

## 1) But

Le but de ce travail est de mettre en pratique les concepts d'interface graphique (AWT et Swing). En outre, l'étudiant aura à mettre en œuvre les notions de conteneurs, de composants, de gestionnaires de disposition et d'écouteurs vus en classe.

## 2) Travail demandé

Alpha vient d'ouvrir une clinique vétérinaire d'urgence sans rendez-vous 24h / 24h pour prendre soins de nos animaux de compagnie (félin, canin, et autres). Le nom de la nouvelle clinique est **Soins Pour Tous** (SPT). Ayant appris que vous avez développé un ensemble de services pour gérer une liste d'éléments selon leur priorité, Alpha s'adresse à vous pour utiliser quelques services à travers des interfaces graphiques pour gérer la liste de ses patients reçus en urgence. Ci-dessous la description du travail à faire.

### 2.1. Les classes à compléter et la séparation des paquetages

Vous devez compléter les classes `GestionUrgenceCliniqueVeterinaire` et `Patient` disponibles dans Moodle et ajouter toutes les classes / interfaces pour construire les interfaces graphiques. Vous pouvez ajouter d'autres méthodes dans ces classes si vous jugez nécessaire.

Pour une bonne séparation entre les différentes interfaces et classes, vous devez créer les paquetages suivants :

- ✓ `ca.uqam.inf2120.tp1.adt` : contient les interfaces `FileAttenteTda` et `OrdonnableParPrioriteEtDateHeure`.
- ✓ `ca.uqam.inf2120.tp1.adt.impl` : contient la classe `FileAttenteImpl`.
- ✓ `ca.uqam.inf2120.tp3.modele` : contiendra les classes `GestionUrgenceCliniqueVeterinaire` avec un attribut `listePatients` de type `FileAttenteTda` développé dans le travail pratique 1 et `Patient` qui implémente l'interface `OrdonnableParPrioriteEtDateHeure`.
- ✓ `ca.uqam.inf2120.tp3.interfacegraphiques` : contiendra toutes les classes développées pour l'implantation des interfaces graphiques et la gestion des événements (les fenêtres et les écouteurs).

Aucune importation des interfaces ou des classes du paquetage `ca.uqam.inf2120.tp3.interfacegraphiques` n'est permise dans les trois (3) autres paquetages.

La figure 1 vous montre la séparation des paquetages avec les différentes interfaces / classes.

