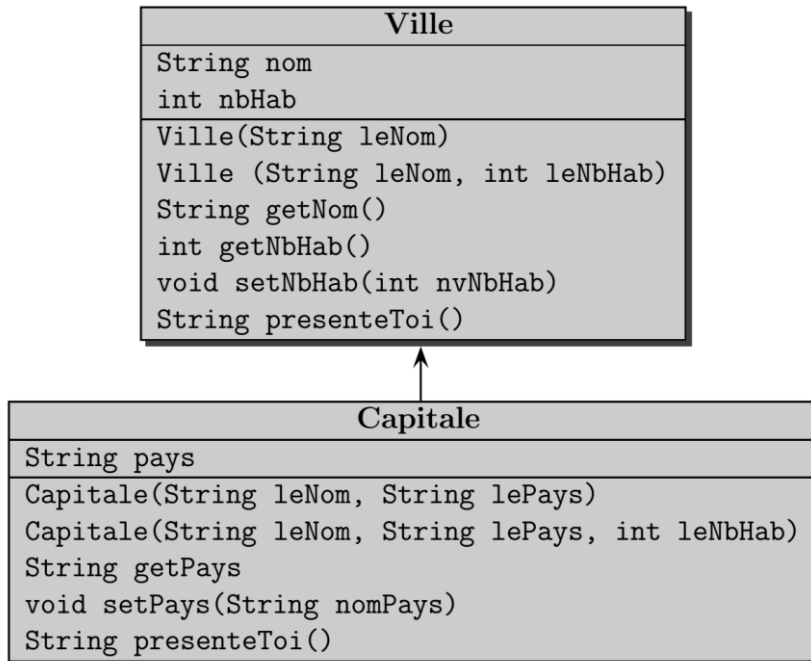


Module : POO

TD Encapsulation, Héritage et polymorphisme

Considérons un exemple complet d'une classe Parent Ville et d'une classe Enfant Capitale.



Partie 1 : Encapsulation

- 1) Implémentez la classe « Ville » à l'image du diagramme de classe UML précédent :
 - a. Conformément au concept d'encapsulation vous déclarerez tous les attributs avec un modificateur de visibilité « **private** » et les méthodes avec un modificateur « **public** ».
 - b. Vous initialiserez le nombre d'habitants à -1 dans le premier constructeur. Cela signifie qu'il est inconnu.
 - c. Le deuxième constructeur est une surcharge du premier. Il devra le compléter par l'initialisation du nombre d'habitants qui ne doit en aucun cas être inférieur à 0. Auquel cas un message d'erreur est affiché et le nombre d'habitants sera initialisé à -1.
 - d. `presenteToi()` est une méthode qui retourne le nom et le nombre d'habitants d'une ville. Si ce nombre est inconnu, il indiquera « inconnu » pour le nombre d'habitant.
 - e. Peut-on modifier la variable d'instance `nomVille` depuis l'extérieur de la classe ?

Partie 2 : Héritage

- 2) Implémentez la classe « Capitale » qui hérite de la classe « Ville »:
 - a. Expliquer le mécanisme de construction d'une classe dérivée. Tester les différents cas : appel explicite ou nom au constructeur de la classe mère ; existence ou nom d'un constructeur sans arguments.
 - b. La classe Capitale redéfinit la méthode présenteToi() qui en plus du nom et du nombre d'habitants affiche le nom du pays de la capitale.
 - c. Changer les modificateurs d'accès des données membres de la classe mère, en remplaçant private par protected. Peut-on accéder à ces variables de l'extérieur de la classe Ville ?
- 3) Réalisez une classe « Monde » dans laquelle :
 - a. vous créez 2 Villes et 2 Capitales en exploitant chacun des constructeurs définis dans les différentes classes.
 - b. Vous afficherez pour chacun des objets ses caractéristiques en faisant appel à la méthode « présenteToi »

Partie 3 : Polymorphisme

- 4) Montrer que l'héritage permet le polymorphisme. Pour ce faire créer un tableau de 5 Villes comprenant des villes et des capitales.
- 5) Affichez les caractéristiques de l'ensemble des éléments du tableau.