



UNIVERSITÉ
DE MONTPELLIER

Projet Java ToDoList

Par
FERAUD Fabien
NOVELLON Florian

HLIN505

Année Universitaire : 2016/2017

Diagramme de classe :

La classe **ListeTaches** contient une instance de **ListeTachesSerializable** contenant les listes de **Tache** de l'application, elle peut être aussi enregistrée dans un fichier pour pouvoir être récupérée plus tard. L'attribut *cursor* est incrémenté à chaque fois que l'on ajoute une tâche et devient ainsi l'*ID* de la tâche.

Le même concept a été fait entre **Categories** et **CategoriesSerializable**.

L'objet **Tache** contient un *ID* qui est unique et qui permet l'identification d'une tâche dans la liste. L'*idCategories* permet l'identification de la catégorie dans la classe **Categories**, ainsi si l'id est introuvable, la catégorie de la tâche est "Sans catégorie".

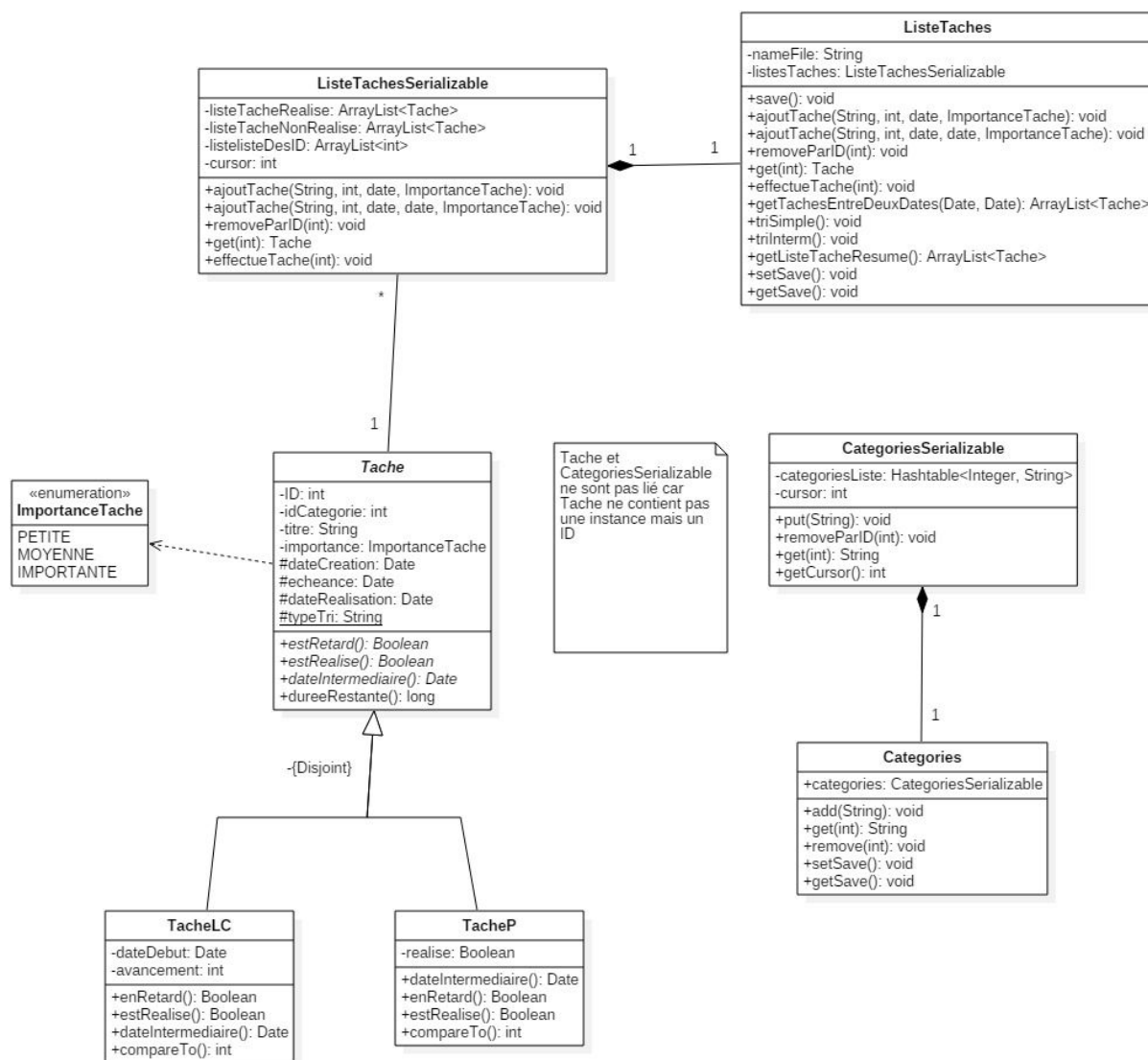
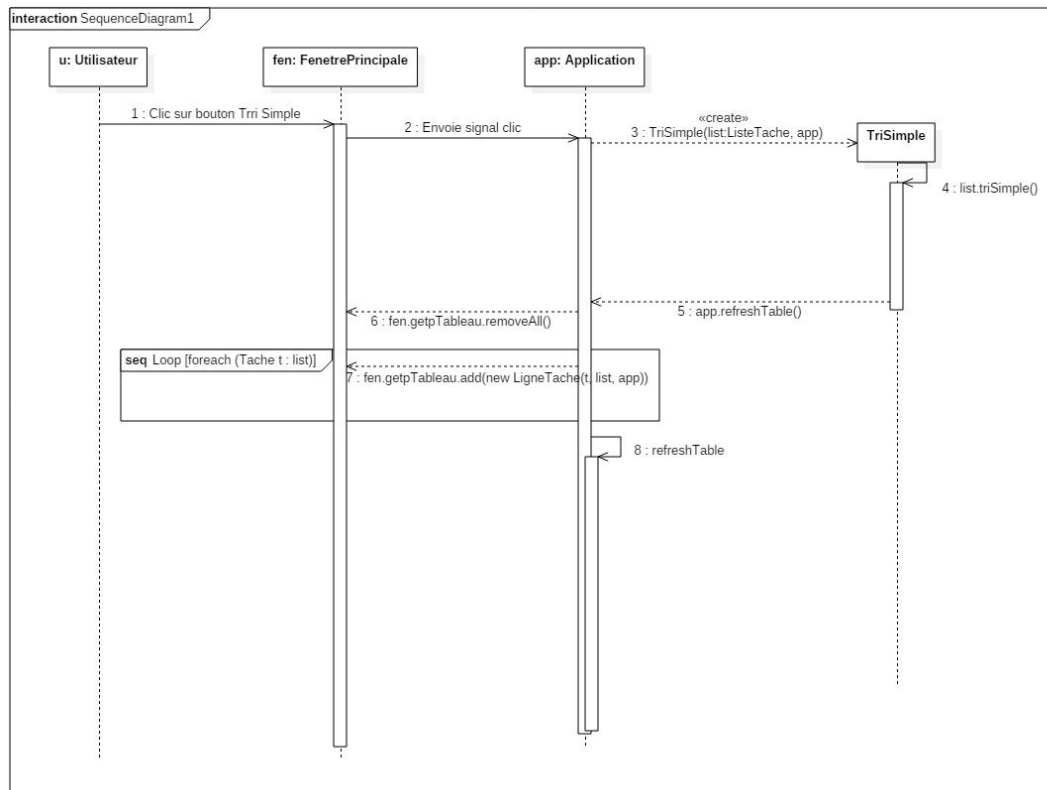


Diagramme de séquence :

Lorsqu'un utilisateur clique le bouton "Tri Simple", celui-ci positionné dans FenetrePrincipale envoie un signal à l'écouteur d'événement dans l'Application. App instancie l'objet TriSimple qui prend en paramètre la liste des tâches et l'application. Ensuite la liste est triée et l'application rafraîchit l'affichage du tableau de tâches.



