

Applications orientées données (NSY135)

Présentation du cours

Auteurs: Raphaël Fournier-S'niehotta et Philippe Rigaux
(prenom.nom@cnam.fr)

Département d'informatique
Conservatoire National des Arts & Métiers, Paris, France

Plan du cours

- 1 **Sujet du cours**
- 2 Objectifs
- 3 Prérequis
- 4 Organisation
- 5 Les ressources

Sujet : le problème

Ce cours est dédié aux mécanismes **d'interaction** entre une **application informatique** et une **base de données**. Très précisément :

- les applications auxquelles on s'intéresse sont écrites en orienté-objet, avec un langage comme java ou PHP.
⇒ dans ces applications, les **données** sont des **objets**, liés par des **références**.
- les bases de données sont des bases relationnelles, comme MySQL, Oracle, PostgreSQL, etc.
⇒ dans ces bases, les **données** sont des « lignes » dans des « tables », liées par des valeurs de clés communes.

Le problème

La cohabitation des deux **représentations** différentes soulève des problèmes d'intégration, de conversion, d'implémentation, et plus généralement de **conception** et **d'ingénierie logicielle**.

Sujet : la solution

Sujet principal

Le cours est consacré aux **méthodes** (conception et ingénierie) permettant d'intégrer une base de données à une application objet.

Mots clés : *Object-relational mapping* (ORM), modèles de données, accès aux données, architecture logicielle.

Exemples. En java : JPA (Hibernate, EclipseLink) ; en PHP (Doctrine, CakePHP) ; en Python (SQLObject, SQLAlchemy)

Sujet : la solution

Sujet principal

Le cours est consacré aux **méthodes** (conception et ingénierie) permettant d'intégrer une base de données à une application objet.

Mots clés : *Object-relational mapping* (ORM), modèles de données, accès aux données, architecture logicielle.

Exemples. En java : JPA (Hibernate, EclipseLink) ; en PHP (Doctrine, CakePHP) ; en Python (SQLObject, SQLAlchemy)

Sujet secondaire

Le cours se place dans la perspective de l'utilisation **d'un framework** pour illustrer les bonnes pratiques de structuration des applications.

Mots clés : *frameworks de développement*, modèle MVC, applications Web.

Exemples. En java : Spring, JSF ; en PHP : Zend, Symfony ; en Python : Django ; etc.

Plan du cours

- 1 Sujet du cours
- 2 Objectifs
- 3 Prérequis
- 4 Organisation
- 5 Les ressources

Objectifs

Ce que vous allez apprendre en détail

- **Principes ORM** : la modélisation des accès aux BD relationnelles (**JPA**) en **java**.
- **la mise en œuvre** avec un environnement logiciel standard : Eclipse, Hibernate, MySQL.

Objectifs

Ce que vous allez apprendre en détail

- **Principes ORM** : la modélisation des accès aux BD relationnelles (**JPA**) en **java**.
- **la mise en œuvre** avec un environnement logiciel standard : Eclipse, Hibernate, MySQL.

Ce que vous allez apprendre pour l'essentiel

- **L'intégration** d'une couche ORM dans une application complète (application Web)
- le rôle d'un **framework** et de quelques **design patterns** importants.

Objectifs

Ce que vous allez apprendre en détail

- **Principes ORM** : la modélisation des accès aux BD relationnelles (**JPA**) en **java**.
- **la mise en œuvre** avec un environnement logiciel standard : Eclipse, Hibernate, MySQL.

Ce que vous allez apprendre pour l'essentiel

- **L'intégration** d'une couche ORM dans une application complète (application Web)
- le rôle d'un **framework** et de quelques **design patterns** importants.

Ce que vous n'apprendrez pas, ou peu

Tout le reste, et notamment : les bases relationnelles, conception et SQL ; la programmation objet ; les frameworks MVC ; le JEE dans son ensemble.

Plan du cours

- 1 Sujet du cours
- 2 Objectifs
- 3 Prérequis**
- 4 Organisation
- 5 Les ressources

Prérequis

Les connaissances suivantes sont requises pour suivre ce cours.

- **Bases de la programmation objet**, et des notions de base en java.
- **Compréhension des bases relationnelles**, soit au moins la conception d'un schéma, SQL, et des notions de base sur les transactions.
- **Une aisance minimale dans un environnement de développement**. Editer un fichier, lancer une commande, ne pas paniquer devant un nouvel outil, savoir résoudre un problème avec un minimum de ténacité, etc.

⇒ Vous n'avez pas besoin d'être un développeur aguerri, mais vous devez reproduire les exemples donnés.

⇒ **La mise en œuvre**, dans le cadre d'un petit projet, est essentielle pour vérifier la compréhension du sujet.

Environnement matériel et logiciel

Le cours repose beaucoup sur la mise en pratique. Vous avez besoin d'un ordinateur pour travailler.

