

DAM2 – Lliuraments UF6

Pràctica. Persistència ObjectDB + JavaFX

L'exercici consisteix en modificar l'aplicació **JavaFX** de gestió de l'avaluació dels exàmens per integrar la persistència amb **ObjectDB**.

Apartat 1. Importat projecte

Importar el projecte «**201718-UF6-alumnes.zip**» del Moodle

Apartat 2. Gestionar persistència.

Classe «**Model**»

initConnexio()	Inicialitza els atributs emf (EntityManagerFactory) i em (EntityManager)
closeEM()	Tancar emf (EntityManagerFactory) i em (EntityManager)
getExamens(): examens	Consulta de tots els exàmens des de la base de dades per actualitzar la llista «examens»
setExamens(examens)	Actualitzar els exàmens que gestiona l'Entity Manager
getEstudiants(): estudiants	Consulta de tots els estudiants des de la base de dades per actualitzar la llista «estudiants»
setEstudiants(estudiants)	Actualitzar els estudiants que gestiona l'Entity Manager
getUnitats(): unitats	Consulta de totes les unitats des de la base de dades per actualitzar la llista «unitats»
setUnitats(unitats)	Actualitzar les unitats que gestiona l'Entity Manager
updateUnitatFormativa(unitat)	Actualitzar la unitat (Unitat Formativa) a la base de dades
addUnitatFormativa(cicle, modul, num, titol, hores)	Afegir una nova Unitat Formativa a la base de dades
removeUnitatFormativa(unitat)	Esborrar la unitat de la base de dades
updateEstudiant(estudiant)	Actualitzar l'estudiant a la base de dades
removeEstudiant(estudiant)	Esborrar l'estudiant de la base de dades
addEstudiant(nom, cognoms, edat)	Afegir un nou Estudiant a la base de dades



Examen addExamen(unitat)	Afegir un nou Examen de la unitat a la base de dades
removeExamen(examen)	Esborrar l'examen de la base de dades

Els següents mètodes es relacionen amb l'atribut «**selectedExamen**», aquest serà la instància actualment seleccionada per exemple a la vista «**Examins > Avaluació**»

The screenshot shows the 'Avaluacio' web application. The top navigation bar includes 'Fitxer', 'Unitats Formatives', 'Alumnes', 'Examens', 'Consultes', and 'Ajuda'. The main header features the 'Avaluacio' title and the 'INS Marianao' logo. Below the header, a dropdown menu for 'Examen' is set to 'DAM - M03. Programació. UF4: Programació orientada a objectes (POO). Fonaments'. The interface is divided into two main sections: 'Assistencia' and 'Resultats'. The 'Assistencia' section contains a table of students with columns 'Nom' and 'Lliurat', and a 'Lliuraments' section with a 'Revisions' table. The 'Resultats' section contains a table with columns 'Estudiant', 'Nota', and 'Revisar'. Arrows indicate the flow of data between the 'Assistencia' and 'Resultats' sections.

inscriureEstudiantSelectedExamen(estudiant)	Inscriure estudiant a «selectedExamen»
anularMatriculaEstudiantSelectedExamen(estudiant)	Anul·lar matrícula estudiant a «selectedExamen»
apilarLliuramentSelectedExamen(estudiant)	Apilar un nou lliurament de l'estudiant a «selectedExamen»
desapilarLliuramentSelectedExamen()	Treure el darrer lliurament apilat (cim de la pila) a «selectedExamen»
corregirLliuramentSelectedExamen(nota)	Corregir el darrer lliurament apilat (cim de la pila) a «selectedExamen»
solicitarRevisioSelectedExamen(lliurament)	Sol·licitar revisió del lliurament a «selectedExamen»
atendreRevisioSelectedExamen(nota)	Atendre primera revisió (posar nota) a «selectedExamen»



