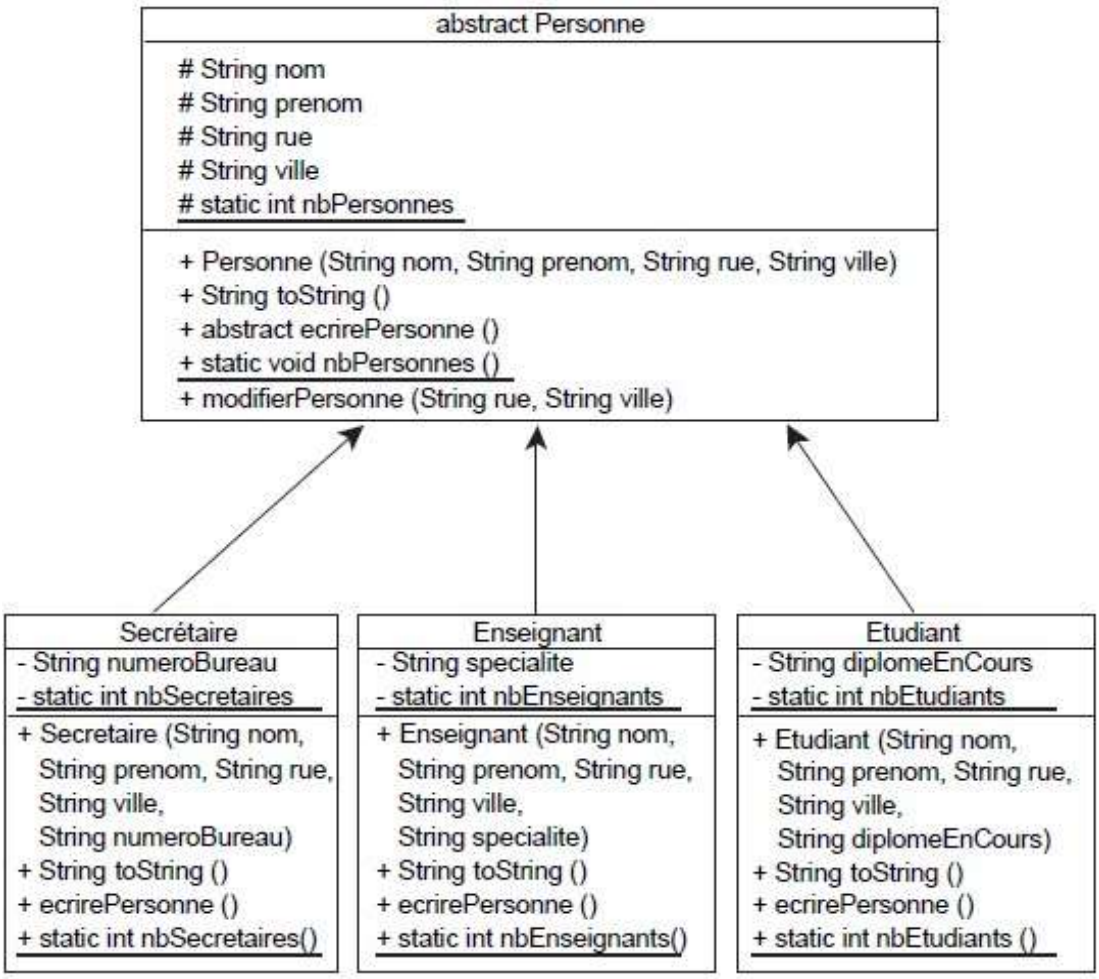
TP 5 : Abstract class and Interface

Exercice 01 : classe abstraite

In an educational institution, there are three types of people: administrative staff represented by the category secretary, teachers and students. Each person is characterized by their surname and first name, their address (street and town), which are attributes that are protected and common to all people. A person can be represented using a UML diagram as shown in the figure..



Arbre d'héritage.

Les classes Secretaire, Enseignant et Etudiant sont des classes dérivées de la classe Personne. La classe Personne est abstraite et ne peut être instanciée.

# indicates a protected attribute, - indicates a private attribute, + indicates a public attribute.

The instance variables (attributes) are surname, first name, street and town. nbPersonnes is a class variable (i.e. static) that counts the number of People in the establishment.