- au Cnam, tout vous est fourni ;
- Cnam à distance, ou en auditeur libre, il vous faut une machine raisonnablement puissante et dotée d'au moins 8 GO de mémoire ;
- tous les systèmes sont possibles : Linux, Mac/OS ou Windows.

Environnement matériel et logiciel

Le cours repose beaucoup sur la mise en pratique. Vous avez besoin d'un ordinateur pour travailler.

- au Cnam, tout vous est fourni ;
- Cnam à distance, ou en auditeur libre, il vous faut une machine raisonnablement puissante et dotée d'au moins 8 GO de mémoire :
- tous les systèmes sont possibles : Linux, Mac/OS ou Windows.

Tous les logiciels utilisés sont libres de droits. Nous travaillons essentiellement avec

- **Eclipse**, notre environnement intégré pour le contrôle de notre projet.
- **Tomcat**, un serveur web pour applications java.
- Java et **Hibernate**.
- **MySQL** pour la base de données.

L'installation de ces outils (et leur utilisation) sont couverte dans le cours.

Plan du cours

- 1 Sujet du cours
- 2 Objectifs
- 3 Prérequis
- 4 Organisation**
- 5 Les ressources

Organisation du cours

Le cours est découpé en **chapitres**, couvrant un sujet bien déterminé, et en **sessions**.

L'unité de travail est la session. Nous essayons de structurer les sessions pour qu'elles ne demandent environ 2 heures de travail personnel (bien sûr, cela dépend également de vous).

Organisation du cours

Le cours est découpé en **chapitres**, couvrant un sujet bien déterminé, et en **sessions**.

L'unité de travail est la session. Nous essayons de structurer les sessions pour qu'elles ne demandent environ 2 heures de travail personnel (bien sûr, cela dépend également de vous).

Pour assimiler une session vous pouvez combiner les ressources suivantes :

- **La lecture** du **support en ligne**, également disponible en **PDF** ou en **ePub**.
- **Le suivi du cours**, en vidéo ou en présentiel.
- **Le test des exemples de code** fournis dans chaque session.
- **La réalisation des exercices** proposés en fin de session.

La réalisation des exercices est essentielle pour vérifier que vous maîtrisez le contenu.

Mener votre travail personnel

Vous devez maîtriser le contenu des sessions **dans l'ordre où elles sont proposées.**

Commencez par lire le support, jusqu'à ce que les principes vous paraissent clairs.

Reproduisez les exemples de code. Tous les exemples donnés sont testés et doivent donc fonctionner.

Cherchez à résoudre les problèmes par vous-mêmes au besoin : c'est le meilleur moyen de comprendre.

Finissez par les exercices. Organisez vos solutions dans un espace bien structuré (nous donnons des conseils pour cela).

Mener votre travail personnel

Vous devez maîtriser le contenu des sessions **dans l'ordre où elles sont proposées.**

Commencez par lire le support, jusqu'à ce que les principes vous paraissent clairs.

Reproduisez les exemples de code. Tous les exemples donnés sont testés et doivent donc fonctionner.

Cherchez à résoudre les problèmes par vous-mêmes au besoin : c'est le meilleur moyen de comprendre.

Finissez par les exercices. Organisez vos solutions dans un espace bien structuré (nous donnons des conseils pour cela).

Nous sommes là pour vous aider

Si vous êtes inscrits au Cnam, nous sommes là pour vous ; sinon, nous vous assisterons dans la mesure de nos disponibilités.

Evaluation

Nous proposons un projet consistant à réaliser une mini-application fortement centrée sur la conception du modèle de données et sa réalisation avec JPA/Hibernate.

La réalisation du projet doit montrer que vous maîtrisez les notions suivantes :

- conception du modèle de données avec UML (ou équivalent) en distinguant bien les classes, les associations et les relations de généralisation (héritage) ;
- création du schéma de la base de données et le modèle Java/JPA ;
- implantation d'une mini-application Web MVC qui permet de naviguer dans la base et de consulter les données ;
- et, en option, quelques fonctions avancées montrant une transaction, une approche d'optimisation des performances, etc.

Le projet est suivi et évalué par les enseignants du Cnam.

Plan du cours

- 1 Sujet du cours
- 2 Objectifs
- 3 Prérequis
- 4 Organisation
- 5 Les ressources

Ressources

Contact avec les enseignants : Raphaël Fournier-S'niehotta et Philippe Rigaux (prénom.nom@cnam.fr).

- Généralités sur le cours et documents complémentaires :
<http://www.bdpedia.fr/applications>
 - Présentations (slides)
 - Liens vers les vidéos
 - Autres ressources
- Support de cours en HTML : <http://orm.bdpedia.fr/>
- Support de cours en PDF :
<http://orm.bdpedia.fr/files/orm.pdf>
- Support de cours en ePub :
<http://orm.bdpedia.fr/files/orm.epub>

Pour ceux/celles qui suivent le cours au Cnam : <http://lecnam.net>.