

# ANALYSE OBJET 420-1SY-SW

## PROJET FINAL AUTOMNE 2023

ÉVALUATION SYNTHÈSE **40%**

## Énoncé – Travail INDIVIDUEL

À partir d'un projet existant vous devez progressivement ajouter les demandes clients sur l'application console en C#. Ce projet devra respecter la méthodologie TDD vue en classe.

### Étape 1 – Création de votre projet

- Créer un dépôt GitHub ou Bitbucket (autres options non acceptés).
  - Utiliser le format suivant pour le nom de votre projet : 1SYA23\_XX\_Final
  - XX étant vos initiales. Exemple : 1SYA23\_MAD\_Final
- Faites un partage de ce projet à [madufresne@cshawwi.ca](mailto:madufresne@cshawwi.ca).
- Récupérer le projet de base et l'inclure à votre dépôt git.

### Étape 2 – Complétez une fonctionnalité

Des tests unitaires ont déjà été incorporer au projet. Vous devez donc compléter les fonctions adéquates pour les faire tous passer. (Note certains sont déjà fonctionnel)

### Étape 3 – Nouvelle fonctionnalité

Le système n'a été prévu que pour un langage actuellement. Mais après développement le besoin de gérer de nouveau langage est apparu. Il faut donc implémenter une traduction des éléments. Prendre note qu'il ne faut éviter de trop modifier les classes déjà existantes pour implémenter ce changement. Implémenter la commande « switch XX » expliquer dans le README.md

1. Ajouter au schéma les objets / classes nécessaires pour ajouter la gestion de cette fonctionnalité.
  - a. Note : il existe un patron de conception (Design Pattern) pour faire l'ajout de cette fonctionnalité tout en respectant les normes vues en classe.
  - b. Veillez à mettre une nouvelle version du schéma en version image (v1, v2, etc.) pour cumuler toutes les itérations du schéma dans votre git.
2. Planifier les tests unitaires qui vont avec cette nouvelle fonctionnalité.
  - a. Ajouter au plan de tests.
  - b. Programmer les tests.
3. Développer la fonctionnalité.

### Étape 4 - Questionnement

Quel(s) autres patrons de conception j'aurais pu utiliser pour simplifier mon code au niveau de la gestion des listes génériques (skills, langues et visions)? Répondez à cette question dans le fichier README.md

## Remise

- ❖ Vous devez faire votre remise via Github ou Bitbucket avant la date limite de la remise.
  - En cas de retard, les pénalités indiquées au plan de cours seront appliquées.
  - Par défaut je corrigerai la dernière version AVANT la date limite.
    - Si vous voulez que ce soit une version plus récente, vous devrez m'aviser avant la date limite.
  - En cas de retard, les pénalités indiquées au plan de cours seront appliquées.
- ❖ La date de remise est indiquée sur LÉA.
- ❖ Veuillez à respecter un langage académique en tout temps. Éviter les anglicismes lorsque possible.

## Critères

/ 54

| Critères                            | Valeur    |
|-------------------------------------|-----------|
| Partage Git début de projet         | 1         |
| Git - Fréquence des commits         | 1         |
| 2 - Qualité du code                 | 5         |
| 2 - Tests validés                   | 5         |
| 3 - Schéma : Norme UML et analyse   | 5         |
| 3 - Schéma : DP et justification    | 5         |
| 3 - TDD                             | 1         |
| 3 - Tests unitaires : planification | 5         |
| 3 - Tests unitaires : code (AAA)    | 10        |
| 3 - Code : Qualité du code          | 5         |
| 4 - DP(s) et justification(s)       | 5         |
| Respect normes README.md            | 6         |
| <b>Total</b>                        | <b>54</b> |