

**Travail pratique #6 - Tests de régression
Automne 2018**

1. Mise en contexte théorique

Une **régression** est un défaut introduit dans un logiciel suite à des corrections de défauts ou n'importe quel changement apporté : ajout de nouvelles fonctionnalités, modification de fonctionnalités existantes ou modification d'une dépendance, librairie tierce etc...

Un [test de régression](#) est un ensemble de cas de tests d'un programme préalablement testé, après une modification, pour s'assurer que des défauts n'ont pas été introduits ou découverts dans des parties non modifiées du logiciel. Ce type de test complète les tests unitaires et les tests d'intégration.

2. Objectifs

- Implémentation d'un test de régression avec la librairie JUnit.

3. Mise en contexte pratique

Durant ce travail pratique, vous allez modifier et tester le mini projet du TP5 en suivant les instructions spécifiées dans la section suivante.

4. Travail à effectuer

1. La première étape est de modifier la méthode `build` de la classe `LinkedList` du projet étudié au TP5 de la façon suivante :

La méthode prendra les paramètres suivants :

- Une valeur entière qui indique l'opérateur utilisé : (**1** pour l'opérateur "**UNION**", **2** pour l'opérateur "**INTERSECTION**", **3** pour l'opérateur "**DIFFERENCE**" et **4** pour l'opérateur "**SYMMETRIC_DIFFERENCE**").
- Ensemble d'entiers : `val1`.
- Ensemble d'entiers : `val2`.
- Une valeur booléenne de tri : avec la valeur **true** pour trier l'ensemble de résultat en ordre croissant et la valeur **false** pour trier l'ensemble de résultat en ordre décroissant.

L'ensemble résultant ne doit pas contenir des entiers ayant la même valeur (les doublons).

Supprimer le code source des opérateurs non utilisés (***IS_SUBSET*** et ***IS_SUPERSET***).

Exemple 1:

```
suiteChaine.build(1, {1, 3}, {2, 3},1)
```

→ My List : {1, 3}, {2, 3}, {1, 2, 3}

Exemple 2:

```
suiteChaine.build (1, {1, 3}, {2, 3},0)
```

→ My List : {1, 3}, {2, 3}, {3, 2, 1}

2. En se basant sur le CFW du TP5, trouver toutes les classes qui doivent être testées de nouveau après le changement.
3. Pour toutes les classes à tester de nouveau, créer les cas de test pour chaque classe. Vous êtes libre de choisir le critère de test.

Note: Les cas de tests du TP5 peuvent être réutilisés dans ce TP lorsque pertinents.

5. Livrable à remettre, procédure de remise et retard

1. Une classe de test unitaire qui contient tous vos cas de tests. **NB: votre package doit être nommé TP6.**
2. Le projet java modifié.
3. Tableau des classes à tester de nouveau.

Veuillez envoyer vos fichiers dans une archive de type *.zip (et seulement zip, pas de rar, 7z, etc) qui portera le nom : **LOG3430_lab6_MatriculeX_MatriculeY.zip** tel que :
MatriculeX < MatriculeY.

Date limite pour la remise :

Groupe B1 & B2 : Lundi 3 décembre à 23h55

6. Barème de correction

Tableau de classes à tester de nouveau	5 points
Test avec JUnit : <ul style="list-style-type: none">- Ordre des tests selon les niveaux de tests- Test exécutable- Propreté, clarté et qualité des cas de test	15 points

Aucun travail ne sera accepté après lundi 3 décembre 23h55. Si votre dépôt ne respecte pas la nomenclature définie ci-dessus, 0.5 point de pénalité sera appliqué.