

Cours 420-4A6-LI Applications monopostes I Hiver 2020 Cégep Limoilou Département d'informatique	Tp 3  Travail intégrateur  25 %
---	---

Nom : \_\_\_\_\_

Gr : 1

Nom : \_\_\_\_\_

**Objectif**

- Créer, un modèle (diagramme) de classes pour illustrer les composants de votre application.
- Utiliser du FXML pour créer une interface graphique et des boîtes de dialogue personnalisées pour mettre à jour de l'information.
- Lier du FXML à du JavaFX.
- Respecter le modèle MVC avec FXML vu dans le cours.
- Utiliser des concepts à partir d'information prise sur le web.
- Utiliser une base de données et des fichiers d'information.

**Activités à réaliser**

- Vous avez à programmer une application de gestion de permis animalier.

**Gestion de permis animaliers**

Fichier Permis Listes de données Aide

**Liste de numéros de permis**

**Permis**

Numéro  Date de début

Territoire  Date de fin

**Animal**

Nom  Type

Sexe ☐ Mâle ☐ Femelle ☒ Inconnu Poids en Kg

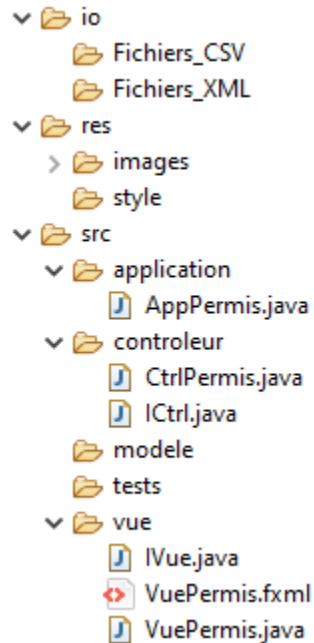
Date de naissance  Couleur

Caractéristiques supplémentaires ☐ Vacciné ☐ Stérilisé ☐ Micropuce ☐ Dangereux

Nouveau Ajouter Modifier Quitter ?

## À remettre et/ou à présenter

- Pour le code source, les ressources et les fichiers de données, vous devez respecter minimalement l'arbre de packages, de dossiers et de fichiers, suivant (les noms et le découpage). Je vous laisse décider du reste des ressources que vous aurez besoin :



- Pour la disposition des contrôles et leur mise en forme, il faut respecter l'image écran présentée ci-haut et les consignes du présent travail.
- Tout votre code doit être documenté avec de la JavaDoc.
- Des tests unitaires JUnit doivent être faits pour vos classes métiers du package « modele ». Une preuve de couverture sera demandée.
- Une **démonstration** du fonctionnement de votre code devra obligatoirement être faite dans la semaine du **18 mai 2020**.
- Avant de faire votre démonstration, vous devez remettre votre code du projet « Eclipse » complet, compressé (.zip) dans un fichier du nom « noms des équipiers – Tp3.zip » en le déposant sur le réseau dans le dossier « Q:\!Dépôt\JGoulet\420-4A6-H20\Tp 3 – Code ».

### Durée de l'activité

- Ce travail peut être réalisé en équipe de 2 maximum.
- Vous avez jusqu'au 22 mai à 12h pour faire votre démo. Nous allons devoir faire un horaire pour la planification des démos :o).

### Avant de commencer

- Il faut avoir réalisé quelques exercices dans les formatifs 6 et 7.
- Il faut avoir lu les documents des notes de cours des semaines 8, 10 et 11 inclusivement.
- Il faut avoir compris les exemples fournis par le professeur pour les semaines 8, 10 et 11.

### Étape 1 : Stratégie et diagramme de classe (non obligatoire, mais fortement recommandé)

- Votre première tâche consiste à fabriquer un diagramme de classes de toutes vos classes impliquées dans votre architecture MVC. Utilisez l'outil « **Visual Paradigm** » ou « **ObjectAid** » installé dans Eclipse (revoir, dans le formatif 2, les références à la fin du cas 1).