Machine à états :

Lorsqu'une tâche est créée on vérifie :

-Pour une tâche ponctuelle, si sa date d'échéance est bien supérieure à la date du jour.

-Pour une tâche long cours, si sa date d'échéance est bien supérieure à la date du début de tâche.

Elle entre ensuite dans l'état "En cours"

Une tâche passe dans l'état "Terminé" quand:

-Lorsqu'on clique sur le bouton "Terminer" d'une tâche ponctuelle

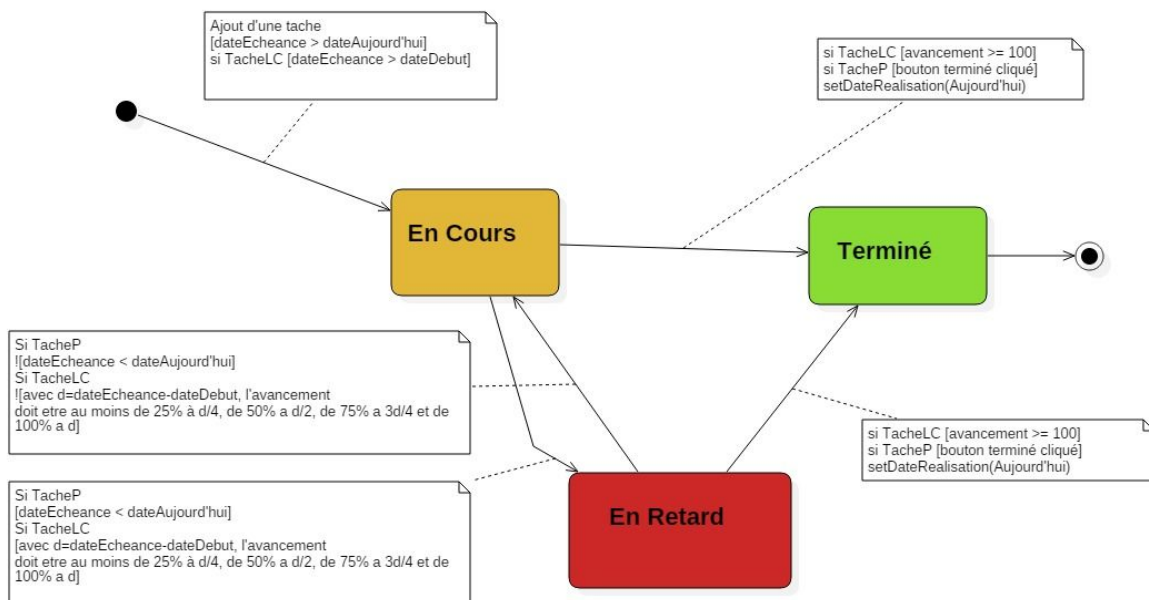
-Lorsque l'avancement d'une tâche est supérieur à 100

Ensuite on passe la date de réalisation de la tâche de null à la date du jour.

