela Simón
Curs 2024-2025
M03. Programació II
SM3.2.- POO amb persistència en BD

Projecte (Primera Part): Fonaments de programació orientada a objectes

Descripció del projecte

Es desenvoluparà un prototip d'aplicació d'escriptori basada en **Java** per gestionar la biblioteca "Baracoa SA", situada a la comarca de la **Terra Alta**. Aquesta aplicació permetrà millorar la gestió de les operacions més comuns, com ara:

- **Alta / Modificació / Eliminació / Llistat** de clients, llibres, treballadors, etc.
- **Gestó dels préstecs**, incloent-hi dates d'inici i devolució.
- **Cerca per criteris** (DNI, telèfon, email, autor, editorial, etc.).
- **Control de disponibilitat** i gestió de llistes d'espera.
- **Possibilitat de penalitzacions** per devolucions fora de termini.

Funcionalitats principals

- Els clients poden ser **privats** o **escoles de música**:
    - **Clients privats:** Només poden demanar llibres (paper i audiollibre).
    - **Escoles de música:** Només poden demanar discos de vinil.
    - Només es pot tenir **un préstec actiu** a la vegada.
- La biblioteca obté **audiollibres d'un proveïdor web**, del qual cal emmagatzemar:
    - Nom, empresa propietària, identificador mercantil, domicili social i telèfon.
- **Gestió de treballadors**, amb dades pertinents.
- **Menús interactius** per gestionar l'aplicació.

Requeriments tècnics

- **Llenguatge:** Java
- **Paradigma:** Programació Orientada a Objectes (POO)
- **Estructura:**

    - Creació de menús per gestió.
    - **Diagrama UML** de classes.
    - **Herències i herències múltiples.**
    - **Relacions entre classes.**
    - **Polimorfisme i classes abstractes.**
    - **Interfícies i enumeracions amb atributs i mètodes.**
- **Estructures de dades:**
    - **ArrayList** per dades dinàmiques.
    - **Iterators** per recórrer llistes.
    - **Mètodes de llistes com `compareTo()`, `clone()`, `equals()`.**
- **Tractament de dates amb `Calendar`**
- **Documentació del codi amb Javadoc.**
- **No es permet la persistència de dades en bases de dades ni en fitxers.**

Implementació del `Main`

L'aplicació ha de comptar amb un `main` que permeti la seva execució i validació. Aquest haurà d'incloure:

- **Dades de prova prèviament carregades**, incloent llibres, discos i clients.
- **Opcions interactives** per executar totes les funcionalitats.
- **Control d'errors** en l'entrada de dades.

Etapes de desenvolupament i lliurament

Primer lliurament:

1. **Anàlisi de requeriments**
2. **Disseny UML** (classes, atributs, operacions, herències i relacions)
3. **Implementació de les classes en Java**

Segon lliurament:

4. **Implementació completa de la solució.**
5. **Dades de prova preparades.**
6. **Documentació del codi amb Javadoc.**
7. **Presentació i validació del funcionament de l'aplicació.**

Format de lliurament

El projecte es lliurarà en format **ZIP**, amb:

- **Document PDF** amb requeriments.
- **Imatge del diagrama UML.**
- **Codi font complet.**

- **README.txt** amb informació addicional.
- **Document PDF amb captures de pantalla demostrant el funcionament.**

També es podrà lliurar mitjançant **GitHub**, enviant l'enllaç al professor.

Criteris d'avaluació

- **Completesa dels requeriments (0,30 punts).**
- **Diagrama UML correcte (0,40 punts).**
- **POO i estructures adequades (0,60 punts).**
- **Herència, polimorfisme i interfícies (1,30 punts).**
- **Gestió de llistes i recorregut correcte (1 punt).**
- **Menús i gestió de dades (0,50 punts).**
- **Execució correcta de totes les funcionalitats (2 punts).**
- **Documentació, diagrama UML i captures de pantalla (0,50 punts).**

Nota final:

El projecte **només serà vàlid si inclou un `main` executable que permet comprovar totes les funcionalitats.**