Cette étape est fortement recommandée pour vous permettre de prendre plusieurs décisions. En plus

de vous permettre de réfléchir à votre stratégie de conception, il vous permettra de déterminer les interfaces, les classes abstraites et les héritages. Bien entendu, il n'y a rien de parfait, faites de votre mieux. Je peux, si vous me le demandez, regarder votre diagramme avec vous. Commencez par lire le présent travail au complet, ce qui va vous permettre de faire un premier diagramme papier de haut niveau.

- Comme la section « À remettre ou à présenter » le précise, il n'y a rien à remettre à cette étape.

### Étape 2 : Programmation et validation des classes métiers

- Programmer vos classes en lien avec les données de la partie « modèle » du MVC, les classes de manipulation des données avec la BD et de lecture et écriture dans les fichiers.
- Programmer vos classes en lien avec l'application pour permettre de répondre aux événements de la vue (ajout, modification et suppression de données) de la partie « modèle » du MVC, les classes de traitements des données, de la gestion de certaines des listes et les classes de données transitoires.
- Les tests unitaires doivent être faits pour vos classes du package « modele ». Une preuve de couverture de code sera demandée.

### Étape 3 : Réalisation et programmation du fonctionnement de l'interface

#### Les composants :

- Reproduire la vue telle que présentée dans cette figure et utiliser une feuille de style pour améliorer son apparence (le css est obligatoire, mais son contenu est laissé à votre convenance). Une preuve sera demandée pour voir l'utilité de votre feuille de style.

1. Le champ de recherche permet de rechercher dans la liste des numéros de permis. La recherche se fait lorsqu'on appuie sur le bouton (loupe), ce qui positionne le numéro de permis trouvé, visible dans la liste et présente les valeurs de ce permis dans les champs de la fenêtre, c'est le permis courant.

Si aucun permis n'est trouvé, c'est le permis courant avant la recherche qui reste sélectionné. Le champ de recherche est désactivé, si la liste de numéros est vide.

2. Le champ « Numéro » de permis, il n'est pas éditable, il est attribué par votre système et doit être unique. Les numéros supprimés ne sont pas réutilisés. Il est toujours croissant. Il est vide si pas de permis courant.

3. Les champs « Date de début », « Date de fin » et « Date de naissance » sont de vrais champs date (composant date). Ils ne sont pas éditables, mais la date peut être modifiée à l'aide du composant.

Lors de l'ajout d'un nouveau permis, à l'initialisation des champs, la date de début prend la date du jour et la date de fin prend la valeur de la date de début plus 1 an. La date de naissance non définie dans les données importées prendra la date de début du permis. Ils sont vides si aucun permis courant.

**Attention :**

- La date de début doit toujours être plus ancienne que la date de fin de 1 mois au moins.
  - La date de naissance de l'animal doit toujours être plus ancienne ou égale à la date de début.
4. Le bouton supprimé permet de supprimer l'enregistrement courant avec une demande de confirmation à l'utilisateur. Le permis est supprimé de la BD et de la liste de numéros de permis. Le nouveau permis courant est le numéro précédent de celui supprimé dans la liste. Si la liste est vide, donc aucune sélection possible, les champs du formulaire doivent respecter les consignes ci-après mentionnées dans cette situation.
  5. Les champs « Territoire » et « Type » sont des listes déroulantes non éditables et triées en ordre croissant. Si le territoire ou le type n'est pas défini dans les données, le champ doit afficher le mot « Inconnu ».

Le bouton à côté du champ permet d'ouvrir une boîte de dialogue personnalisée qui permet de faire la gestion (ajout et modifier seulement) de nouveaux territoires ou de nouveaux types. Après un ajout ou une modification, la liste est mise à jour.

Voici les valeurs à mettre dans les tables (BD) pour les listes « Territoire » et « Type » de départ :

Liste de territoires au départ du système :

**Inconnu**, Vimont, Sainte-Dorothée, Sainte-Rose, Saint-François, Chomedey, Pont-Viau, îles-Laval, Laval-des-Rapides, Laval-Ouest, Laval-sur-le-Lac, Saint-Vincent-de-Paul, Duvernay, Auteuil, Fabreville

Liste de type au départ du système :


**Inconnu**, Chat, Chien

**NOTE :** Voici un lien à partir duquel on vous présente comment lier et administrer les données d'une boîte de dialogue personnalisée en FXML : <https://code.makery.ch/fr/library/javafx-tutorial/part3/>.

