

Conception orientée objet

Application du cours 1 : Village gaulois

Le sujet permet de nous entraîner à implémenter une modélisation UML. Nous aborderons un diagramme statique : le diagramme de classe et un diagramme dynamique : le diagramme de séquence.

I. Environnement du problème

Nous travaillons sur les habitants de l'Armoricae d'après l'œuvre de René Goscinny, Albert Uderzo et Jean-Yves Ferri : « Astérix ». Nous étudierons donc le village gaulois et les camps romains installés à proximité.

II. Questions

- 1- Compléter le diagramme de classes (page 2) des forces romaines faisant intervenir les classes **CampRomain**, **Legion**, **Centurie** et **Soldat**. Le code de ces classes vous est donné page 5 à 6 (ne prendre en compte que la partie 1 de chacune de ces classes).
- 2- Modifier le diagramme de classe (page 2) et le code (page 7) pour que l'on ne puisse pas créer des personnages mais seulement des gaulois, des romains ou des soldats.
- 3- A l'aide du diagramme de séquence (page 3) compléter le constructeur de la classe **Armoricae** (page 8) pour créer le camp Aquarium, et l'ajouter à son tableau *campsRomains*.
- 4- Compléter la classe **Legion** (page 5), pour que l'on puisse ajouter une nouvelle centurie. Compléter le diagramme de classe (page 2) afin de prendre en compte cette nouvelle méthode.
- 5- Compléter le diagramme de séquence détaillé et de classes participantes, pour que l'on puisse ajouter un nouveau soldat.
- 6- Compléter le constructeur de la classe **Armoricae** (page 8), afin de créer le camp d'« Aquarium ».
- 7- Terminer le diagramme de séquence détaillé (page 4) afin d'obtenir la chaîne suivante à l'affichage :

```
Camps romains en Armoricae
Aquarium est le camp Romain dirigé par Alavacomgetepus : PREFET
Les légions sous ses ordres sont :

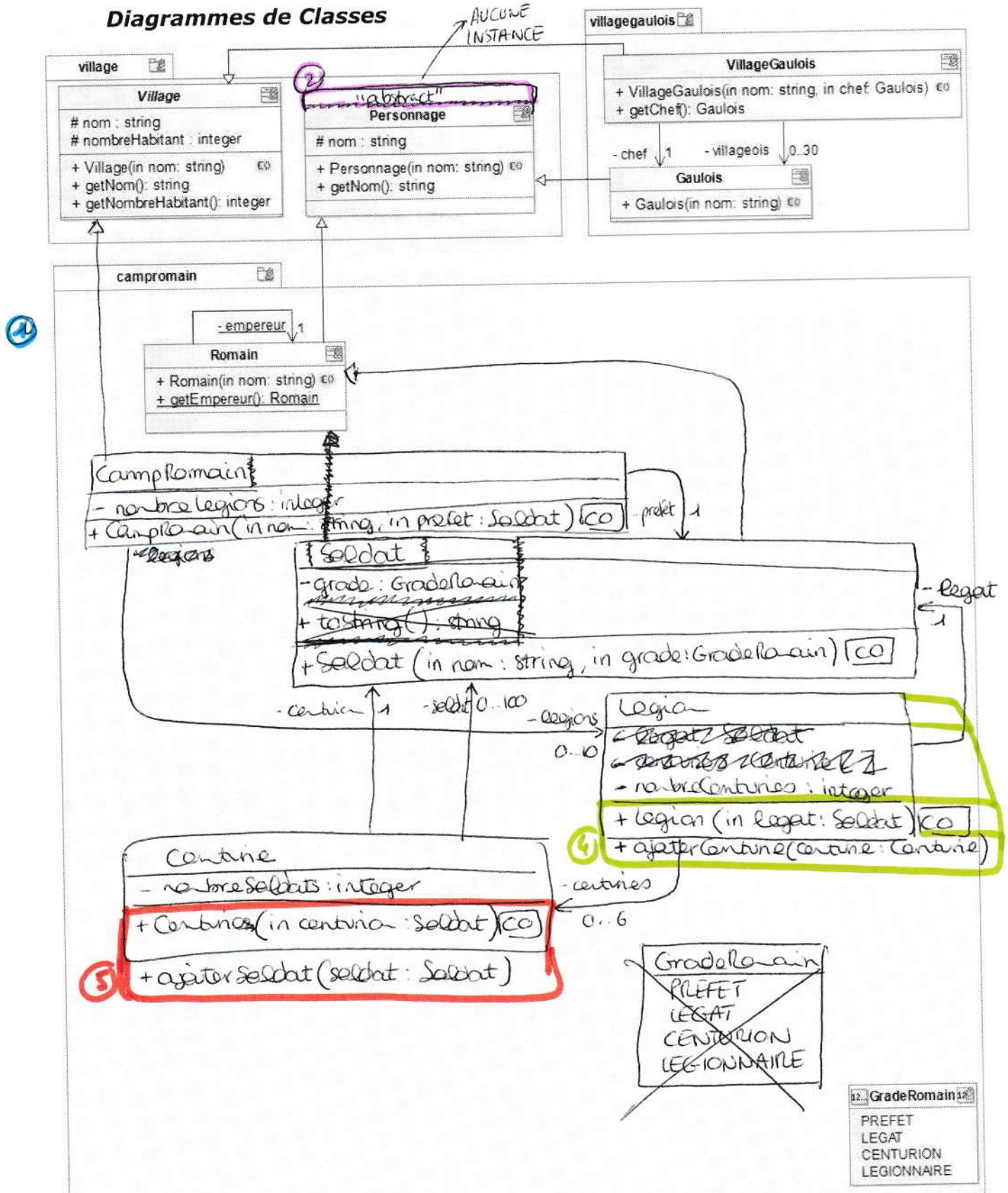
- La légion dirigée par Aérobus : LEGAT
  Les centurions sous ses ordres sont :
  * La centurie dirigée par Oursenplus : CENTURION
    Les soldats sous ses ordres sont :
    . Cornedurus : LEGIONNAIRE
    . Deprus : LEGIONNAIRE

  * La centurie dirigée par Mordicus : CENTURION
    Les soldats sous ses ordres sont :
    . Montladsus : LEGIONNAIRE
    . Savancosinus : LEGIONNAIRE
    . Huméris : LEGIONNAIRE
```

