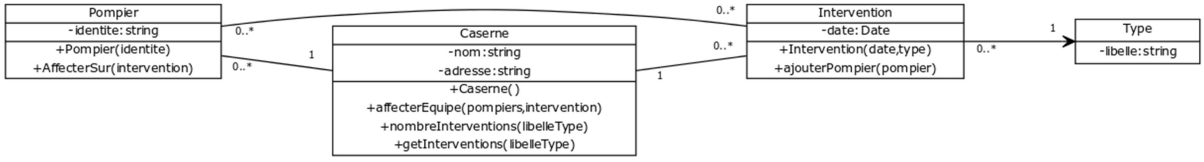
Préparation à E5	**2ème BTS SIO SLAM**
SLAM4-5	J-C HERON
CONTENU : Solutions applicatives	**Evaluation n°1**
Durée max	1 heure
Domaines abordés	**D4.1 - Conception et réalisation d'une solution applicative**
Fichier	DS1.docx

Partie 1

UML :

A partir du diagramme de classes de l'application **SDIS 14** permettant de gérer les interventions des pompiers :



1. Donner la description textuelle des classes du diagramme, en mentionnant tous les membres de données, leur type et le type d'accès (private, protected, public).

Partie 2

Implémentation :

Implémenter les méthodes ci-dessous :

Dans la classe Pompier :

2. Le modificateur **setIdentite** et l'accesseur **getIdentite** de la propriété **identite**
3. Le Constructeur **Pompier(String identite)**
   `//Initialise tous les membres de la classe pompier`
4. La méthode **void affecterSur(Intervention intervention)**
   `//Affecte le pompier sur l'intervention passée en paramètre`

Dans la classe Caserne :

5. La méthode **void affecterEquipe(Collection<Pompier> pompiers,Intervention intervention)**
   `//Affecte les pompiers passés en paramètre sur l'intervention`
6. La méthode **int nombreInterventions(String libelleType)**
   `//Retourne le nombre d'interventions effectuées correspondant au libellé du type passé en paramètre`
7. La méthode **Collection de <Intervention> getInterventions(String libelleType)**
   `//Retourne les interventions effectuées correspondant au libellé du type passé en paramètre`

```php
/**
 * Classe technique Collection
 */
class Collection{
	/**
	 * Retourne le nombre d'objets dans la collection
	 * @return int
	 */
	public function cardinal (){…}
	/**
	 * Vide la collection de ses objets
	 */
	public function vider(){…}
	/**
	 * Retourne l'objet à l'index index (à partir de 0)
	 * @param int index
	 * @return Object
	 */
	public function obtenirObjet(int index){…}
	/**
	 * Ajoute l'objet objet dans la collection
	 * @param Object objet
	 */
	public function ajouter(objet){…}
	/**
	 * Supprime l'objet objet de la collection
	 * @param Object objet
	 */
	public function supprimer(objet){…}

}
```

Pour instancier une collection : `uneCollection = new Collection() de <classe>`

Pour parcourir par itération tous les éléments d'un objet Collection :

```php
for(variable_objet in uneCollection){
	// instructions avec variable_objet
}
```

Yuml :