6. Le champ « Poids » peut prendre les valeurs réelles entre 0 à 500 kg. La valeur 0 ne correspond à aucune valeur, elle sera la valeur par défaut ou si le poids n'est pas défini dans les données importées.
7. Le champ « Couleur » est une liste déroulante éditable et triée en ordre croissant. Vous pouvez donc choisir une couleur à partir des couleurs déjà disponibles des autres animaux du système (BD) ou directement entrer la couleur voulue. La nouvelle couleur sera sauvegardée avec les données de l'animal et elle sera ajoutée à la liste. La couleur par défaut ou lorsqu'elle n'est pas définie dans les données importées est « Inconnu ». Prévoir de mettre le mot « Inconnu » dans la liste de couleur si elle n'est pas dans les données du système (BD tables permis).
8. La case à cocher « Dangereux » mettra son texte en **rouge et gras** si elle est sélectionnée, pour permettre d'attirer l'attention de l'utilisateur.  
Les boutons « Nouveau » et « Ajouter » permettent d'ajouter un nouveau permis. Ils sont mutuellement dépendants. Lorsque je suis sur un enregistrement déjà existant le bouton « Nouveau » est disponible, mais pas le bouton « Ajouter », c'est le mode normal. Si je clique sur le bouton « Nouveau » le système me propose un nouveau numéro de permis, met les autres champs à des valeurs d'ajout (voir le tableau ci-après), désactive le bouton « Nouveau » et active le bouton « Ajouter », c'est le mode ajout. Si je clique sur le bouton « Ajouter », la filtration des données s'effectue (notification s'il y a des problèmes) et si tout est conforme, le nouveau permis est enregistré dans la BD et son numéro est ajouté à la liste

de permis dans l'application. Il y a aussi une notification pour les données non valides. Après l'ajout, l'application revient en mode normal avec le nouveau permis, comme permis courant.

9. **Le bouton « Modifier » est disponible seulement si une des valeurs du permis courant a été modifiée et qu'on est en mode normal.** Lorsqu'il est cliqué, la filtration des données s'effectue (notification s'il y a des problèmes), la validation se fait aussi avec notifications et si tout est conforme, les modifications du permis sont enregistrées dans la BD. L'application revient en mode normal avec le permis modifié, comme permis courant et le bouton « Modifier » se désactive.
10. La liste de permis permet d'afficher, en ordre croissant, tous les numéros de permis enregistrés dans la BD. Le numéro du permis courant (sélectionné) est toujours visible dans la liste. À chaque changement de sélection dans la liste, les champs du permis courant se mettent à jour. C'est aussi vrai pour la recherche et l'ajout d'un nouvel enregistrement.
  - Le champ nom est vide par défaut, longueur maximum de 40 caractères.
  - Le sexe par défaut est « inconnu » et en ajout c'est « Femelle ».
  - Aucune caractéristique n'est sélectionnée par défaut.
  - Le bouton « Quitter » quitte l'application avec le consentement de l'utilisateur.
  - Le bouton « ? » présente un court texte de ce que fait cette application dans une boîte de dialogue d'information.
  - Voici un tableau permettant de voir la disponibilité des boutons et des options du menu « Permis » selon l'état du système :

État du système	Nouveau	Ajouter	Modifier	Supprimer 
Lors de l'ouverture de l'application, liste de no de permis vide.	actif	inactif	inactif	inactif
Lors de l'ouverture de l'application, liste de no de permis remplie, avec sélection d'un permis.	actif	inactif	inactif	actif
Bouton « Nouveau » a été cliqué. (Reste comme ça tant que les données ne sont pas ajoutées ou que l'on sélectionne un numéro dans la liste de numéros de permis).	inactif	actif	inactif	inactif
Bouton « Ajouter » a été cliqué et données bonnes.	actif	inactif	inactif	actif
Sélection d'un numéro dans la liste	actif	inactif	inactif	actif
Modification d'une des valeurs du permis courant. <b>(Attention, ici comme vous allez passer en mode modification, il est important de notifier l'utilisateur avant de passer dans un autre mode pour qu'il puisse sauvegarder les données).</b>	actif	inactif	actif	actif