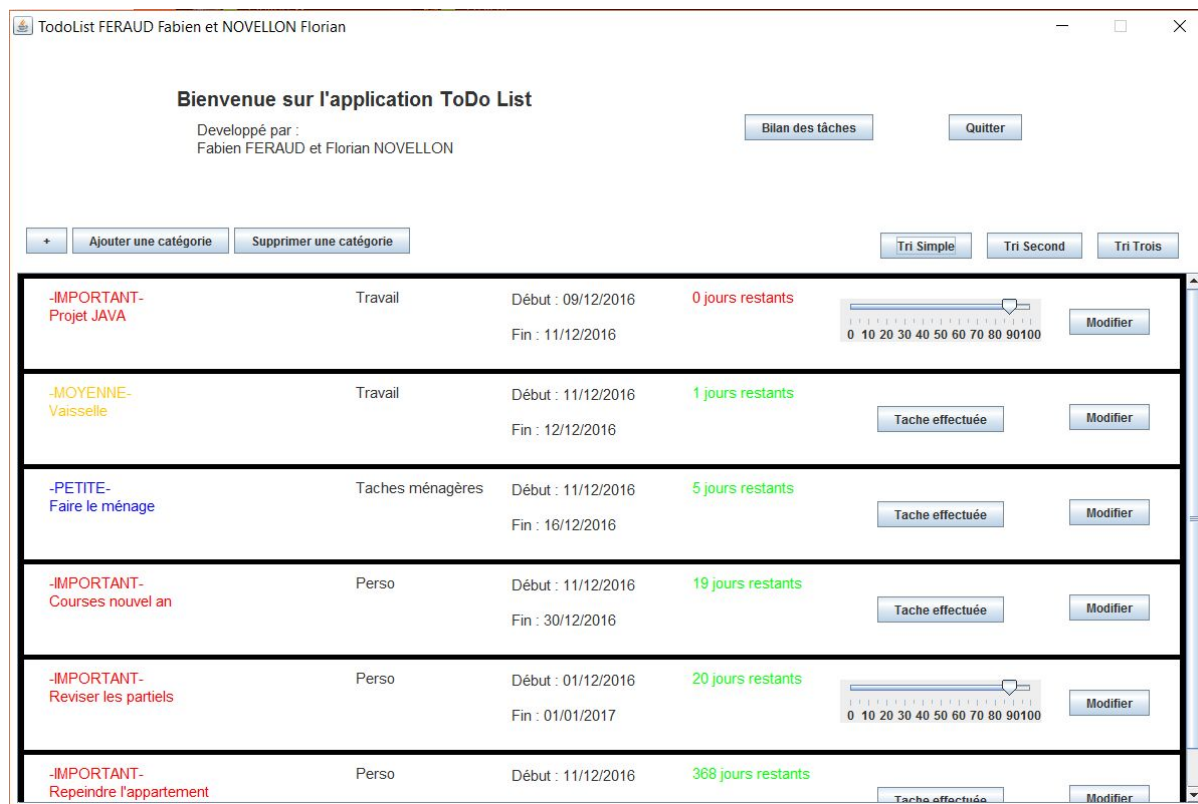
Une tâche passe dans l'état "En Retard" quand:

-Pour une tâche ponctuelle, si la date d'échéance est inférieure à la date du jour

-Pour une tâche long cours, avec $d = \text{date d'échéance} - \text{date début}$, l'avancement doit être au moins de 25% à $d/4$, de 50% à $d/2$, de 75% à $3d/4$ et de 100% à d .



Manuel d'Utilisation



En lançant l'application, on arrive sur la fenêtre d'accueil qui est la partie centrale de notre application. En effet la partie inférieure est réservée à l'affichage des tâches qui ne sont pas encore réalisées et la partie supérieure nous donne accès à différentes actions.

Commençons par la partie supérieure, elle se compose :

- un bouton "+" nous donnant accès à un formulaire pour la création de tâches
- Un bouton "Ajouter catégorie" pour accéder à une fenêtre permettant l'ajout de catégorie à notre application
- Un bouton "Supprimer catégorie" pour gérer la suppression de catégories de la même manière que pour le bouton précédent
- Un bouton "Tri simple" pour trier notre liste des tâches en fonction de la proximité de date d'échéance pour les tâches qui ne sont pas en retard et par ordre d'urgence pour les tâches en retard
- Un bouton "Tri interm" pour trier notre liste de tâche de façon à prendre en considération les échéances intermédiaires pour une tâche long cours et ainsi trier tous les tâches en fonction de l'échéance de celle-ci sachant que l'échéance intermédiaire d'une tâche ponctuelle est sa date d'échéance.
- Un bouton "Tri résumé" pour afficher que 8 tâches contenant une tâche importante, 3 moyennes et 5 petites
- Un bouton "Bilan des tâches" pour accéder à une fenêtre où l'on pourra avoir un résumé des tâches sur une période donnée.

