

# Gestion de projet

## Objectif

Le but de cette feuille est de :

1. commencer à organiser le projet
2. attribuer des rôles à chacun
3. utiliser des *issues* et des *milestones* pour affecter des tâches aux membres
4. créer des branches

## 1 Éditeur VS Code

### 1.1 Sur les machines de l'Université

Exécuter sur votre machine de l'Université la commande :

```
echo 'export PATH=/public/kn/code/bin:"$PATH"' >> ~/.bashrc
```

puis fermer et rouvrir vos fenêtres de terminal

### 1.2 Sur vos machines personnelles

— Installer l'éditeur Visual Studio Code

<https://code.visualstudio.com/>

— Installer Java (minimum version 8, la version courante est la 17)

— Installer l'utilitaire maven

<https://maven.apache.org/install.html>

— Installer l'extension **Extension Pack for Java** dans visual studio Code (aller sur l'icône ) et mettre **Extension Pack for Java** dans la barre de recherche

## 2 Premier Sprint

Dans le cadre d'une gestion *agile* des projets, ces derniers sont découpés en *sprint*. Un sprint est une période de temps relativement courte (1 semaine exemple) durant laquelle une certaine fonctionnalité doit être ajoutée.

Le prochain sprint devra se terminer le :

**jeudi 6 avril mars 2023, à 17h00**

Il devra consister en :

1. Ajout d'une classe **Task** complète (voir la section suivante)
2. Ajout d'une classe **MemTaskManager** complète
3. Ajout d'une interface **TaskManager** complète
4. Ajout d'une classe **MemTestTaskManager** permettant de tester le code

On utilisera pour le suivi de ces tâches :

— Une *milestone* (cf. démo)

- Une *issue* par élément à produire
- On ajoutera une dépendance entre les *issues*

Il convient évidemment de vous répartir les tâches à faire. Pour ce premier *sprint* les tâches ne sont pas vraiment parallélisables, chaque membre du groupe doit donc faire une tâche et les autres l'aident. Cela sera contrôlé par les auteurs des commits.

### 3 Import du projet

Récupère l'archive de projet Java disponible sur la page du cours. Décompresser l'archive et ajoute le répertoire `todo-list` ainsi que tout son contenu au dépôt `git`.

La structure de ce projet complexe sera expliquée dans les séances suivantes.

Ouvrir le projet dans VS Code (se placer dans le répertoire `todo-list` dans un terminal puis faire `code .` ou ouvrir ce répertoire depuis VS Code).

Localiser les 4 fichiers Java correspondant aux 3 classes et interfaces demandées (ils existent déjà mais ne sont que des squelettes).

Choisir une personne pour écrire le code de la première issue, et réfléchir à l'implémentation de la classe. Cette personne doit créer une branche. Procéder ensuite de même pour toutes les issues avec chaque fois une branche différente.

Une fois une branche terminée, ne pas faire un merge en ligne de commande, mais faire une merge request (demo).