

Mini-Projet : TO DO LIST APP

NB : En plus des fonctionnalités applicatives, la qualité du code sera prise en compte dans la correction de ce travail. De ce fait, vous devrez envoyer les livrables suivants :

- Un lien vers la vidéo de présentation du résultat d'au maximum dix (10) minutes (hébergée sur YouTube)
- Un lien vers le dépôt Git (public) contenant le code source

Les liens en question devront être envoyés au plus tard le **19 Novembre 2022 à 23h59mn59s** à **envoip@gmail.com**. L'objet du mail devra être **MP_TODOLIST_Groupe_X_DIT2**, X étant le numéro du groupe. Chaque groupe devra être constitué d'au plus deux (2) étudiants.

Le non-respect de ces contraintes rendra le travail irrecevable.

Description

L'objectif de ce mini-projet est de faire un programme sous Java qui permet à un utilisateur de pouvoir gérer la liste de ses tâches à faire.

Travail à faire

Pour réaliser ce travail, il vous est demandé de :

1) Créer la classe **Tache** sachant que celle-ci dispose des attributs suivants :

- **identifiant** : un entier identifiant chaque tâche de manière unique
- **dernierId** : un attribut de classe initialisé à un et qui contient le prochain **identifiant** à affecter à une tâche. Il est incrémenté à chaque instanciation d'une nouvelle tâche.
- **titre** : une chaîne de caractères décrivant la tâche
- **état** : une chaîne décrivant l'état de la tâche et ne doit prendre que les valeurs *EN_COURS*, *TERMINE* ou *PREVU*)
- **date de création** : de type Date (voir java.util.Date)

La classe devra également disposer des méthodes suivantes :

- **Tache (String titre)** : un constructeur qui initialise le titre avec le paramètre fourni, l'identifiant avec la valeur de **dernierId**, l'état à « PREVU » et la date de création à celle courante.

- **Tache (String titre, String etat)** : un constructeur qui initialise l'identifiant avec la valeur de **dernierId**, le titre et l'état avec les paramètres fournis, la date de création à celle courante.
- **Tache (String titre, String etat, Date dateCreation)** : un constructeur qui initialise l'identifiant avec la valeur de **dernierId**, les autres attributs avec les paramètres fournis
- Des **getters** et **setters** pour chacun des attributs d'instance
- La méthode **toString()** qui retourne la description textuelle d'une tâche

2) Créer la classe **GestionnaireTache** disposant des attributs suivants :

- **taches** : un tableau d'objets de la classe **Tache**
- **nombreTache** : le nombre de Tache qu'il y'a effectivement dans le tableau

Cette classe dispose également des méthodes suivantes :

- **GestionnaireTache (int nombreMax)** : un constructeur qui initialise *nombreTache* à zéro, puis crée un tableau initialement vide pouvant contenir jusqu'à *nombreMax* objets de classe *Tache*.
- **boolean ajouter (Tache tache)** : ajoute la *Tache* dans le tableau si ce dernier n'est pas encore plein (*nombreTache* a atteint la taille du tableau). Retourne *true* ou *false* suivant que l'ajout se soit effectué ou non.
- **boolean modifier (Tache tache)** : Modifie la *Tache* dont l'identifiant correspond à celui de la tâche fourni en paramètre. La modification se fera en considérant les valeurs des attributs du paramètre. Retourne *true* ou *false* suivant que la mise à jour se soit effectuée ou non.
- **boolean supprimer (int id)** : Supprime la *Tache* dont l'identifiant correspond au paramètre. Retourne *true* ou *false* suivant le succès ou non de la suppression.
- **Tache[] lister()** : retourne la liste des tâches effectivement présentes dans le tableau.
- **Tache[] lister(String etat)** : retourne la liste des tâches effectivement présentes dans le tableau et qui sont dans l'état spécifié (en cours, terminée ou prévue)

3) Créer une classe **GestionTache** qui dispose d'un programme principal (*main*) permettant à l'utilisateur d'effectuer des opérations de base sur les tâches (ajout, modification, suppression, liste de toutes les tâches, liste des tâches en cours, terminées, ou prévues, etc.). Un menu devra toujours lui être affiché afin qu'il puisse choisir l'opération qui l'intéresse.