Exemples de mètodes de get i set. Molt important, per garantir que les dades de les vistes s'actualitzen correctament, un cop inicialitzades les estructures (ObservableList) no es poden tornar a inicialitzar, només afegir i treure elements.

```
private ObservableList<Examen> examens;  
  
[ ... ]  
  
public ObservableList<Examen> getExamens() {  
    TypedQuery<Examen> query = this.em.createNamedQuery("Examen.findAll", Examen.class);  
    this.examens.clear();  
    this.examens.addAll(query.getResultList());  
    return this.examens;  
}  
  
public void setExamens(ObservableList<Examen> examens) throws Excepcio {  
    try {  
        this.em.getTransaction().begin();  
        for (Examen examen : examens) this.em.merge(examen);  
        this.em.getTransaction().commit();  
    } catch (Exception e) {  
        throw new Excepcio("Model", e.getMessage());  
    } finally {  
        if (em.getTransaction().isActive()) em.getTransaction().rollback();  
    }  
}
```

Apartat 3. Implementar consultes

Afegir les següents consultes paginades a la vista «Obrir > Consultes»

- **Consulta 1:** Seleccionar exàmens (cicle, mòdul, número, títol i hores de la unitat formativa de l'examen), el nombre total de preguntes, el nombre total d'alumnes, el nombre total de lliuraments, el nombre total de correccions, el nombre total de revisions. Ordenat per cicle, mòdul i número. El resultat paginat: Pàgines de 1 element
- **Consulta 2:** Seleccionar exàmens (cicle, mòdul, número, títol i hores de la unitat formativa de l'examen), el nombre total de correccions de l'examen i la mitjana de les notes de les correccions. Ordenat per cicle, mòdul i número. Resultat paginat: Pàgines de 1 element
- **Consulta 3:** Seleccionar lliuraments (cicle, número de la unitat formativa de l'examen del lliurament, Nom+ " "+Cognoms concatenats i edat de l'alumne del lliurament, la nota de l'examen. Ordenat per cicle, número i nota descendent. Resultat paginat: Pàgines de 3 elements

Aquesta consulta 3 té les següents condicions: lliuraments que tingui nota > a un

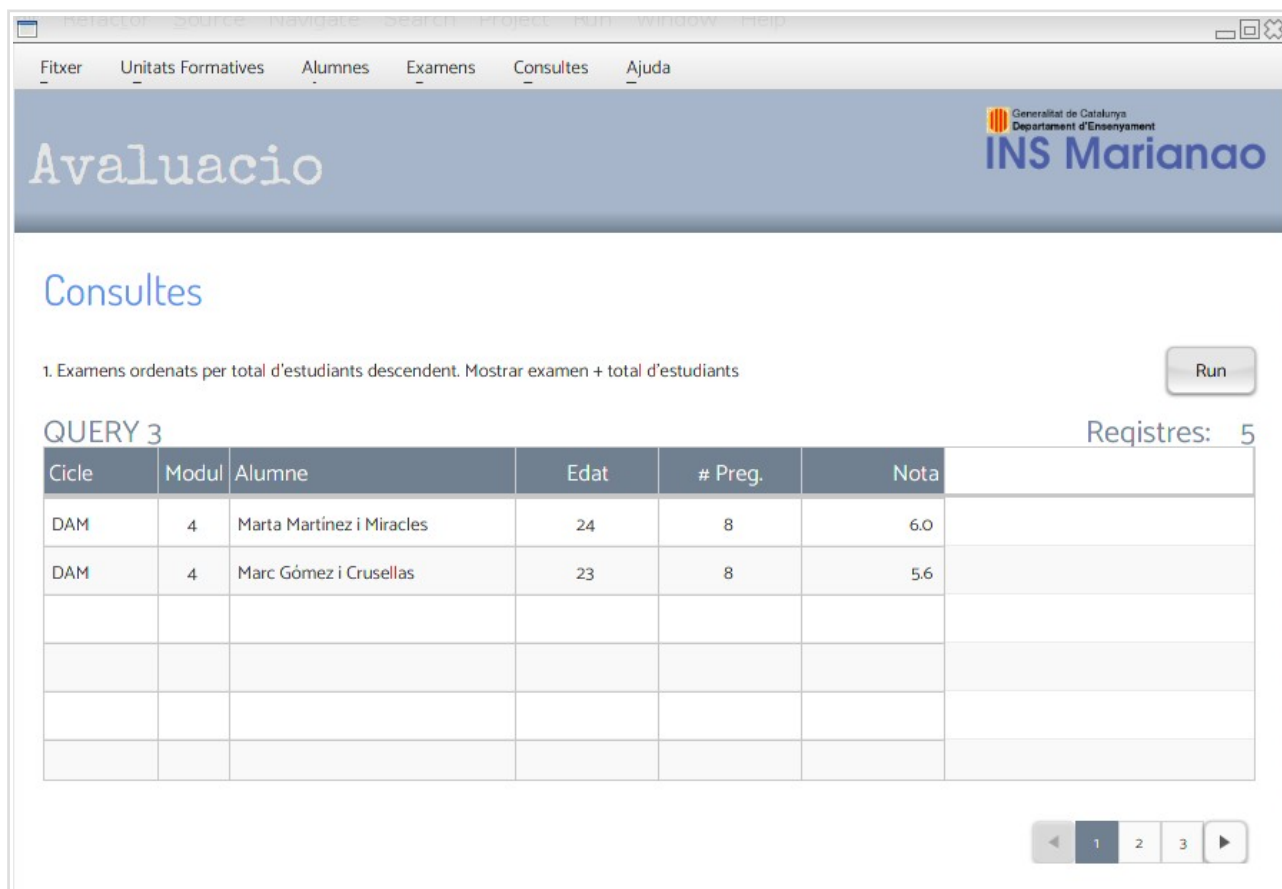
paràmetre PARAM1 i dels alumnes que NO tinguin un nom d'una llista PARAM2.