La partie inférieure représente un tableau de tâche où chaque ligne est une tâche. On pourra voir dans l'ordre de gauche à droite :

- le nom de la tâche avec son importance et un code couleur selon l'importance (rouge pour important, jaune pour moyen et bleu pour petite)
- la catégorie
- la date de début et de fin
- le nombre de jours restants avant d'arriver à l'échéance (affichage en rouge si retard sur la tâche sinon vert)
- Un bouton pour terminer la tâche si c'est une tâche ponctuelle ou un slider représentant l'avancement de la tâche. Pour terminer une tâche long cours il suffit de mettre le slider à 100 signifiant que la tâche a été effectuée à 100%. Attention on ne peut pas revenir en arrière sur le slider.
- Un bouton "Modifier" permettant de modifier une tâche.

Passons aux différentes fenêtres évoquées plus haut.

La fenêtre création de tâche :

On doit donc choisir si c'est une tâche ponctuelle ou à long cours en cochant une des deux cases en haut de la fenêtre. On rentre alors le nom de la tâche, sa catégorie. Si c'est une tâche ponctuelle on laisse le champ "Date de début" vide sinon on rentre la date de début de la tâche. Enfin on rentre une date d'échéance de la tâche qui doit être supérieure ou égale au jour de création pour une tâche ponctuelle et supérieure ou égale à la date de début pour une tâche long cours.

En appuyant sur "Valider" ça crée une nouvelle tâche si tout est correct sinon le formulaire ne sera pas envoyé.

La fenêtre d'ajout de catégorie et de suppression de catégorie :

Création de tâche
Entrez le nom de la nouvelle catégorie et appuyez sur "Ajouter"

Votre nouvelle catégorie

Nouvelle catégorie

Retour Ajouter

Suppression d'une catégorie
Choisissez la catégorie que vous voulez supprimer et appuyez sur "Supprimer"

Catégorie:

Retour Supprimer

Sur la fenêtre de gauche on entre le nom de la nouvelle catégorie que l'on veut avoir et on appuie sur "Ajouter" pour envoyer le formulaire. Sur la fenêtre de droite on choisit la catégorie que l'on veut supprimer et on appuie sur "Supprimer".

Enfin la fenêtre de bilan des tâches :

Bilan sur une période
Développé par : Fabien FERAUD et Florian NOVELLON

Retour

Date de début et date de fin de la période de bilan: Voir

Tâches réalisés : 0% Tâches réalisés en retard : 0% Tâches en cours : 100%

Projet JAVA	Travail	Début : 09/12/2016 Fin : 11/12/2016	0 jours restants	<input type="text" value="0 10 20 30 40 50 60 70 80 90 100"/>
Vaisselle	Travail	Début : 11/12/2016 Fin : 12/12/2016	1 jours restants	
Faire le ménage	Tâches ménagères	Début : 11/12/2016 Fin : 16/12/2016	5 jours restants	
Courses nouvel an	Perso	Début : 11/12/2016 Fin : 30/12/2016	19 jours restants	
Reviser les partiels	Perso	Début : 01/12/2016 Fin : 01/01/2017	20 jours restants	<input type="text" value="0 10 20 30 40 50 60 70 80 90 100"/>

Grâce au formulaire intégré à la page on rentre la date de début et de fin de la période que l'on veut voir, la date de début devant être inférieure à la date de fin. En appuyant sur le bouton "Voir" on verra alors le pourcentage de tâche réalisé ayant leur échéance dans la période donnée. Il y aura aussi le pourcentage de tâche réalisée en retard sur la période et le pourcentage de tâche en cours sur cette période.