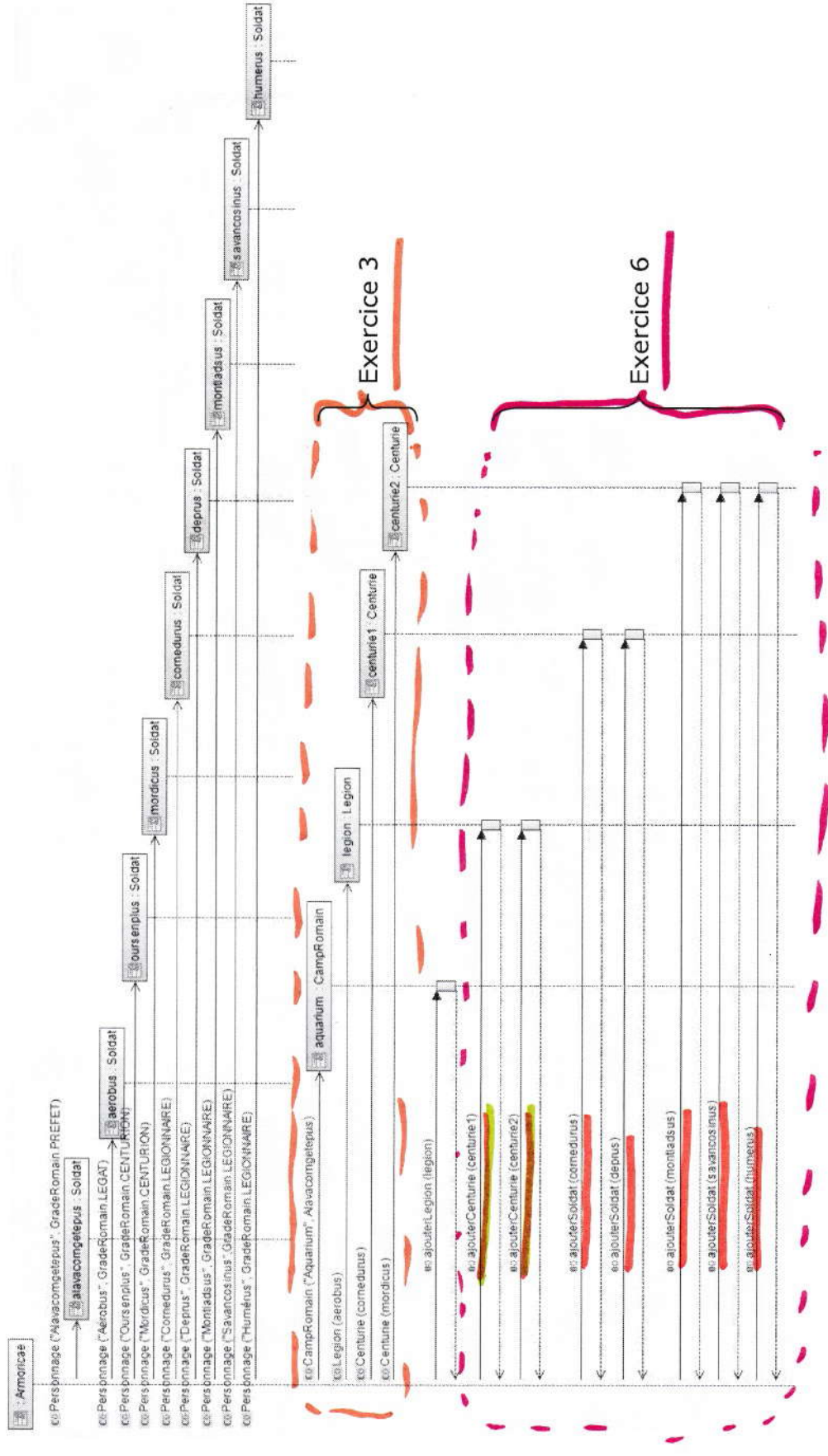
III. Diagrammes UML

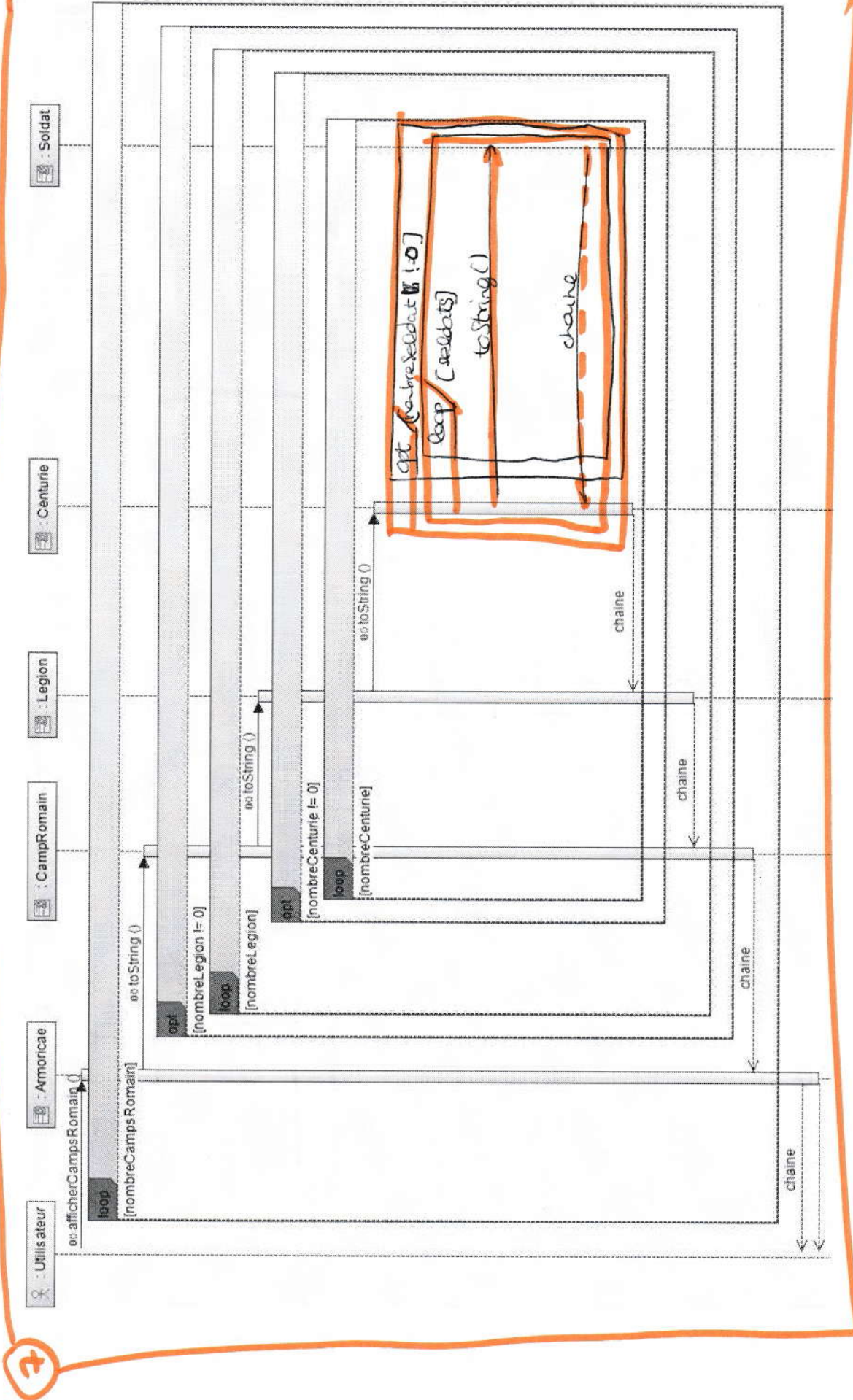
Diagrammes de Classes

un personnage est soit un romain soit un Gaulois

AUCUNE
INSTANCE

Diagrammes de Séquence détaillé





IV. Code JAVA

Classe « CampRomain »

```
package campromain;
import village.Village;

public class CampRomain extends Village {
    //Partie 1
    private Soldat prefet;
    private Legion[] legions = new Legion[10];
    private int nombreLegions = 0;

    public CampRomain(String nom, Soldat prefet) {
        super(nom);
        this.prefet = prefet;
    }

    //Partie 2
    public String toString() {
        String chaine = nom + " est le camp Romain dirigé par " + this.prefet;
        if (nombreLegions != 0) {
            chaine += "\nLes légions sous ses ordres sont : ";
            for (int i = 0; i < nombreLegions; i++)
                chaine += "\n" + legions[i];
        }
        chaine += "\n";
        return chaine;

        public void ajouterLegion(Legion legion) {
            legions[nombreLegions] = legion;
            nombreLegions++;
        }
    }
}
```

Classe « Legion »

```
package campromain;

public class Legion {
    //Partie 1
    private Soldat legat;
    private Centurie[] centuries = new Centurie[6];
    private int nombreCenturies = 0;