PARAM1 i PARAM2 han de ser camps de text afegits a la vista «Consultes»

Classe «Model»

Mètodes d'exemple

ModelQueryResult query1(pag, perPag)	Mètode d'exemple per obtenir una pàgina d'una consulta i carregar-la a la taula de la vista «Consultes»
--------------------------------------	---



Fitxer Unitats Formatives Alumnes Exams Consultes Ajuda

Avaluacio

Generalitat de Catalunya
Departament d'Ensenyament
INS Marianao

Consultes

1. Exams ordenats per total d'estudiants descendent. Mostrar examen + total d'estudiants

Run

QUERY 3 Registres: 5

Cicle	Modul	Alumne	Edat	# Preg.	Nota
DAM	4	Marta Martínez i Miracles	24	8	6.0
DAM	4	Marc Gómez i Crusellas	23	8	5.6

1 2 3



Exemples de mètodes de consulta. Retornen una instància de «**ModelQueryResult**» que gestiona les dades de la consulta i tot el necessari per afegir-les a un component «**TableView**» de JavaFX. Té els següents camps: **Títol** de la consulta, **total** d'elements, **noms** de les columnes, **amplada** de les columnes, **alineació** de les columnes, **resultat** de la consulta.

```
/**
 * Consultes
 *
 */
public ModelQueryResult query1(int pagina, int perPagina) {
    Vector<String> colNames = new Vector<String>(Arrays.asList(new String[] {"Col1", "Col2"}));
    Vector<Integer> colWidth = new Vector<Integer>(Arrays.asList(new Integer[] {60, 200}));
    Vector<String> colAlign = new Vector<String>(Arrays.asList(
        new String[] {"center-column", "left-column"}));

    String strQuery = "SELECT u.col1, u.col2 ";
    strQuery += " FROM Classe u ";
    strQuery += " ORDER BY u.col3";

    TypedQuery<Object[]> query = this.em.createQuery(strQuery, Object[].class);

    int count = query.getResultList().size();

    query.setFirstResult((pagina - 1) * perPagina).setMaxResults(perPagina);

    return new ModelQueryResult("QUERY 1", count, colNames, colWidth, colAlign,
        query.getResultList() );
}
```