

Gestionnaire de tâches (To-Do List) en Rust

Vous devez créer un gestionnaire de tâches simple en Rust qui permet à l'utilisateur d'ajouter, afficher et supprimer des tâches. Les tâches peuvent être décrites par une chaîne de texte. L'importance sera définie à l'aide d'une énumération avec au moins trois niveaux : Faible, Moyenne et Élevée.

Fonctionnalités attendues :

1. **Ajouter une tâche** : L'utilisateur peut ajouter une nouvelle tâche en spécifiant une description et une importance. Chaque tâche doit avoir un identifiant unique.
2. **Afficher les tâches** : L'utilisateur peut afficher la liste de toutes les tâches avec leurs identifiants descriptions et importances correspondantes.
3. **Supprimer une tâche** : L'utilisateur peut supprimer une tâche en spécifiant son identifiant.

Contraintes :

1. Utilisez des structures de données appropriées pour stocker les tâches.
2. Gérez les erreurs potentielles, telles que la tentative de suppression d'une tâche inexistante.
3. Le programme doit afficher un menu interactif permettant à l'utilisateur de choisir parmi les différentes actions (ajouter, afficher, supprimer, quitter).

Exemple d'utilisation :

```
$ ./gestionnaire_taches
```

1. Ajouter une tâche
2. Afficher les tâches
3. Supprimer une tâche
4. Quitter

```
Choisissez une option : 1
```

```
Entrez la description de la tâche : Faire les courses
```

```
Choisissez l'importance (1: Faible, 2: Moyenne, 3: Élevée) : 2
```

```
Tâche ajoutée avec succès !
```

```
Choisissez une option : 2
```

```
Liste des tâches :
```

1. Faire les courses | (Importance : Moyenne)

Choisissez une option : 3

Entrez l'identifiant de la tâche à supprimer : 1

Tâche supprimée avec succès !

Choisissez une option : 2

Liste des tâches : (vide)

Choisissez une option : 4

Au revoir !

Conseils :

- Pensez à utiliser des structures.
- Utilisez des vecteurs pour stocker les tâches.
- Explorez la gestion des erreurs en Rust pour une expérience utilisateur robuste.
- Structurez votre code en fonctions pour améliorer la lisibilité.

Bonne programmation ! Si vous avez des questions ou si vous avez besoin d'aide pour l'implémentation, n'hésitez pas à demander.