- Voici un tableau permettant de voir l'état des champs du formulaire selon l'état du système. L'appellation « Les champs du formulaire » inclut le champ de recherche et la liste de numéros de permis, sauf avis contraire :

État du système	État des champs du formulaire
Lors de l'ouverture de l'application, liste de no de permis vide (tables de permis vide). <u>On est en mode normal</u> avec une liste de numéros de permis vide.	Tous les champs du formulaire sont désactivés, sauf les boutons de gestion des listes de « territoires » et des « types » et les options de menu associées.
Lors de l'ouverture de l'application, liste de no de permis remplie (tables de permis avec données), le premier numéro de permis est sélectionné. <u>On est</u>	Tous les champs du formulaire sont activés et remplis par les données du <u>permis courant</u> .

<u>en mode normal</u> avec une liste de numéros de permis <b>remplie</b> .	
<u>En mode ajout</u> (bouton nouveau cliqué).	Tous les champs du formulaire sont activés, le numéro de permis est calculé et tous les autres champs sont remplis par des données indiquées ci-après. Pour le champ de recherche et la liste de numéros, je vous laisse gérer cette situation.
Sortir du <u>mode ajout</u> .	Retour au mode normal selon que la liste de numéros de permis est vide ou pas!
<u>Mode modification</u> .	Comme le mode normal. Il faut juste faire attention lors de la sortie de ce mode, à ne pas perdre les modifs.

- Voici un tableau permettant de voir les valeurs des données lues ou des champs du formulaire selon les circonstances d'utilisation et les limitations associées (domaine de données). Ici on n'inclut pas le champ de recherche ni la liste de numéros de permis :

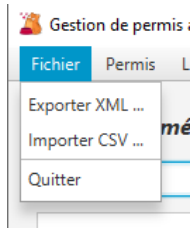
Champ	Non définie lors de la lecture des données CSV	Par défaut	En ajout	Domaine de donnée à tenir compte en tout temps
Numéro	rejeté	vide	calculé	Un entier croissant, unique et non réutilisable, non éditable
Date début	du jour	vide	du jour	Une date valide plus récente ou égale à la date de naissance
Date fin	de début + 1 an	vide	de début + 1 an	Une date valide plus récente de moins 1 mois à la date de début
Territoire	inconnu	vide	inconnu	Selon la liste de départ ou mise à jour
Nom	vide	vide	vide	40 caractères maximum
Type	inconnu	vide	chat	Selon la liste de départ ou mise à jour
Sexe	inconnu	Inconnu	femelle	Selon les choix et obligatoirement un
Poids	0	0	0	Un réel entre 0 et 500 kg
Date naissance	date de début	vide	date de début	Une date valide plus ancienne ou égale à la date de début
Couleur	inconnue	vide	inconnue	Selon la liste de départ lue dans les permis ou mise à jour
Carac. Sup.	aucune	aucune	aucune	Selon les choix offerts, mutuellement indépendants. Le choix « Dangereux » sera en <b>rouge et gras</b> , s'il est sélectionné

**N.B.** Lors de la lecture du fichier CSV, un enregistrement est rejeté si au moins une de ses valeurs est erronée ou est en exception avec le tableau ci-haut.

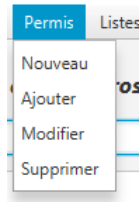
- À part les notifications par boîte de dialogue demandées, je vous laisse prendre vos propres décisions pour tous les autres cas que vous jugez importants d'être notifiés. Ce qui est important, c'est de rester convivial avec votre utilisateur (il ne faut pas harceler l'utilisateur) et uniforme dans votre présentation.