En dessous on pourra voir les tâches en cours sur la période. Pour les tâches long cours l'avancement grâce au slider sera aussi affiché.

La dernière fenêtre est la fenêtre de modification de tâche :

Cette fenêtre est accessible en appuyant sur le bouton “Modifier” d’une tâche. Les champs du formulaire sont pré-rempli par les valeurs actuelles de la tâche. On peut alors modifier les champs, appuyer sur “Valider” et la tâche aura alors été modifiée dans la fenêtre principale.

Complément du cahier des charges :

Pour spécifier un peu plus le cahier des charges nous allons maintenant voir les choix que nous avons fait pour le développement de l’application.

Le cahier des charges nous laissait libre de la représentation de l’application en nous imposant juste de faire un interface graphique avec Java Swing. Nous avons donc décidé de représenter les tâches sous la forme d’un tableau recréé à notre manière car la contrainte d’affichage d’objets différents dans une JTable ne rendait pas la tâche aisée ainsi nous avons créé notre classe qui hérite de l’objet JPanel et nous avons agencé le JPanel pour créer des lignes personnalisées.

Pour gérer les dates et pouvoir faire les calculs de jours restants, les dates d’échéances sont mises à 23h59 et les dates de création ou de début selon la tâche sont mises à 00h01.

Nous avons décidé de gérer l’accomplissement de la tâche différemment selon le type de celle-ci. En effet pour les tâches ponctuelles nous avons placé un bouton dans notre ligne du tableau qui permet à l’utilisateur de dire si sa tâche est terminée en appuyant dessus. Pour les tâches long cours nous avons mis un slider grâce auquel l’utilisateur peut spécifier l’avancement dans la tâche. Quand le slider arrive à 100 cela signifie que la tâche est terminée et qu’ainsi nous pouvons la retirer du tableau.

Pour le fait qu’une tâche ponctuelle ait pour date de début la date du jour de création alors que pour la tâche long cours elle doit être spécifiée, nous avons décidé que dans le formulaire il suffirait de ne pas remplir le champ date de début si l’utilisateur coche la case tâche ponctuelle. Un message permet à l’utilisateur de l’avertir de cela et en interne quelque soit la date dans le champ date de début, nous attribuons à la date de création de la tâche la date du jour. Si par contre l’utilisateur coche la case de tâche long cours, le champ date de début sera requis. Dans le cas des deux tâches, la vérification sur l’ordre de la date sera faite et le formulaire ne sera pas envoyé tant que le problème de date n’est pas réglé.

Pour le bilan, nous avons décidé de ne pas afficher le même tableau que dans la fenêtre d’accueil. Il n’y a de figurer que le nom de la tâche, sa catégorie, ses dates, le nombre de jours restants et pour les tâches long cours le slider d’avancement ne pouvant être modifié. De plus dans le bilan nous n’affichons pas l’importance de la tâche avec son nom comme dans la fenêtre d’accueil. Dans le bilan on ne s’intéresse pas à connaître le degré d’importance d’une tâche mais à savoir combien de tâche en cours nous avons sur la période donnée grâce aux pourcentages calculés. Il nous semblait donc pas utile de l’afficher.

Pour la gestion du Modèle-Vue-Contrôleur (MVC) nous avons donc fait 3 packages pour chacune des parties. Nous avons donc toutes les classes qui utilisent Java Swing dans le package vue, toutes les classes qui servent à stocker les données et interagir avec elles se trouvent dans le modèle et enfin toutes les classes qui permettent de lier le modèle à la vue se trouvent dans le package du contrôleur. On y trouve les classes qui héritent de ActionListener et permettent de faire des actions lors d'un événement dans la vue. Ces actions sont des appels de méthodes du modèle, c'est pourquoi le contrôleur fait la liaison entre la vue et le contrôleur

Enfin, nous avons décidé d'utiliser la sérialisation de Java pour la persistance des données