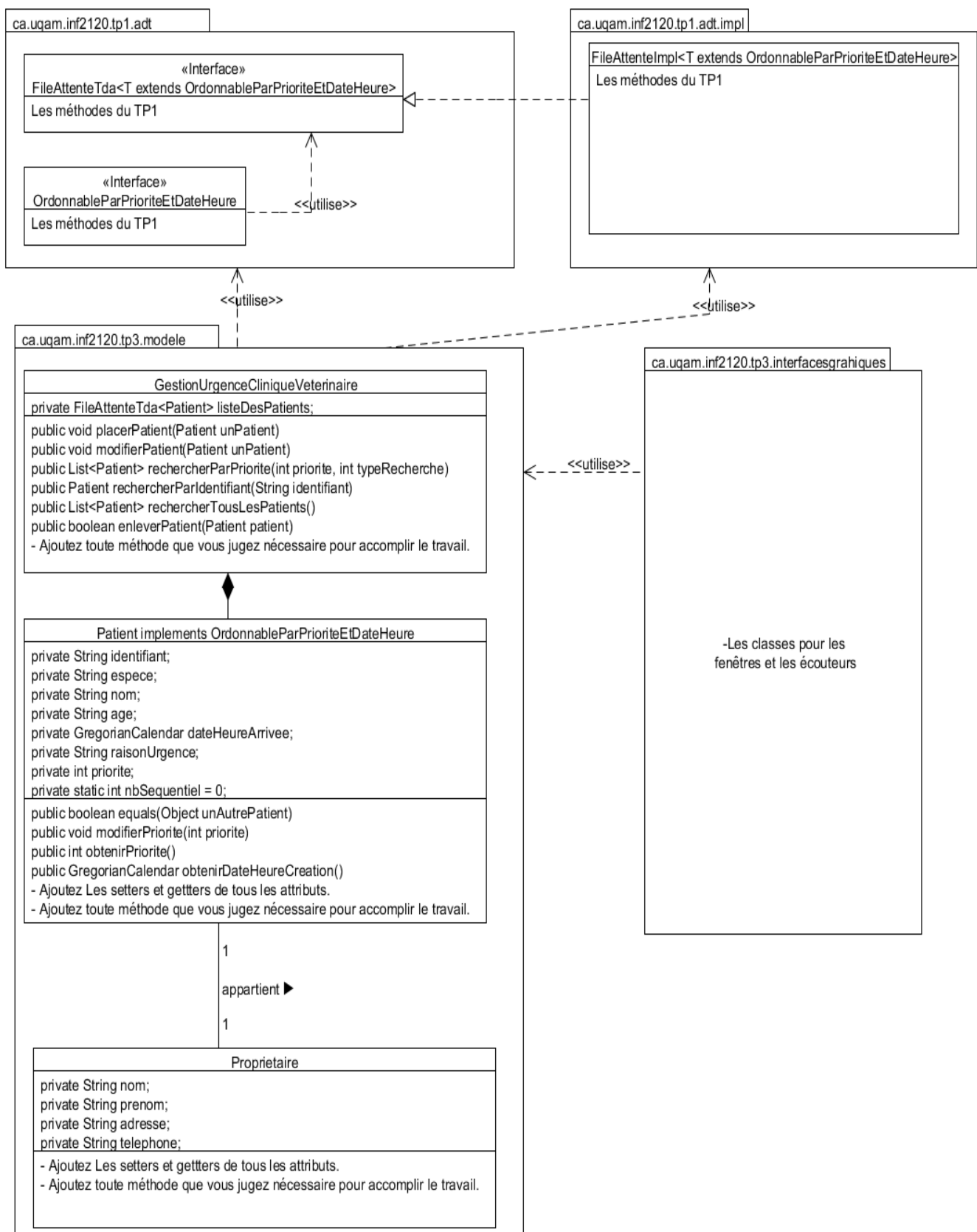


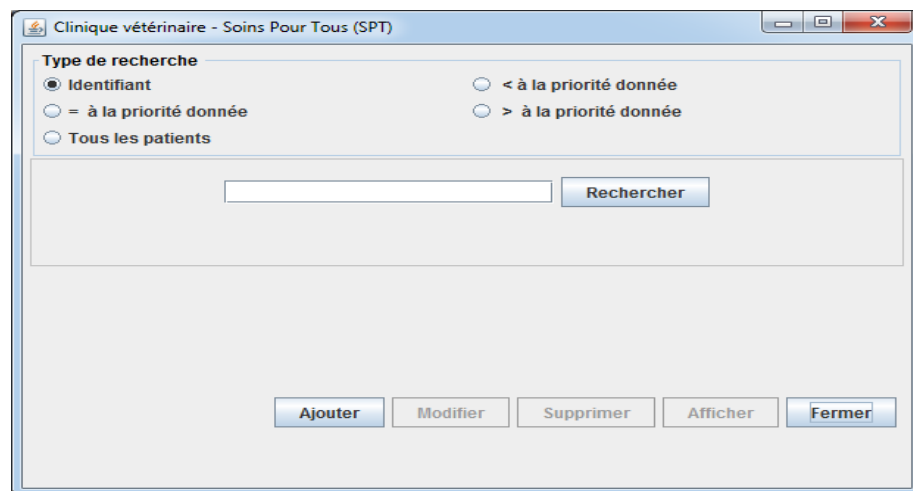
Figure 1 : Diagramme de paquetages (Interfaces / Classes)

## 2.1. Les interfaces graphiques de SPT

Vous devez développer les interfaces graphiques qui permettront :

- ✓ de rechercher un ou plusieurs patients selon un critère donné
- ✓ d'ajouter un patient ;
- ✓ de modifier les informations d'un patient ;
- ✓ de supprimer un patient ;
- ✓ d'afficher les informations d'un patient ;

### a) Interface de recherche



**Figure 2 : Interface de recherche d'un ou plusieurs patients**

Cette interface est le point d'entrée dans l'application et permet de rechercher un ou plusieurs patients selon les critères suivants :

- ✓ l'identifiant du patient ;
- ✓ selon la priorité (égale à, inférieure à ou supérieure à);
- ✓ tous les patients.

