Conception UML – Mise en œuvre – Jeux Olympiques

Description et objectifs

A l'occasion des futurs JO de Paris 2024, votre société a été sélectionnée pour réaliser une application permettant de naviguer aisément dans les résultats de toutes les épreuves sportives depuis la création des Jeux Olympiques modernes.

A ce stade vous ne disposez pas encore du cahier des charges avec l'expression de besoin du client mais vous pouvez d'ores-et-déjà vous occupez d'intégrer les données des résultats JO. Vous disposez à cet effet d'un fichier CSV contenant l'ensemble de ces résultats (271 000 lignes).

Vous allez devoir concevoir cette application en commençant par :

- Identifier les entités métier
- Réaliser le diagramme de classes,
- Réaliser le modèle physique de données
- Réaliser un diagramme de séquences pour un traitement décrit un peu plus loin.

Dans ce TP il n'y a pas de développement à réaliser. Cependant cela n'est pas interdit, vous gérez votre temps comme vous le souhaitez.

Tâches à réaliser

- Analyser le fichier CSV afin d'identifier les différentes entités métier (i.e. le modèle métier).
    - Exemple : Athlete, Epreuve, etc.
- Réaliser un dossier technique rassemblant :
    - Le diagramme de classes des entités métier,
    - Le modèle physique de données (i.e. les tables).
    - Le diagramme de séquences pour le traitement d'insertion des athlètes (cf ci-dessous).
- Créer un dépôt GitHub appelé **projet-jo**
    - Créer un répertoire **conception** et déposez-y votre dossier technique au format **PDF ou Word**
- **Le diagramme de séquences à réaliser :**

L'idée est de réaliser le diagramme de séquence pour le traitement suivant : ce traitement parse le fichier et met en base de données uniquement les athlètes. Comme ce traitement doit être rejouable il n'est pas question de créer les athlètes en double.

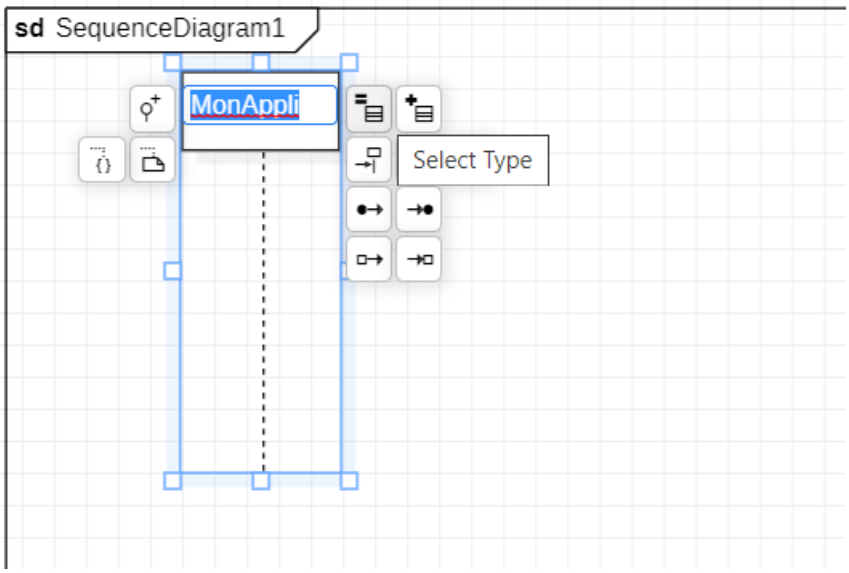
- **1.** La classe qui lance le traitement est une classe exécutable nommée **InsertionAthlete**
- **2.** L'extraction des athlètes du fichier doit être réalisée par une classe de service appelée **AthleteService**.
    - Cette classe possède une méthode **getAthletes** qui prend en paramètre le path d'accès au fichier et retourne la liste d'athlètes.

- **3.** Les échanges avec la base de données sont réalisés via une classe type DAO appelée **AthleteDao**. Cette DAO possède 3 méthodes :
  - Une méthode qui vérifie si un(e) athlète existe en base ou non. Cette méthode prend en paramètres ce qui permet d'identifier un athlète de manière unique et retourne un booléen.
  - Une méthode qui permet de mettre en base de données l'athlète si il/elle n'existe pas.
  - Une méthode qui permet de mettre à jour les informations de l'athlète si il/elle existe.

Aide pour le diagramme de séquence sur StarUML

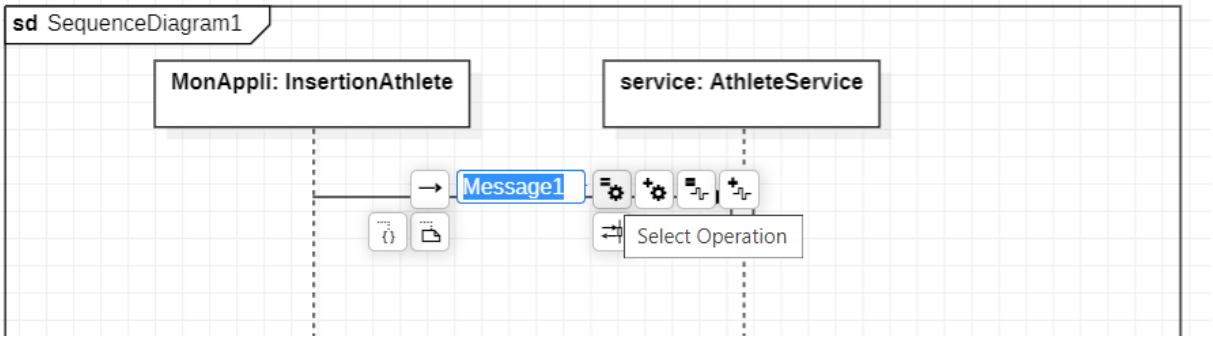
Le diagramme de séquences sur StarUML est assez compliquée, c'est la raison pour laquelle je propose de vous accompagner pour la réalisation du début du diagramme de séquences.

