Cours 420-243-LI **Développement informatique** **Hiver 2024** **Cégep Limoilou** **Département d'informatique** **Professeur : Walid Boulabiar**	**Travail pratique 1** **développement d'un modèle avec VP et synchronisation avec le code** **12% de la session**

Objectifs

- Utiliser un outil de conception UML
- Concevoir un modèle de classe
- Synchroniser code et modèle du code

À remettre :

- Le travail sera remis sur Léa à la date indiquée.

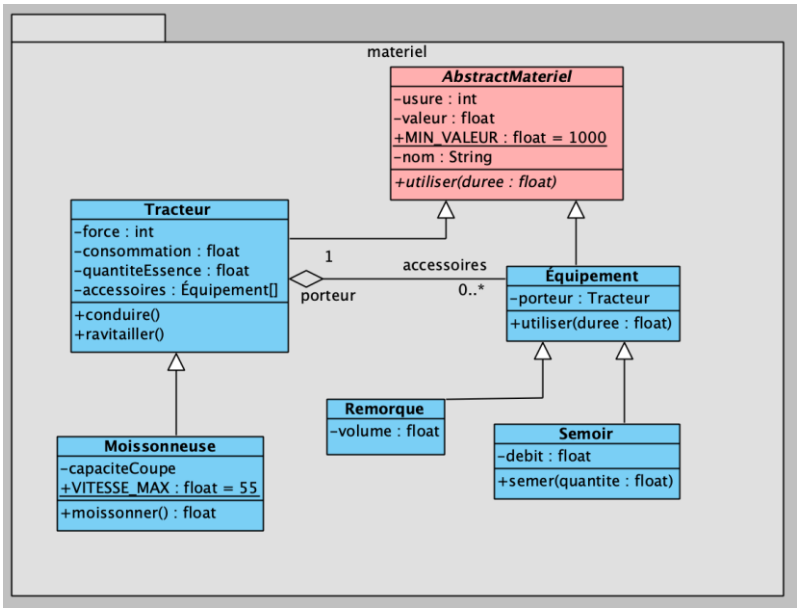
Contexte :

- **Le travail se fait seul ou en équipe de 2 personnes.**
- **On veut produire le modèle d'une ferme. On va procéder en 5 parties :**
  - Refaire la partie qui vous est donnée en image (le matériel de la ferme) 25%
  - Faire la partie où l'on doit produire une hiérarchie avec les classes déjà fournies. (les produits de la ferme) 25%
  - Faire le modèle qui correspond à l'énoncé texte (l'ensemble de la ferme) 15%
  - Imaginez une partie du modèle par vous-même en réfléchissant au problème sans avoir d'énoncé (personnel de la ferme) 20%
  - Faire générer le code Java 15%

À faire

- Utilisez le fichier «*TP1 vos initiales.vpp*» qu'on vous a fourni dans le dossier «*tp1 depart*». Renommer le fichier en changeant vos initiales par vos véritables initiales. Dans ce modèle, vous trouverez un diagramme de classe nommé «*tp1-class diagram2*».

- L'image suivante vous montre la partie du diagramme qui touche le matériel agricole.



- Dans le package produits (ce qui est vendu par la ferme) vous trouverez plusieurs produits ayant des attributs et des méthodes en communs. On vous demande d'ajouter les classes parentes manquantes pour réutiliser au maximum toutes les méthodes et tous les attributs. On vous demande également de faire en sorte que toutes les méthodes avec la même signature puissent être appelées à partir d'un même parent (polymorphisme).
- Attention au *Boeuf* et aux *Patate*. Sur la ferme, ces éléments peuvent être autant achetés que vendus.
- On va maintenant s'occuper de la vue d'ensemble de la ferme. Vous devrez produire un modèle qui respecte les énoncés suivants :
    - La ferme tourne autour de 2 types de *transactions* : les *achats* et les *ventes*.
        - Les achats concernent principalement l'acquisition de Matériel (tracteur, remorque, Moissonneuse…) et de semences pour les champs.
            - Lorsqu'on achète des semences, on s'intéresse surtout au type de graines ainsi qu'à la quantité. Sur un même achat, on peut faire plusieurs acquisitions autant des semences que du matériel.
            - Vous devez connecter votre partie qui s'intéresse au matériel, avec la ferme.
        - Noter que deux des produits de ventes peuvent également être achetés. Il s'agit de Bœuf et Patate. En effet, l'élevage de bovin doit être renouvelé par des achats et les patates sont elles-mêmes leur propre semence.
        - Une vente permet de vendre des produits de la ferme. Vous avez déjà créé la hiérarchie de ses produits dans une étape précédente. Il vous suffit maintenant de les connecter à la ferme.
    - La ferme retient toutes les transactions qui sont effectuées.
    - La ferme possède également un inventaire du matériel et des produits qu'elle vend.

- On vous demande ***d'imaginer*** et de concevoir vous-même la partie qui montre le personnel impliqué sur cette ferme.
    - Créez d'abord un package *personnel*.
    - Concevez votre modèle du personnel de la ferme et essayez de préciser les relations entre ceux-ci le plus clairement possible.
    - Mettre au moins un attribut et au moins une méthode sur chaque classe
    - Connectez le personnel au reste du diagramme.
- Lorsque vous serez satisfait de votre design, faites générer le code qui correspond à votre modèle dans un projet IntelliJ que vous aurez créé. Assurez-vous de bien arranger le code et le modèle pour qu'il compile.
- On doit également faire un peu de mise en forme générale sur notre diagramme
    - Les classes abstraites doivent être de couleur *Pink*
    - Les interfaces doivent être de couleur Orange
    - La classe principale doit être d'un bleu plus pâle
    - Assurez-vous que les multiplicités sont bien placées et visibles
    - Assurez-vous que les associations sont suffisamment longues pour qu'on y voie tout
    - Placez autant que possible les parents au-dessus de leurs enfants

Critères d'évaluation

1. Les consignes sont respectées et le travail est complet.
2. Les diagrammes et le code sont propres et faciles à consulter. (éviter de croiser inutilement les lignes d'association, éviter les superpositions inutiles d'éléments)
3. Le code final compile et il est clair (ne pas s'occuper des Javadoc)
4. Le code final reflète au modèle
5. Les bons types d'associations ont été utilisés (composition, agrégation, héritage et implémentation)
6. Les associations sont associées aux attributs qui les concernent (important)
7. Toutes les multiplicités importantes sont présentes
8. Vous utilisez les classes abstraites adéquatement
9. Le modèle est simple et pertinent.
10. Tous les paramètres et tous les attributs ont des types adéquats.
11. Tous les noms de classes et d'attributs respectent les standards Java.
12. Les consignes de mise en forme sont respectées

Dépôt

IL faut déposer sur LEA un zip contenant votre projet vpp, votre projet java et un fichier Word avec les nom des coéquipiers si vous travailler en équipe de 2.