En plus de ces critères, l'utilisateur doit entrer l'information à saisir selon le type de recherche choisi. Par défaut, le type de recherche est "Identifiant". Lorsque le type de recherche est "Tous les patients", le champ de saisie de la recherche doit être non éditable.

### b) Interface de recherche après une recherche fructueuse

Après une recherche fructueuse, l'identifiant du patient, le nom du patient, la date et l'heure d'arrivée, le nom et le prénom du propriétaire doivent être affichés. Par défaut, la première ligne de la table doit être sélectionnée. Les patients doivent être affichés selon l'ordre de grandeur de la

priorité (de la plus grande priorité à la plus petite priorité). La table des patients (JTable) ne doit pas être éditable.

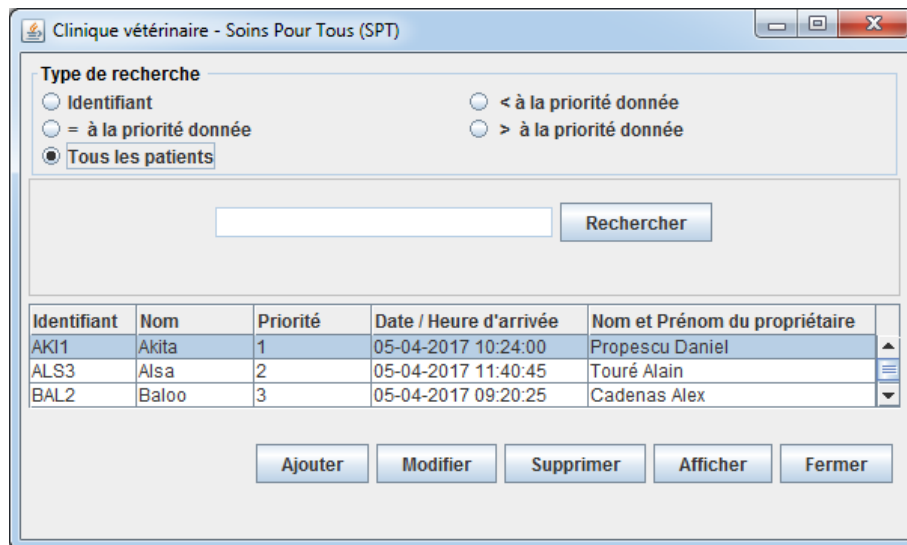


Figure 3 : Interface de recherche de patients après une recherche fructueuse

### c) Interface de recherche après une recherche infructueuse

Après une recherche infructueuse selon un type de recherche, un message d'information doit être affiché à travers une boîte de dialogue modale (JOptionPane).

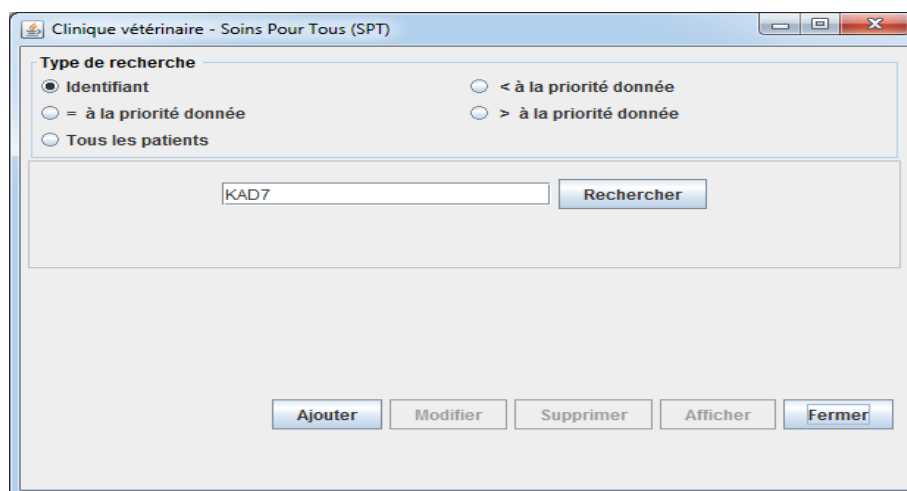


Figure 4 : Recherche infructueuse d'un patient avec l'identifiant "KAD7"

