Travail Pratique

- Ce TP doit être réalisé en **binôme** et se déroule sur **plusieurs séances**.
- A la fin de la première séance, vous devrez remettre un **rapport** expliquant votre conception: identification des classes (incluant leurs types: interfaces, énumérations ou classes abstraites), méthodes et attributs. Un schéma des classes précisant les relations entre elles doit être inclus dans ce compte rendu.
- La qualité du rapport est prise en considération.

My desktop planner

En discutant avec votre camarade de première année à l'ESI, il vous confie sa difficulté à s'organiser: entre les cours à l'école, ses devoirs, ses TP et les activités extra-académiques qu'il fait. Il se trouve perdu et impuissant pour tout accomplir à temps. En rentrant chez vous, vous retrouvez votre maman à la maison. Après lui avoir demandé ce qu'elle avait préparé à manger et en lui faisant remarquer que vous n'aimez pas trop ce qu'elle vous proposait, elle a commencé à se plaindre du manque de temps pour faire toutes ses tâches quotidiennes tout en satisfaisant les désirs culinaires de la famille. Vous avez eu l'impression d'entendre une autre version du discours de votre camarade mais cette fois-ci de la part d'une autre personne. Afin d'apporter de l'aide aux personnes de votre entourage et pourquoi pas à vous même, vous décidez de réaliser une application bureau de planification de tâches.

Après réflexion et discussions avec votre maman et votre camarade, vous avez pu établir une liste préliminaire de fonctionnalités devront être assurées par votre future application bureau que vous baptisez "*My desktop planne*r" .

L'application peut être installée sur une machine utilisée par plusieurs utilisateurs. Chaque utilisateur est identifié par son pseudo, et peut introduire une tâche sur son calendrier personnel en précisant son nom, sa durée, sa priorité (Low, Medium, High) et éventuellement une date limite (*deadline*). L'utilisateur peut également classer la tâche dans une catégorie (studies , work , hobby, sport, health, etc.) et affecter une couleur aux tâches de même catégorie. L'application offre aussi la possibilité de créer un projet défini par un nom, une description et un ensemble de tâches.

L'utilisateur peut planifier une tâche manuellement en choisissant un créneau libre d'une journée sur le calendrier, ou laisser son application la planifier. Pour que l'application puisse effectuer la planification, l'utilisateur spécifie, en premier, ses créneaux libres durant lesquels les tâches peuvent être programmées (exemple: 13h-15h et 18h-22h pour une journée donnée) ainsi que la durée minimale d'un créneau (30 minutes par exemple).

Lors de la planification, un créneau libre peut être décomposé. Autrement dit, l'ajout d'une tâche à un créneau libre conduira à sa décomposition en deux créneaux: celui qui sera occupé par la tâche ajoutée, et un créneau libre ayant la durée restante. A titre d'exemple, supposons que l'utilisateur

dispose d'un créneau libre de 3 heures et qu'il souhaite introduire une tâche d'une heure. Ce créneau de 3 heures sera décomposé en deux: un créneau d'une heure qui sera planifié pour la tâche ajoutée, et un autre de deux heures qui restera libre. Notez que la durée minimale du créneau doit toujours être respectée lors de la décomposition sinon celle-ci est interdite. Si nous reprenons l'exemple précédent avec un créneau libre de trois heures et une tâche de 2h45, le temps libre restant étant inférieur à la durée minimale de 30 minutes, la décomposition est impossible et le créneau de trois heures sera entièrement alloué à la tâche.

L'utilisateur peut demander à l'application de lui planifier une tâche ou un ensemble de tâches déjà introduites. Il peut également fixer une période pendant laquelle il souhaite planifier ses tâches (allant d'une date début à une date fin, par exemple: du 23 mars au 9 avril 2023). Si aucune période n'est fixée, les tâches seront planifiées dans les créneaux libres de la journée, puis celles qui suivent, en respectant les dates limites de chacune des tâches introduites, sinon l'application informe l'utilisateur que la tâche n'a pas pu être planifiée et aura l'état "unscheduled".

Une tâche peut être simple ou décomposable. Les tâches simples ne peuvent être planifiées que dans un seul créneau, et peuvent être périodiques (planifiées tous les n jours). Cette périodicité doit être spécifiée par l'utilisateur, sinon la tâche n'est planifiée qu'une fois. Une tâche décomposable peut être planifiée en plusieurs créneaux jusqu'à atteindre la durée prévue. Ses sous-tâches auront le même nom que la tâche décomposée auquel sera concaténée le numéro de la sous tâche. Par exemple l'utilisateur introduit la tâche décomposable dont le nom est "révision" d'une durée de 4 heures. Si le système trouve un créneau dont la durée est supérieure ou égale à quatre heures, elle sera planifiée. Mais dans le cas où aucun créneau libre de cette durée n'est trouvé, le système peut proposer à l'utilisateur de la décomposer selon les créneaux disponibles. Supposons que nous disposons d'un créneau de 2 heures et un autre de 2h30mn. Le système décompose donc la tâche "révision" en deux sous-tâches "révision 1" et "révision 2" pour les planifier dans les deux créneaux qui sont disponibles (le second créneau de 2h30 sera décomposé en un créneau de 2h qui sera planifié pour la sous tâches "révision 2" et un autre libre de 30mn). Évidemment, l'utilisateur peut valider ou refuser cette proposition. Dans le cas de validation, il peut changer le nom des sous-tâches.

