

Prénom Nom :

Groupe :

EVALUATION N°1

UML et Java

Dans ce devoir, les bonnes pratiques ne sont pas toujours respectées, notamment en ce qui concerne le nommage des variables. C'est volontaire pour vous éviter de perdre du temps en écriture

Écrivez le code directement sur le sujet.

Exercice 1 *La classe Ville*

6 pts

Ville
- nom:String - nb:int //nombre d'habitants de la ville
+ Ville(nom:String, habitants:int) + getNb():int // renvoie le nombre d'habitants + getNom():String + equals(obj:Object):boolean

1.1 Dans le diagramme de classe, ajoutez un setteur à la classe `Ville`

1.2 Donnez le code Java de la classe `Ville`. Vous documenterez uniquement la méthode `getNb()`.

Exercice 2 *Compiler, exécuter*

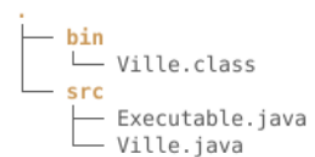
3 pts

Voici le code Java d'une classe exécutable :

```
public class Executable {
    public static void main(String[] args) {
        Ville orleans = new Ville("Orléans", 114644);
        Ville olivet = new Ville("Olivet", 21639);
        Ville cenabum = new Ville("Orléans", 114644);
        assert orleans.getNb() == 114644;
        assert orleans.getNom().equals("Orléans");
        assert !orleans.equals(olivet);
        assert orleans.equals(orleans);
        assert orleans.equals(cenabum);
    }
}
```

On admet que la classe Ville a été codée et compilée.

On ouvre un terminal et voici le contenu du dossier courant :



2.1 Quelle ligne de commande permet de compiler le code Java de la classe Executable ?

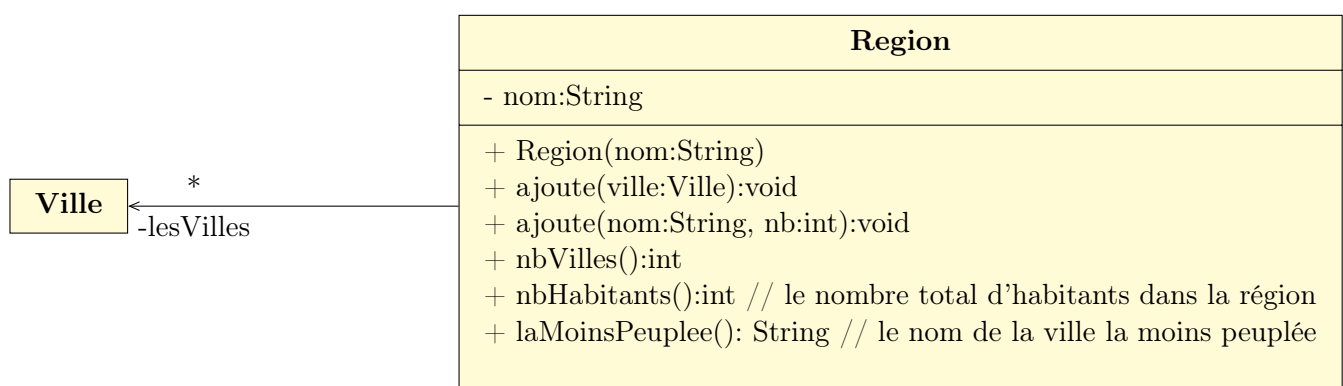
2.2 Quelle ligne de commande permet d'exécuter l'Executable (en activant les tests) ?

2.3 Quelle ligne de commande permet de générer la javadoc dans un dossier doc ?

Exercice 3 *La classe Region*

8 pts

On complète le diagramme de classes de notre projet de la façon suivante. La classe **Ville** n'est pas détaillée : elle correspond à la classe de l'exercice 1.



On complète le code de la classe exécutable de la façon suivante :

```
public class Executable {  
    public static void main(String[] args) {  
        // ... code donné à l'exercice 2  
  
        Region centre = new Region("Centre Val de Loire");  
        centre.ajoute(orleans);  
        centre.ajoute("Olivet", 21639);  
        centre.ajoute("Montargis", 14738);  
        centre.ajoute("Pithiviers", 9067);  
        centre.ajoute("Feins-en-Gâtinais", 38);  
  
        assert centre.nbVilles() == 5;  
        assert centre.nbHabitants() == 160126;  
        assert centre.laMoinsPeuplee().equals("Feins-en-Gâtinais");  
    }  
}
```

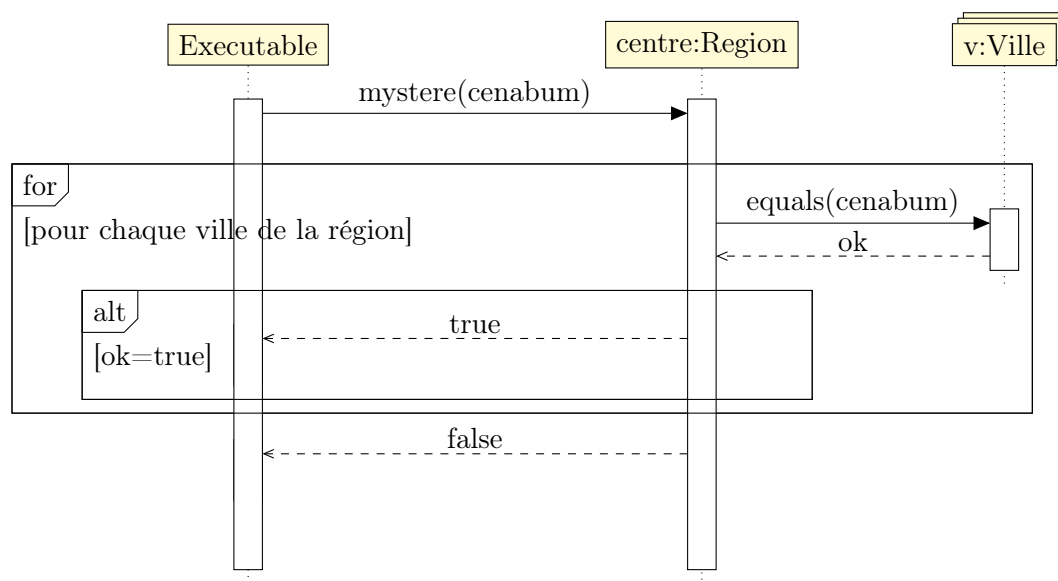
3.1 Donnez le code Java de la classe `Region`. On ne demande pas la documentation ici.

Exercice 4 *Code et diagramme de séquence*

3 pts

On ajoute une méthode `mystere(Ville ville)` à la classe `Region`, voici le diagramme de séquence correspondant à l'exécution de la ligne de code

`System.out.println(centre.mystere(cenabum));` dans la classe `Executable`.



4.1 Proposez un nom plus approprié pour la méthode `mystere` et donnez son code Java, y compris la documentation.