The following public methods are defined for the **Personne** class:

• the **Personne** constructor (String surname, String first name, String street, String city): creates and initialises an object of type Person.

• String **toString** (): provides a string of characters corresponding to the characteristics (attributes) of a person.

• **ecrirePersonne** (): could write the characteristics of a person. In the example below, it does nothing. It is declared abstract.

• static **nbPersonne** ():writes the total number of people and the number of people per category. This is a class method.

• **modifierPersonne** (String street, String city): modifies a person's address and calls **ecrirePersonne**() to verify that the modification has been made.

Questions :

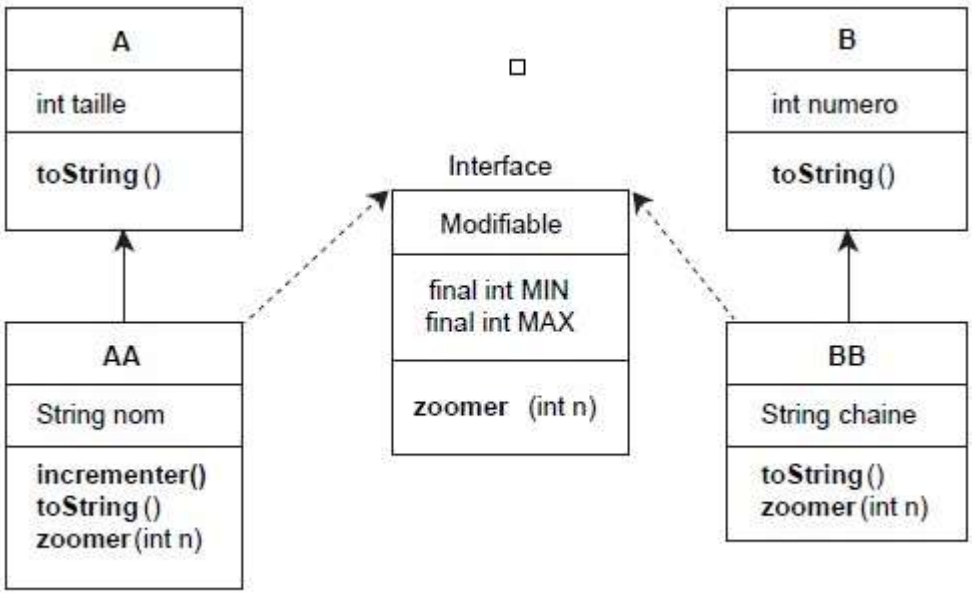
1) Implement classes.

2) Complete the following PPPersonne1 executable class:

```
public class PPPersonne1 { // classe exécutable public
static void main(String[] args) {
Secretaire Sec=new Secretaire("Seba","Zohra","rue des rosiers", "Alger", "A326");
Enseignant Ens=new Enseignant ("Fridi","Boualem","rue des mimosas", "Alger", "Mathematique");
Etudiant Etu=new Etudiant ("Smati","Ryad","rue des lilas", "Alger", "Informatique") ;
// à compléter … display the number of persons
System.out.println("\nAprès modification:\n");
/* à compléter … modify the adresse of « Seba Zohra »  with : "rue des orangers, ville de Blida. */
/* à compléter … modify the adresse of « Fridi Boualem » with : "rue des
marguerites", "Kouba". */
// à compléter … display the information of all persons
}
```

Exercice 02 : interface

We define a class AA that inherits from class A and a class BB that inherits from class B. The Modifiable interface allows us to view classes AA and BB as having a common property, that of being modifiable. An interface only defines constants and method prototypes (signatures).



Héritage et interface.

Questions :

1) Implement the classes and interface above.

2) Complete the following executable class PPInterface:

```
public class PPInterface { // classe exécutable

public static void main (String[] args) {

// to be completed ...  Create a table of Modifiable containing two objects AA and BB

// to be completed ...  display the table

// to be completed ...  trigger the zoomer() method for the elements in the array.
// The AA elements are divided by 2.
// The BB elements are divided by 4.

// to be completed ...  display the table

}
}
```