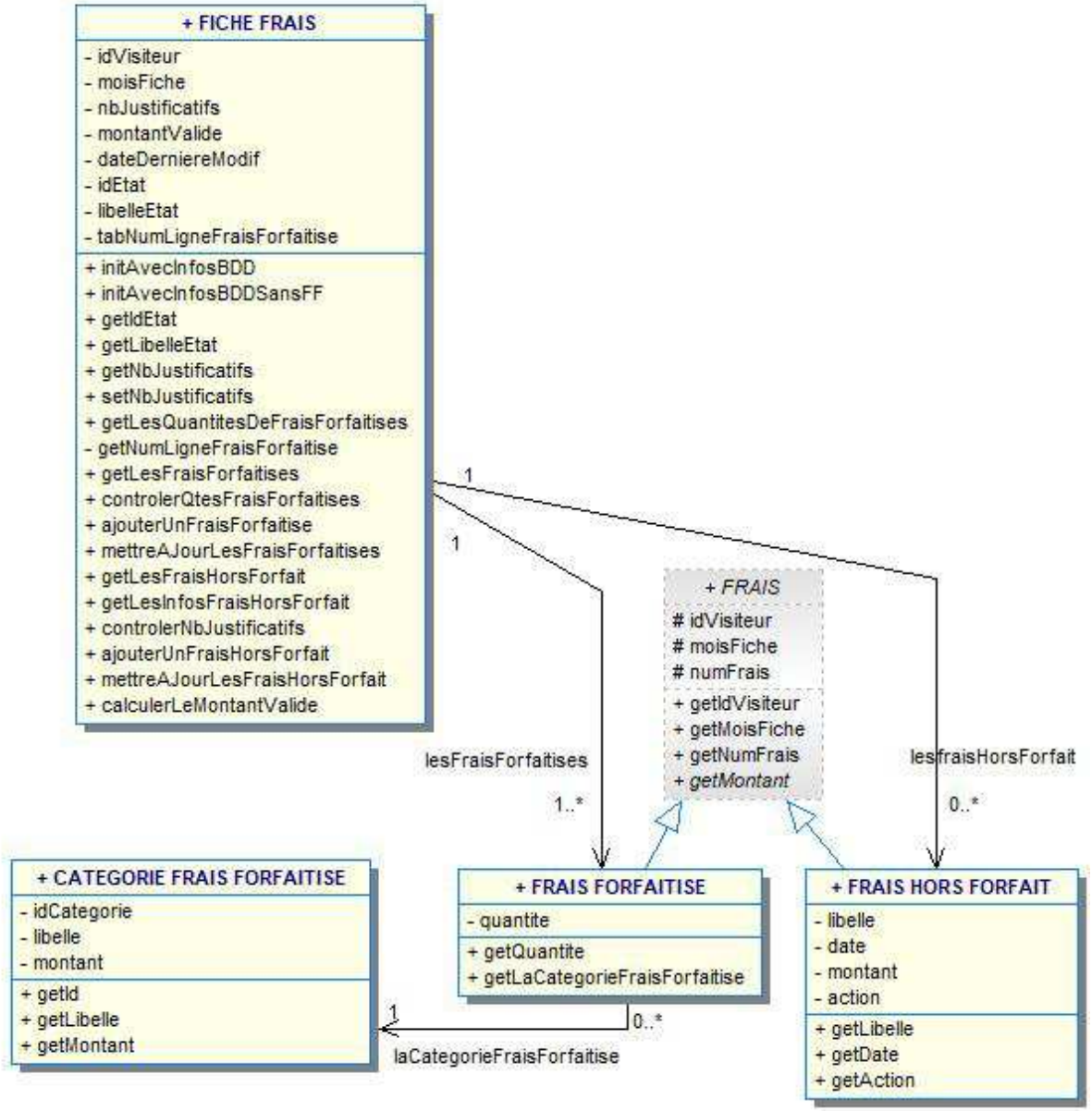
7.3.4 Etape 4 Développement du formulaire de validation d'une fiche de frais

A ce stade du projet, les classes métiers suivantes ont été modélisées (rappel : dans l'architecture MVC les classes métier font partie du modèle) :



Questions

1. *A partir du diagramme des classes et de la maquette de la fiche de frais, expliquez pourquoi on a choisi de relier la classe FICHE FRAIS à FRAIS FORFAITISE et à FRAIS HORS FORFAIT plutôt que de relier la classe FICHE FRAIS directement à FRAIS.*
2. *Pourquoi la classe Frais est-elle abstraite ?*

Si ce n'est pas encore fait, intégrez à votre projet les classes fournies dans les ressources. Vous allez travailler sur ces classes.

7.3.4.1 Les classes d'objets

7.3.4.1.1 La classe CategorieFraisForfaitise

Il manque un accesseur à la classe CATEGORIE FRAIS FORFAITISES : ajoutez-le.
Le constructeur de la classe utilise la méthode `getInfosCategorieFrais()` pour obtenir les informations sur la catégorie de frais. Développez cette méthode et la procédure stockée nommée SP_CATEGORIE_FF_GET_INFOS qui va avec.

