

Secours 2030

Détail de la partie «Programmation Orientée Objets» (R2.01)

A. BELMOUHCINE (S2.01), R. FLEURQUIN (R2.01)

2024 - 2025

1 Architecture Modèle-Vue-Contrôleur (MVC)

L'application finale suivra une architecture logicielle «**Modèle-Vue-Contrôleur**» (voir cours R2.02). Cette architecture permet de définir trois modules relativement indépendants :

- **Le modèle** qui décrit le modèle de données choisi ainsi que les traitements (mathématiques, par exemple) à effectuer sur les données ;
- **La vue** qui gère les éléments visibles de l'interface graphique (graphes, tableaux, boutons...) et la logique permettant d'afficher ces éléments ;
- **Le contrôleur** qui traite les actions de l'utilisateur en faisant le lien entre modèle et vue (mise à jour d'un affichage de la vue, ajout de données dans le modèle...).

En R2.01, nous travaillerons sur la partie «modèle» : vous transformerez votre base de données en code Java en adoptant une architecture orientée objets. L'objectif est de manipuler vos données, non plus comme des tables issues d'une base de données MySQL ou Oracle, mais sous forme d'objets Java, afin de pouvoir effectuer des traitements directement sur ces derniers.

2 Travaux à réaliser : Traduction BDD vers POO

1. Le diagramme de la Figure 1 (ci-dessous) est une modélisation de la base de données « **Secours 2030** ». En vous basant sur cette modélisation des tables, proposez un diagramme de classes permettant de manipuler ces différentes tables sous forme d'objets. Choisissez des types pertinents pour vos attributs¹. Ajoutez des méthodes (constructeurs, getters/setters, affichage, manipulation des objets...). Chaque classe devra avoir au moins une méthode (hors constructeur, getters, setters et `toString`).
2. Choisissez un scénario de cas d'utilisation et réalisez le diagramme de séquence correspondant. Ne représentez pas la manipulation des tableaux et/ou des listes ni les appels à `System.out.println`. Il n'est pas nécessaire de réaliser un diagramme de séquence détaillé. La complexité du scénario choisi sera prise en compte.
3. Implémentez le code correspondant à l'ensemble de votre diagramme de classes. Les classes correspondantes devront être dans un package `modele` (dans lequel vous pouvez ajouter des sous-packages si besoin).

¹Par exemple, les champs avec un domaine prédéfini réduit peuvent être représentés sous forme de type `Enum`. Une énumération en Java permet de lister le nombre fini de valeurs que peut prendre une variable. Les énumérations sont créées dans un fichier séparé, comme une classe (cf. <https://docs.oracle.com/javase/tutorial/java/java00/enum.html>). Vous pouvez utiliser tout ce qui a été vu en cours (par exemple, ajout d'interfaces, d'héritage, etc.).

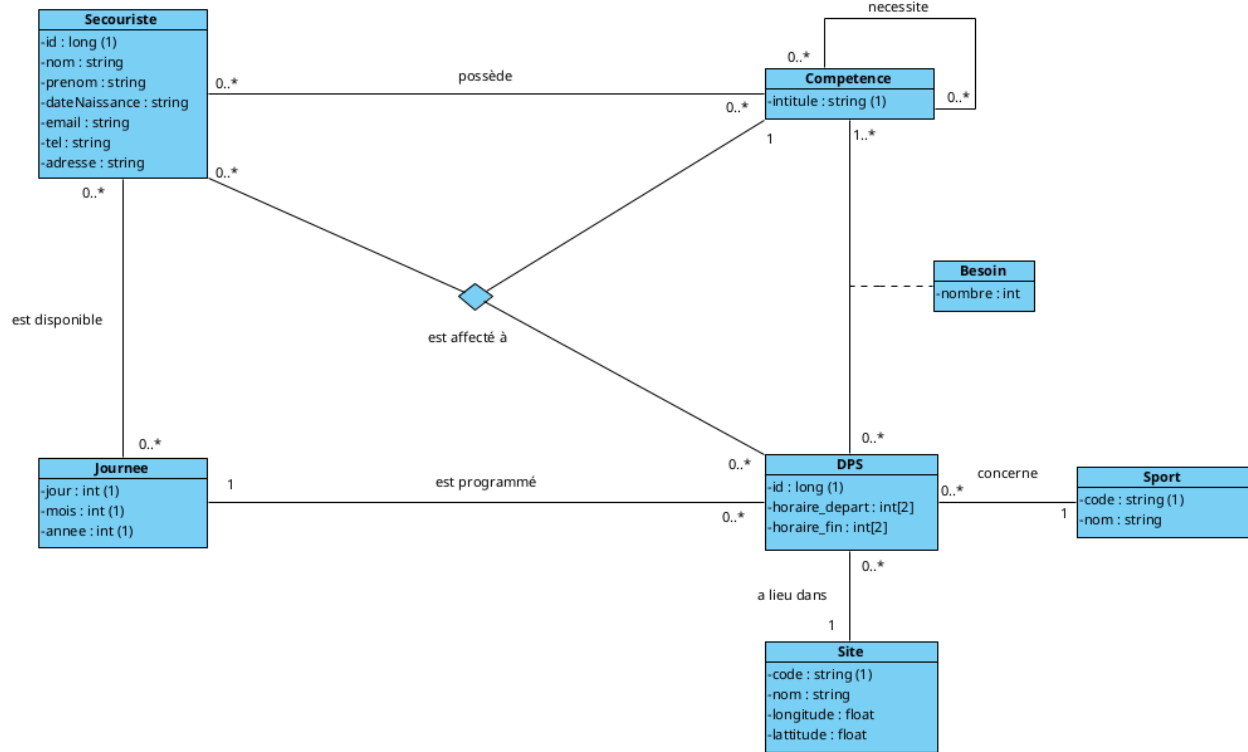


Figure 1: Diagramme de conception de la base de données

- Implémentez une classe `Scenario` testant le scénario modélisé par votre diagramme de séquence. Cela correspond à une classe principale en Java qui testera le scénario dans sa méthode `main` (un simple test dans la `main` suffit).

3 Livrables

Chaque groupe doit rendre une archive contenant :

- Le diagramme de classes correspondant au diagramme de conception de la BDD (Figure 1), accompagné d'une explication textuelle des différents choix effectués (explication des méthodes ajoutées, des associations, des types des attributs lorsque ceux-ci ne sont pas évidents, etc.) ;
- Le diagramme de séquence du scénario choisi ;
- Le code complet de la partie « modèle », commenté en français ou en anglais ;
- Une classe `Scenario` permettant d'implémenter et tester le scénario modélisé par le diagramme de séquence.

Cette archive devra être remise au plus tard **jeudi 19 juin 2025 à 23h59** sur l'espace Moodle de la SAÉ **S2.01**, en un exemplaire par groupe. Les noms de tous les membres du groupe doivent être présents sur le rendu.

En plus de l'archive, chaque étudiant devra rendre **une page** contenant une description des différentes tâches réalisées ou en cours de réalisation, au plus tard **vendredi 13 juin 2025 à 23h59**.

Le travail devra être réalisé lors des séances de **R2.01** des semaines 12 et 22, des séances de **S2.01** des semaines 23 et 25, ainsi que pendant les créneaux SAÉ de travail en autonomie.