    public Legion(Soldat legat) {
        this.legat = legat;
    }

    //Partie 2
    public String toString() {
        String chaine = "\n- La légion dirigée par " + this.legat;
        if (nombreCenturies != 0) {
            chaine += "\n Les centuries sous ses ordres sont : ";
            for (int i = 0; i < nombreCenturies; i++)
                chaine += "\n" + centuries[i];
        }
        chaine += "\n";
        return chaine;
    }
}
```


④ `public void ajouterCenturie (Centurie centurie#) {`
`centuries[nombreCenturies] = centurie#;`
`nombreCenturies++;`
`}`

Classe « Centurie »

```
package campromain;
```

```
public class Centurie {
```

```
//Partie 1
```

```
private Soldat centurion;
```

```
private Soldat[] soldats = new Soldat[100];
```

```
private int nombreSoldats = 0;
```

```
public Centurie(Soldat centurion) {
    this.centurion = centurion;
}
```

```
//Partie 2
```

```
public String toString() {
    String chaine = " * La centurie dirigée par " + this.centurion;
    if (nombreSoldats != 0) {
        chaine += "\n Les soldats sous ses ordres sont : ";
        for (int i = 0; i < nombreSoldats; i++)
            chaine += "\n . " + soldats[i];
    }
    chaine += "\n";
    return chaine;
}
```

⑤ `public void ajouterSoldat (Soldat soldat) {`
`soldats[nombreSoldat] = soldat;`
`nombreSoldat++;`
`}`