Pour se motiver, l'utilisateur peut introduire pour chaque jour de son planning, l'état de réalisation des tâches programmées (non réalisée: *notRealized*, complétée: *completed,* en cours : *in progress,* annulée*: cancelled,* reportée: *delayed*). L'état de réalisation d'une tâche décomposée ou d'un projet est évalué selon l'état de réalisation des sous-tâches qui les composent. Dans le cas où une tâche n'est pas complètement réalisée (*in progress)* et que l'utilisateur souhaite encore la re-planifier, le système lui demande la durée supplémentaire nécessaire pour l'accomplir, et le nouveau *deadline* si jamais elle en possède un. Il en est de même dans le cas du report d'une tâche non réalisée.

L'utilisateur peut aussi fixer un nombre minimal de tâches à réaliser par jour. Si ce nombre est atteint, le système le félicite par un message d'encouragement. Dans le cas où ce nombre est atteint cinq (5) fois consécutives, un badge "Good" lui est associé. S'il réussit à avoir ce badge trois(3) fois, il obtient le badge "VeryGood", qui peut devenir "Excellent" si le badge VeryGood est obtenu trois (3) fois.

Un exemple de scénario d'exécution du système peut être vu comme suit:

- L'utilisateur s'authentifie en spécifiant son pseudo.
- Le système lui affiche le calendrier, et lui demande s'il veut fixer une période pour son planning. Bien sûr, l'utilisateur ne peut pas choisir une date début de la période antérieure à la date du jour.
- Le système lui demande de fixer les créneaux libres de chaque jour de la période définie (par exemple: 18h-22h pour tous les jours de la période choisie, 18h-21h pour le 21/03/2023, etc.).
- L'utilisateur choisit de planifier manuellement une tâche. Il introduit toutes ses informations et précise le créneau où elle sera effectuée. Il peut bloquer ce créneau pour cette tâche s'il souhaite qu'il ne soit pas modifié en cas de re-planification de l'ensemble des tâches..
- L'utilisateur introduit un ensemble de tâches à faire en précisant leurs priorités, leurs deadlines, et si elles sont décomposables ou pas et demande au système de les planifier. Le système classe les tâches selon ces critères, puis propose un planning que l'utilisateur peut valider, annuler ou modifier. Si des tâches n'ont pas pu être programmées dans les créneaux disponibles de la période fixée, l'application informe l'utilisateur et lui demande s'il souhaite les reporter dans les jours suivant cette période. Si oui, la période sera étalée jusqu'à ce que toutes les tâches soient programmées, sinon elles seront sauvegardées avec l'état "unscheduled". L'utilisateur peut les supprimer ou les programmer plus tard dans une autre période.
- L'utilisateur introduit une autre tâche et demande au système de la planifier avant une date limite. Le système ne trouve pas un créneau libre satisfaisant la priorité et le deadline de cette tâche dans la période définie. Il demande alors à l'utilisateur s'il se souhaite faire une mise à jour de son planning en considérant cette nouvelle tâche. Dans ce cas, les créneaux sont considérés à l'état initial (libre) sauf ceux bloqués explicitement par l'utilisateur, et un nouveau planning est proposé en prenant en compte cette nouvelle tâche.
- En fin de journée, l'utilisateur introduit l'état de réalisation de ses tâches de la journée. Dès qu'il atteint trois(3) tâches complétées (nombre fixé en paramètre par défaut et que l'utilisateur peut modifier), le système le félicite.
- L'utilisateur peut à tout moment avoir des statistiques sur :
  - l'état de réalisation des tâches de la journée
  - l'état de réalisation d'un projet.
  - le nombre/type de badges gagnés.
  - son rendement journalier, égale au rapport entre le nombre de tâches complétées au nombre de tâches prévues.
  - la moyenne de son rendement journalier depuis la date début de la période actuelle
  - le nombre de fois où il a eu un encouragement du système (le nombre de jours où le nombre de tâches réalisées a dépassé le seuil minimal qu'il a fixé).
  - le jour où il a été le plus rentable.
  - la durée de temps passée sur des tâches d'une catégorie pour mesurer sur quoi il passe plus de temps.
- l'utilisateur peut consulter aussi l'historique des anciens plannings : pour chaque planning il peut voir le nombre/type de badges gagnés, le nombre de tâches/projets complétés.

Travail demandé

1. Proposez une modélisation orientée objet pour ce jeu en précisant les différentes classes, leurs attributs et leurs méthodes
    a. Indiquez clairement les types de classes (abstraites, interfaces, énumération, exception) et les liens entre elles (héritage, implémentation, utilisation) sur un diagramme.
    b. Détaillez sur un tableau chacune des classes (attributs, méthodes) selon le format suivant :

Nom classe 1 /*et son type*/	/*extends … implements…*/
Liste attributs	
Mode d'accès Type_attribut nom_attribut	Description
...	…
Liste des méthodes	
Mode d'accès Signature méthode	Rôle
…	…

2. Implémentez votre solution en langage Java à travers une interface graphique permettant à l'utilisateur d'interagir avec l'application.
3. Testez votre application.

Remarque

Nous souhaitons maintenir le développement de cette application dans le futur. De ce fait, votre programme doit respecter quelques spécifications :

- Le programme doit être lisible ; la lisibilité concerne:
    - Le respect des conventions d'encodage Java.
    - Les commentaires : il est conseillé d'utiliser le style JavaDoc afin de faciliter la génération de la documentation.
    - L'interface graphique et le noyau doivent être séparés.
- Les valeurs constantes doivent être facilement modifiables.

Le meilleur travail respectant les fonctionnalités demandées sera récompensé