

# DÉMO

## INF2120

### Démo 4: Les listes chaînées et Junit

Pour cette démo, vous devez implémenter l'interface Liste ci-dessous:

```
public interface Liste <T> {  
    /**  
     * @return true si la liste est vide, false autrement  
     */  
    public boolean estVide();  
  
    /**  
     * @return la taille de la liste  
     */  
    public int taille();  
  
    /**  
     * @return le premier element de la liste  
     * @throws ListeException si liste vide  
     */  
    public T debut();  
  
    /**  
     * @return le dernier element de la liste  
     * @throws ListeException si liste vide  
     */  
    public T fin();  
  
    /**  
     * @return l'element de cette position  
     * @throws ListeException si position < 0 || position >= taille  
     */  
    public T elementPosition(int position);  
  
    /**  
     * @param element l'element d'insertion  
     */  
    public void insererDebut(T element);  
  
    /**  
     * @param element l'element d'insertion  
     */  
    public void insererFin(T element);  
}
```

```

/**
 * @param element l'element d'insertion
 * @param pos position d'insertion
 * @throws ListeException si position < 0 || position >= taille
 */
public void insererPosition(T element, int position);

/**
 * @return le premier element de la liste qui est remplacé
 * @throws ListeException si la liste est vide
 */
public T modifierDebut(T element);

/**
 * @return le dernier element de la liste remplacé
 * @throws ListeException si la liste est vide
 */
public T modifierFin(T element);

/**
 * Remplace et retourne l'element se trouvant à cette position.
 * @throws ListeException si position < 0 || position >= taille
 */
public T modifierPosition(T element, int position);
}

```

Pour y parvenir, vous avez besoin d'une classe Maillon <T> comme décrite ci-dessous:

```

public class Maillon<T> {
    private T element;
    private Maillon<T> suivant;

    // Maillon avec null comme suivant
    public Maillon (T elem) {}

    public Maillon ( T elem, Maillon<T> suivant) {}

    public T getElement () {}

    public Maillon<T> getSuivant () {}

    public void setElement ( T element ) {}

    public void setSuivant ( Maillon<T> suivant ) {}
}

```

## JUnit:

Pour chacune des implémentations de l'interface `Liste`, vous devez mettre en place des tests JUnit qui permettront de valider vos fonctions.