7.3.4.1.2 La classe FraisForfaitise

A quoi sert le mot clé `final` dans la déclaration de cette classe ?

Complétez la définition de la classe :
- ajoutez les propriétés manquantes (appuyez-vous sur le diagramme des classes).
- Complétez le constructeur. Il doit initialiser toutes les propriétés de la classe (5 en tout).
- Implémentez les accesseurs.

7.3.4.1.3 La classe FraisHorsForfait

Complétez la définition de la classe :
- ajoutez les propriétés manquantes (appuyez-vous sur le diagramme des classes) **sauf `action`.**
- Complétez le constructeur. Il doit initialiser toutes les propriétés de la classe (6 en tout).
- Implémentez les 2 accesseurs.

7.3.4.1.4 La classe FicheFrais

Nous nous intéressons d'abord aux frais forfaitisés. D'après le scénario nominal du cas d'utilisation "valider fiche frais", l'instanciation de la classe `FicheFrais` doit être réalisée dans 2 étapes :
- à l'étape 4 : Le système affiche le détail de la fiche de frais – frais forfaitisés et hors forfait.
- à l'étape 5 : L'utilisateur actualise les informations des frais forfaitisés.

Afin de répondre à ce besoin il faut donc commencer par implémenter l'initialisation de la classe. La solution suivante a été retenue :

- le constructeur attend en paramètre l'identifiant visiteur et le mois de la fiche de frais. Il initialise les propriétés correspondantes.

- Une méthode nommée `initAvecInfosBDD()`, appelée après le constructeur, poursuit l'initialisation de toutes les autres propriétés de l'objet :
  - informations de la fiche : état (id et libellé), nombre de justificatifs, montant validé et date de la dernière modification ;
  - frais forfaitisés et leur catégorie ;
  - frais hors forfait.

L'appel du constructeur et de cette méthode permettra de disposer d'une fiche de frais complète qui pourra être affichée comme décrit à **l'étape 4**.

- Une méthode nommée `initAvecInfosBDDSansFF()`, appelée après le constructeur, poursuit l'initialisation :
  - de l'état de la fiche (identifiant et libellé) et du nombre de justificatifs ;
  - des frais hors forfait ;
  - mais pas des frais forfaitisés.

L'appel du constructeur et de cette méthode permettra de disposer d'une fiche de frais sans frais forfaitisés. Elle pourra être complétée ensuite avec les informations des frais forfaitisés qui auront été saisies, ce qui permettra ainsi de réaliser l'actualisation des informations des frais forfaitisés (i.e l'enregistrement de ces informations dans la base de données) décrite à **l'étape 5**.

Travail à faire

- Commencez par compléter le constructeur.

- Les 3 étapes restantes du processus d'initialisation seront appelées en fonction des besoins par `initAvecInfosBDD()` et `initAvecInfosBDDSansFF()`. Implémentez ces 3 étapes sous la forme de 3 méthodes :
  - `initInfosFicheSansLesFrais()` : initialise les autres informations de la fiche, sans les lignes de frais. Si la fiche pour le visiteur et le mois considérés n'existe pas, l'état de la fiche doit être '00'.
  - `initLesFraisForfaitises()` : initialise les frais forfaitisés
  - `initLesFraisHorsForfait()` : initialise les frais hors forfait

  Ces 3 méthodes devront se procurer les informations nécessaires dans la base de données. Dans le modèle MVC les objets métier du modèle font appel à la partie persistance des données de ce même modèle, Donc ici les 3 méthodes devront faire appel à des méthodes de la classe `PdoGsb`.

  - Ajoutez à la classe `FicheFrais` une propriété statique nommée `pdo` et initialisez-la dans le constructeur afin qu'elle contienne une référence à la classe `PdoGsb` (consultez dans la documentation PHP "L'opérateur de résolution de portée (::)" pour savoir comment accéder aux membres `static` d'une classe.

  - Dans la classe `PdoGsb` implémentez les 3 méthodes en veillant à définir correctement leur visibilité … Vous pouvez les nommer respectivement :
    - `getInfosFiche()` ;
    - `getLignesFF()` ;
    - `getLignesFHF()`. Attention ! Une fiche de frais ne contient pas obligatoirement des frais hors forfait. Dans le cas où une requête ne retourne aucun enregistrement, la méthode `fetchAll()` de la classe `pdoStatement` génère une erreur d'exécution. Pour savoir si une requête a retourné des enregistrements il faut utiliser la méthode `rowCount()` de la classe `pdoStatement`. Si c'est le cas cette méthode devra retourner un tableau vide.

- Implémentez les méthodes `initAvecInfosBDD()` et `initAvecInfosBDDSansFF()`.