1. Ici il n'est pas nécessaire de préciser un acteur qui va générer un message initial puisqu'il s'agit d'une application qu'on lance via une ligne de commande ou depuis un IDE. Si vous y tenez vous pouvez créer un acteur Admin qui exécute la méthode main (message main).
2. Tout d'abord vous devrez créer les classes InsertionAthlete, AthleteService et AthleteDao ainsi que les diverses méthodes. Une lifeline est une instance de classe.
3. Lorsque vous placez une LifeLine sur le diagramme de séquence, vous devez l'associer à un type, c'est-à-dire une classe. Pour se faire :
   a. placez la lifeline sur le diagramme,
   b. donnez-lui un nom
   c. puis Cliquez sur l'icône « Select Type », celle qui contient le caractère =. Cela vous permettra de relier votre lifeline à votre classe **InsertionAthlete**.



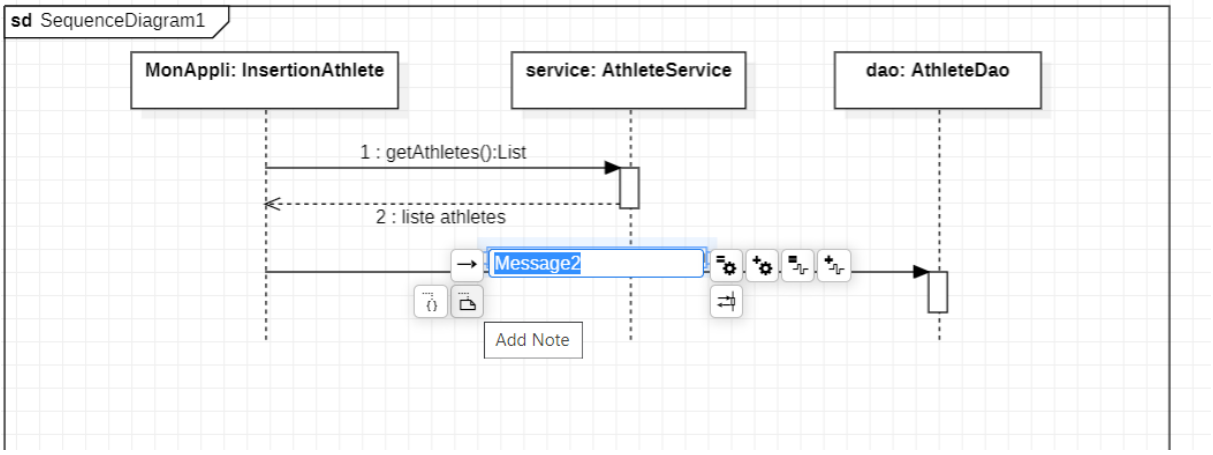
4. Créez également une lifeline pour votre classe AthleteService
5. Un message correspond à un appel de méthode.
   a. Sélectionner « Message » dans le menu de gauche

b. Créer le message entre vos 2 lifelines :
    i. Ne lui donnez pas de nom. Cliquez directement sur l'icône « Select Operation » pour sélectionner la méthode getAthletes que vous avez dû créer au préalable.

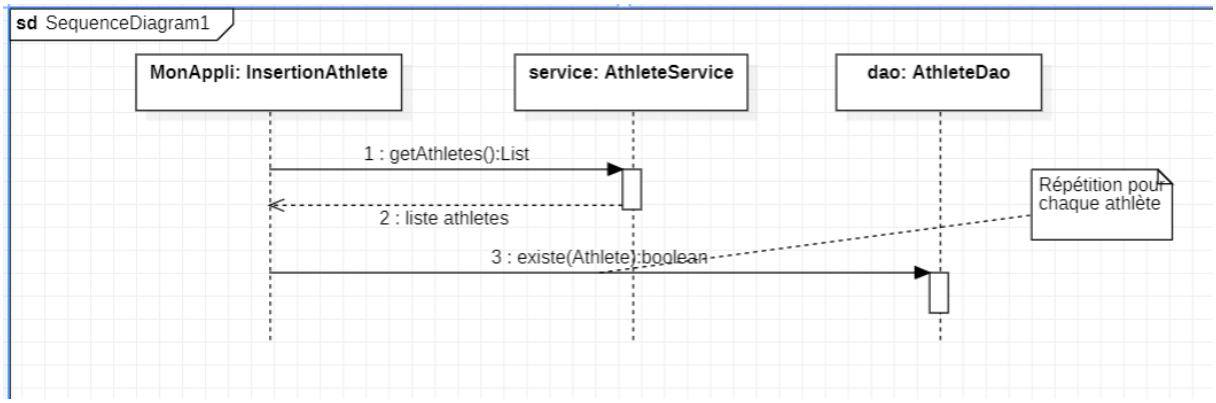


c. Créer ensuite un reply message en indiquant par exemple : Liste athlètes

6. Pour indiquer par exemple qu'un message (appel de méthode) doit être réalisé pour chaque athlète vous pouvez attacher une note au message. Remarquez l'icone add note située sur la gauche.



7. Et voici le résultat obtenu :



8. Pour créer une condition qui permet par exemple de ne créer l'athlète que si il/elle n'existe pas vous pouvez utiliser le **Combined Fragment**, mais je vous laisse trouver comment ça s'utilise. Vous trouverez de la doc sur le web 😊.