- Ce lien vous permettra de trouver des icônes pour agrémenter vos boutons  
« <http://www.iconarchive.com/> ».

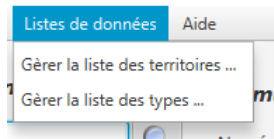
### Les menus :




- « Exporter XML ... » permet de sauvegarder tous les permis contenus dans la BD dans un fichier XML selon un format prédéterminé.
- « Importer CSV ... » permet de lire un fichier CSV, du même format que celui utilisé dans les formats 1 et 3.
- Même action que le bouton « Quitter ».



- Les 4 actions de ce menu ont le même comportement et font les mêmes actions que les boutons de l'interface.



- Les 2 actions de ce menu ont le même comportement et font les mêmes actions que les boutons «  » de l'interface qui sont adjacents aux listes de territoires et de types.



- Même action que le bouton « ? », de l'interface graphique.

### Les données utiles au système :

- Les données du système (données utiles) seront sauvegardées en temps réel dans une base de données (BD). En référence à vos apprentissages passés dans le domaine des bases de données, je vous laisse choisir le SGBD et les technologies avec lesquelles vous êtes à l'aise. Les points importants à respecter sont :
  - Que les données soient mises à jour en temps réel.
  - Que toutes les données utiles du système soient conservées dans cette BD.
  - Que le modèle soit en relation avec un tiers (ORM) pour faire le lien avec la BD.
  - Que le code continu à respecter l'architecture MVC, il doit être bien factorisé et doit aussi utiliser le mieux possible les interfaces pour diminuer les dépendances fonctionnelles avec les autres classes du modèle.
  - Que dans le code, il soit facile de s'y retrouver grâce à des commentaires judicieux et complets.
- Lors de la démonstration, des preuves de concepts et des explications vous seront demandées.

### XML en Java et le format du fichier XML :

- Les règles d'écriture pour le XML sont très spécifiques et simples, l'essentiel de ces règles est présenté dans le document « Spécifications XML.pdf » disponible dans le dossier de ce Tp.
- Pour vous permettre d'écrire ou de lire un document XML en java, je vous réfère à un document sur le web <https://cynoher.developpez.com/tutoriel/java/xml/jdom/>. Comme indiqué dans cette référence vous aurez besoin de JDOM une librairie que vous allez intégrer à votre code Java. Voici le lien pour télécharger cette dernière <http://www.jdom.org/index.html>. Téléchargez la version 2.0.6.

- Je propose ici, un format de fichier XML :

```
<?xml version="1.0" encoding="UTF-8"?>

<animaux>
  <animal type="chat" sexe="mâle">
    <permis numero="66666">
      <date_debut jour="26" mois="08" année="1973"></date_debut>
      <date_fin jour="26" mois="08" année="1974"></date_fin>
    </permis>
    <territoire>Sainte-Rose</territoire>
    <nom>pompon</nom>
    <date_naissance jour="15" mois="06" année="1973"></date_naissance>
    <couleur>tabby gris</couleur>
    <poids metrique="kg">2.1</poids>
    <caracteristiques>
      <vaccine>1</vaccine>
      <sterilise>1</sterilise>
      <micropuce>0</micropuce>
      <dangereux>0</dangereux>
    </caracteristiques>
  </animal>
  <animal type="chien" sexe="femelle">
    <permis numero="88888">
      <date_debut jour="13" mois="07" année="1983"></date_debut>
      <date_fin jour="26" mois="08" année="1984"></date_fin>
    </permis>
    <territoire>Sainte-Rose</territoire>
    <nom>pompon</nom>
    <date_naissance jour="26" mois="05" année="1983"></date_naissance>
    <couleur>brun chocolat</couleur>
    <poids metrique="kg">6.3</poids>
    <caracteristiques>
      <vaccine>1</vaccine>
      <sterilise>1</sterilise>
      <micropuce>0</micropuce>
      <dangereux>1</dangereux>
    </caracteristiques>
  </animal>
  .
  .
  .
</animaux>
```