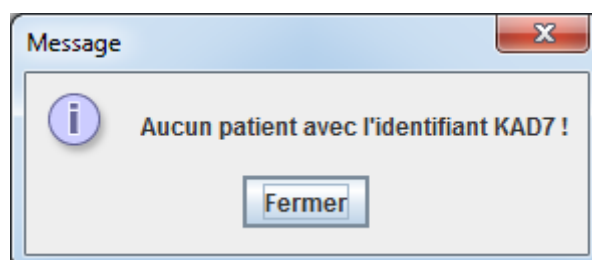


Figure 5 : Message d'information après une recherche infructueuse d'un patient

#### d) Explication des fonctions des boutons sur l'interface de recherche

Les boutons *Modifier*, *Supprimer*, *Afficher* doivent être désactivés tant que la table des patients est vide. Si la table des patients contient au moins un élément, tous les boutons doivent être activés. Les fenêtres d'ajout, de modification, de suppression, et d'affichage sont des boîtes de dialogue de type `JDialog` et elles doivent être modales.

**Ajouter** : permet d'ajouter un nouveau patient dans la table des patients en spécifiant toutes les informations requises. Toutes les informations sont obligatoires lors de l'ajout. Un message d'information doit être affiché à travers une boîte de dialogue modale (`JOptionPane`) si l'utilisateur clique sur le bouton "Ajouter" alors qu'un des champs de saisie est vide ou il ne contient que des espaces. La liste déroulante "Priorité" doit être non éditable et les valeurs disponibles sont : 1, 2, 3, 4, 5.

Les attributs "Identifiant" et "Date / Heure d'arrivée" ne sont pas visibles lors de l'ajout d'un patient, car ils sont générés automatiquement par le système (Voir la classe `Patient` pour les règles de construction de l'identifiant et de l'obtention de la Date / Heure d'arrivée).

Tout nouveau type de patient ajouté doit être visible dans la table des patients s'il répond aux critères de recherche précédemment définis. Voir la figure 6 pour plus de détails sur l'interface d'ajout.

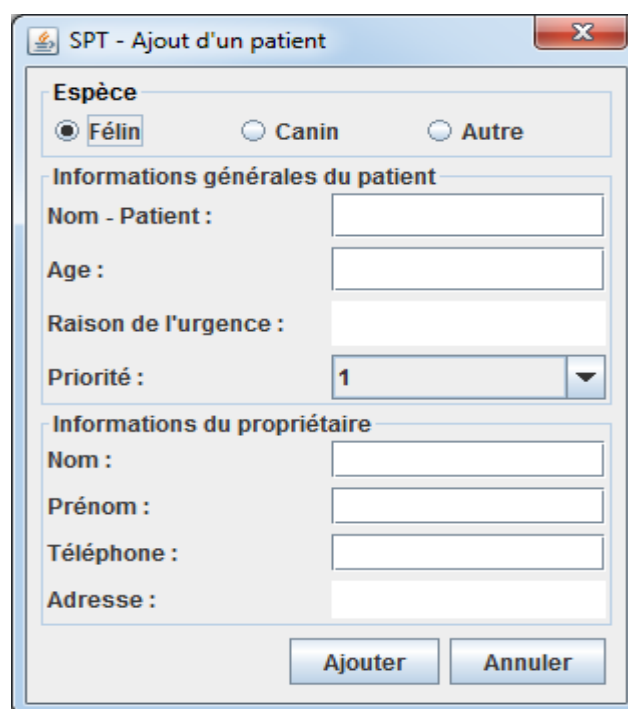
The image shows a Java Swing dialog box titled "SPT - Ajout d'un patient". It has a standard window frame with a title bar, a maximize button, and a close button. The dialog is divided into several sections. At the top, under the heading "Espèce", there are three radio buttons: "Félin" (which is selected), "Canin", and "Autre". Below this is a section titled "Informations générales du patient" containing four text input fields: "Nom - Patient :", "Age :", "Raison de l'urgence :", and "Priorité :". The "Priorité" field is a dropdown menu currently showing the value "1". Below the general information is a section titled "Informations du propriétaire" containing four text input fields: "Nom :", "Prénom :", "Téléphone :", and "Adresse :". At the bottom of the dialog, there are two buttons: "Ajouter" and "Annuler".

