

## **Exercice : Gestion simplifiée d'un hôpital**

On souhaite développer une application simplifiée de gestion d'un hôpital en utilisant les concepts de la programmation orientée objet en Java. L'objectif est de modéliser les entités principales du système (patients, médecins, hôpital) et de gérer leurs interactions en utilisant uniquement des structures de type ArrayList.

Dans un premier temps, on s'intéresse à la classe Patient, qui représente un patient de l'hôpital. Cette classe doit contenir des informations telles que le CIN, le nom, le prénom, le numéro de sécurité sociale ainsi que l'âge du patient. Il est demandé de définir les constructeurs nécessaires, les méthodes d'accès (getters et setters), ainsi que la méthode toString(). La méthode equals() devra également être redéfinie de manière à considérer que deux patients sont égaux si leurs CIN et numéros de sécurité sociale sont identiques.

Ensuite, on souhaite définir une interface nommée InterfacePatient permettant de gérer un ensemble de patients. Cette interface devra déclarer les opérations classiques de gestion, notamment l'ajout, la suppression et la recherche d'un patient à partir de son CIN, ainsi qu'une méthode permettant d'obtenir la liste des patients.

Une classe ListPatients sera ensuite implémentée en se basant sur cette interface. Cette classe devra utiliser un ArrayList pour stocker les patients et fournir une implémentation concrète des différentes opérations définies dans l'interface, en particulier la recherche d'un patient à partir de son CIN.

Par ailleurs, on souhaite modéliser les médecins à l'aide d'une classe Medecin. Cette classe devra contenir des informations telles que l'identifiant du médecin, son nom, son prénom ainsi que sa spécialité. Comme pour la classe Patient, il est demandé de définir les constructeurs, les accesseurs, la méthode toString() ainsi que la méthode equals(), qui devra considérer que deux médecins sont égaux si leurs identifiants sont identiques.

Une classe ListMedecins devra également être définie afin de gérer un ensemble de médecins en utilisant un ArrayList. Cette classe devra permettre l'ajout, la suppression et la recherche d'un médecin à partir de son identifiant, ainsi que la consultation de la liste des médecins.

Enfin, une classe Hopital devra être implémentée afin de représenter le système global. Cette classe devra contenir une liste de médecins et une liste de patients, toutes deux basées sur des ArrayList. Elle devra également permettre d'associer des patients à des médecins en utilisant uniquement des structures de type ArrayList. Des méthodes devront être définies pour ajouter des patients et des médecins, affecter un patient à un médecin, récupérer la liste des patients suivis par un médecin donné et afficher les différentes affectations.

Une attention particulière devra être portée à la gestion des exceptions. En effet, il est demandé de prévoir des traitements appropriés dans les cas où des données invalides sont manipulées, notamment lors de l'ajout de patients ou de médecins, ou encore lors de l'affectation d'un patient à un médecin inexistant ou inversement.

L'ensemble de ce travail devra être réalisé en respectant strictement la contrainte d'utilisation exclusive des ArrayList, sans recours à d'autres structures de données.

On demande de développer une classe principale contenant la méthode main afin de tester l'ensemble des fonctionnalités de l'application.