Classe « Soldat »

```
package campromain;
```

```
public class Soldat extends Romain {
```

```
//Partie 1
```

```
private GradeRomain grade;
```

```
public Soldat(String nom, GradeRomain grade) {
    super(nom);
    this.grade = grade;
}
```

```
//Partie 2
```

```
public String toString() {
    return nom + " : " + grade;
}
```

Classe « Personnage »

abstract

```
package village;
public class Personnage {
    protected String nom;
    public Personnage(String nom) {
        super();
        this.nom = nom;
    }

    public String getNom() {
        return nom;
    }
}
```

Classe « Gaulois »

```
package villagegaulois;
import village.Personnage;
public class Gaulois extends Personnage{
    public Gaulois(String nom) {
        super(nom);
    }
}
```

Classe « Romain »

```
package campromain;
import village.Personnage;
public class Romain extends Personnage{
    private static Romain empereur = new Romain("César");
    public Romain(String nom) {
        super(nom);
    }

    public static Romain getEmpereur() {
        return empereur;
    }
}
```

Classe « Village »

```
package village;
public abstract class Village {
    protected String nom;
    protected int nombreHabitant = 0;
    public Village(String nom){
        this.nom = nom;
    }

    public String getNom() {
        return nom;
    }

    public int getNombreHabitant(){
        return nombreHabitant;
    }
}
```

Classe « Village Gaulois »

```
package villagegaulois;
import village.Village;
public class VillageGaulois
    extends Village{

    private Gaulois chef;
    private Gaulois[] villageois =
        new Gaulois[30];

    public VillageGaulois(String nom,
        Gaulois chef){
        super(nom);
        this.chef = chef
    }

    public Gaulois getChef(){
        return chef;
    }
}
```

Classe « Armoricae »

```

package gaule;
import campromain.CampRomain;           import campromain.Legion;
import campromain.Centurie;             import campromain.Soldat;
import campromain.GradeRomain;         import villageGaulois.VillageGaulois;

public class Armoricae {
    VillageGaulois villageGaulois = new VillageGaulois("village des
        irréductibles Gaulois", new Gaulois("Abraracourcix"));
    CampRomain[] campsRomains = new CampRomain[4];
    int nombreCampsRomains = 0;

    public Armoricae() {
        Soldat alavacomgetepus =
            new Soldat("Alavacomgetepus", GradeRomain.PREFET);
        Soldat aerobus = new Soldat("Aérobis", GradeRomain.LEGAT);
        Soldat oursenplus = new Soldat("Oursenplus", GradeRomain.CENTURION);
        Soldat mordicus = new Soldat("Mordicus", GradeRomain.CENTURION);
        Soldat cornedurus = new Soldat("Cornedurus", GradeRomain.LEGIONNAIRE);
        Soldat deprus = new Soldat("Deprus", GradeRomain.LEGIONNAIRE);
        Soldat montladsus = new Soldat("Montladsus", GradeRomain.LEGIONNAIRE);
        Soldat savancosinus =
            new Soldat("Savancosinus", GradeRomain.LEGIONNAIRE);
        Soldat humerus = new Soldat("Humerus", GradeRomain.LEGIONNAIRE);
    }
}

```

③ ~~CampRomain aquarium = new CampRomain("Aquarium", Alavacomgetepus);~~
~~legion legion = new Legion(aerobus);~~
~~Centurie centurie1 = new Centurie(oursenplus);~~
~~Centurie centurie2 = new Centurie(mordicus);~~
~~campsRomain[nombreCampsRomains] = aquarium;~~
~~nombreCampsRomains ++;~~

aquarium.ajouterLegion(legion);

⑥ ~~aquarium.ajouterCenturie(centurie1);~~
~~legion.ajouterCenturie(centurie2);~~
~~centurie1.ajouterSoldat(cornedurus);~~
~~centurie1.ajouterSoldat(^{deprus}~~montladsus~~);~~
~~centurie2.ajouterSoldat(montladsus);~~
~~centurie2.ajouterSoldat(savancosinus);~~
~~centurie2.ajouterSoldat(humerus);~~

```

    public String afficherCampsRomains() {
        String chaine = "Camps romains en Armoricae";
        for (int i = 0; i < nombreCampsRomains; i++)
            chaine += campsRomains[i] + "\n\n";
        return chaine;
    }
}

```