Figure 6 : Interface d'ajout d'un type de patient

**Modifier** : permet de modifier les informations d'un patient sélectionné dans la table des patients. Lors de la modification, toutes les valeurs du patient sélectionné peuvent être modifiées

sauf l'identifiant, le nom du patient et la date / heure d'arrivée. Ces champs sont non éditables. Voir la figure 7 pour plus de détails sur l'interface de modification.

Figure 7: Interface de modification d'un patient

**Afficher** : permet d'afficher les informations d'un patient sélectionné dans la table des patients. Tous les champs doivent être non éditables (ou non modifiables).

Figure 8 : Interface d'affichage d'un patient

**Supprimer** : permet de supprimer le patient sélectionné dans la table des patients.

**Fermer** : permet de fermer la fenêtre de recherche.

### 3) Livrables et date de remise

#### 3.1. Livrables

Via Moodle, vous devez seulement une copie électronique compressée (.zip ou .rar) qui comprend :

- a) Le dossier du projet composé des quatre (4) paquetages mentionnés précédemment avec toutes les classes/interfaces développées.
- b) Une page de présentation (en Word) comprenant votre nom, votre prénom, le sigle du cours, le groupe du cours, le nom de l'enseignant.

#### 3.2. Date de remise

Vous devez remettre votre travail le mercredi, 26 avril 2017 avant 23H55.

Pour tout travail remis en retard, les pénalités de seront appliquées selon la formule suivante :

**Nombre de points de pénalité =  $m / 144$** , où m est le nombre de minutes de retard par rapport à l'heure de remise. Aucun travail ne sera accepté après cinq (5) jours de retard.

### 4) Pondération

- a) **Code source (40%)**: Clarté du code, indentation, commentaires, choix des identificateurs, respect des spécifications de l'interface, optimisation du code, l'absence de redondance dans le code et bonne utilisation de la généricité.
- b) **Exécution (60%)**: Convivialité des interfaces graphiques, présentation et exactitude des résultats.

**Notez-bien :** La note 0 sera attribuée à tout programme qui ne compile pas.

### 5) Règles de programmation

- a) Les identifiants doivent être aussi significatifs que possible et se conformer aux conventions standards en ce qui concerne l'emploi des majuscules et minuscules.
- b) Ne déclarez pas de variables que vous n'utilisez pas.
- c) N'utilisez pas une même variable pour des usages différents.
- d) Ne faites pas de copier-coller des commentaires Javadoc placés au début des méthodes de l'interface dans la classe qui l'implémente.
- e) N'employez pas de variables "globales" dans le programme client. Si une méthode nécessite des valeurs, passez ces valeurs en tant que paramètres. Si une méthode produit ou modifie des valeurs, elles doivent être retournées en tant que résultats ou paramètres.
- f) Une bonne structure est très importante. Décomposez vos algorithmes ou méthodes en plusieurs algorithmes/méthodes de manière significative en toute occasion où une telle décomposition

améliore la clarté. Si nécessaire, n'hésitez pas d'ajouter des méthodes privées pour éviter d'avoir des méthodes publiques dont le nombre de lignes est supérieur à 25.

- g) Votre programme doit être indenté de façon systématique et les blocs d'instructions doivent être bien commentés et séparés par une ligne blanche pour permettre une bonne aération.

**Le travail est strictement individuel. Le règlement sur le plagiat sera appliqué sans exception. Vous devez ainsi vous assurer de ne pas échanger du code avec des collègues, ni de laisser sans surveillance votre travail au laboratoire. Vous devez également récupérer sans faute toutes vos impressions de programme au laboratoire.**