



**Université des Antilles**  
**UFR des Sciences Exactes et Naturelles**  
**Licence Informatique**

**TP2**  
**Ingénierie Logicielle - Tests Unitaires**

**Exercice: Le développement dirigé par les tests**

On souhaite mettre en place une classe **EnsembleChaine** permettant de manipuler des ensembles de chaîne de caractères. Cette classe doit permettre d'effectuer des opérations ensemblistes classiques : union, intersection, et unionDisjointe.

```
public class EnsembleChaine {
    ArrayList< String > liste;

    public EnsembleChaine(){
        liste = new ArrayList<String>() ;
    }

    //Ajoute la chaîne ch, si elle n'existe pas déjà, à l'ensemble
    public boolean ajouter(String ch) {

    }

    //Test si l'ensemble e est équivalent à l'ensemble courant
    public boolean equals(EnsembleChaine e) {
        return false ;
    }

    //Renvoie l'union de l'ensemble e et l'ensemble courant
    public EnsembleChaine union(EnsembleChaine e){
        return null ;
    }

    //Renvoie l'intersection de l'ensemble e et l'ensemble courant
    public EnsembleChaine intersection(EnsembleChaine e){
        return null ;
    }

    //Renvoie l'union disjointe de l'ensemble e et l'ensemble courant
    public EnsembleChaine unionDisjointe(EnsembleChaine e){
        return null ;
    }
}
```

On souhaite développer cette classe en utilisant une approche de **Développement Dirigée par les Tests (TDD)**. Pour cela nous commencerons par implémenter la classe **EnsembleChaineTest**.

1. Créer la classe **EnsembleChaine** en copiant le code ci-dessus.
2. Commencer par ajouter à la classe **EnsembleChaineTest** la méthode **ajouterTest** qui vérifie que les chaînes de caractères passées en paramètre sont bien ajoutées à l'ensemble. Pour rappel, les chaînes passées en paramètre sont ajoutées à l'ensemble que si elles ne sont pas déjà présentes dans l'ensemble. Lancer le test et vérifier qu'il échoue bien.
3. Implémenter ensuite la méthode **ajouter** de la classe **EnsembleChaine** puis relancer les tests. Retoucher la méthode jusqu'à ce que les tests passent.
4. Effectuer les mêmes opérations pour les méthodes **equals**, **union**, **intersection** et **unionDisjointe**. Vous commencerez par implémenter d'abord les tests, puis vous implémenterez la méthode. La méthode sera retouchée jusqu'à ce que TOUS les tests passent.