Machine Translated by Google

Évaluation

Dans cette tâche d'évaluation, nous allons implémenter une application d'annuaire téléphonique en Java.

Une application Répertoire stocke les informations de contact de vos amis et leur numéro de téléphone et vous aide à rechercher dans toutes les données, y compris le numéro de téléphone ainsi que le prénom et le nom.

Aperçu

Le répertoire contient des entrées de contacts avec le prénom et le nom d'une personne, un ou plusieurs numéros de téléphone et une catégorie attribuée à chaque numéro de téléphone.

Voici un exemple de deux entrées de répertoire téléphonique ; chaque entrée est séparée par une ligne de 3 points (---) :

```
Prénom : Harry
Nom de famille : Potter
MOBILE          : +374 55 123456
BUREAU          : +374 10 875803
MAISON          : +374 10 689651
---
Prénom : Sheldon
Nom de famille : Cooper
MOBILE : +374 77 975892
BUREAU          : +374 11 573203
BUREAU          : +374 11 573204
---
Prénom : Jon
Nom de famille : Snow
MOBILE          : +374 91 543216
```

La catégorie d'un numéro de téléphone peut être MOBILE , BUREAU ou DOMICILE. Notez que nous NE limitons PAS une personne à un seul numéro par catégorie et que nous NE demandons PAS à une personne d'avoir un numéro de téléphone pour chacune des catégories. Cependant, nous limitons chaque personne à avoir au maximum 5 numéros de téléphone dans le répertoire.

L'application Répertoire fonctionne avec des données fournies dans un fichier texte (un exemple de fichier texte est inclus dans la description de la tâche). Par conséquent, l'insertion d'entrées dans le répertoire ne fait pas partie de l'application Répertoire et s'effectue simplement en écrivant dans un fichier texte et en le plaçant à côté de votre application.

L'application commence par demander une action à l'utilisateur via la ligne de commande :

```
*** Application de recherche dans le répertoire téléphonique ***

Choisissez votre action :
1 : Rechercher par nom
2 : Recherche par numéro
> 1
```

Machine Translated by Google

Entrez le prénom ou le nom :
> Har

Enregistrement(s) trouvé(s) :
Prénoms : Harry
Noms de famille : Potter
MOBILE : +374 55 123456
BUREAU : +374 10 875803
MAISON : +374 10 689651

*** Sortie ***

Comme vous l'avez remarqué, le programme recherche soit par prénom, soit par nom et est capable de rechercher par jetons partiels (par exemple, Har est éligible pour Harry et Harrison). Vous pouvez spécifier une longueur minimale de 3 lettres pour la recherche.

S'il existe plusieurs enregistrements éligibles au jeton fourni (« Har » est un jeton de recherche), tous les enregistrements correspondants doivent être affichés.

Il devrait également être possible de rechercher des noms en fournissant un numéro de téléphone :

*** Application de recherche dans le répertoire téléphonique ***

Choisissez votre action :
1 : Rechercher par nom
2 : Recherche par numéro
> 2

Entrez le numéro de téléphone:
> +37411573203

Enregistrement trouvé :
Prénom : Sheldon
Nom de famille : Cooper
MOBILE : +374 77 975892
BUREAU : +374 11 573203
BUREAU : +374 11 573204

*** Sortie ***

Étant donné qu'un numéro de téléphone ne peut appartenir qu'à une seule personne, la recherche par numéro de téléphone ne renverra qu'un seul résultat au plus.

Dans cet exemple, nous avons fourni une correspondance exacte (+37455123456) pour le numéro de téléphone. Si vous pouvez implémenter le programme de manière à ce qu'il puisse également rechercher d'autres variantes du numéro de téléphone (par exemple, « 055 123456 », ou « 123456 », ou « 055123456 ») cela comptera comme une fonctionnalité supplémentaire et sera évalué en conséquence.

Mise en œuvre

Machine Translated by Google

- Votre implémentation doit utiliser les concepts de POO et de modèles de conception chaque fois que possible. Évitez d'écrire l'intégralité du code dans une seule grande classe.
- Utiliser les énumérations Java afin d'implémenter les catégories de numéros de téléphone (MOBILE, BUREAU, DOMICILE).
- Gestion des erreurs : comme l'application accepte les entrées à partir de la ligne de commande, il est possible que l'utilisateur fournisse une entrée incorrecte. Votre programme doit gérer de tels scénarios soit en mettant fin au programme, soit en donnant une autre chance à l'utilisateur de répéter l'étape et de fournir une entrée correcte.
- Fichier d'entrée : un fichier nommé data.txt contient les entrées du répertoire téléphonique qui doivent être utilisées par l'application. Cependant, vous ne devez pas supposer que le contenu du fichier sera toujours correct. Il est possible que quelqu'un fournisse un fichier incorrect (un fichier qui ne contient pas de données au bon format). Par conséquent, vous devez effectuer une validation de base lors du traitement du fichier data.txt et afficher une erreur si le fichier fourni est incorrect.
- Utilisez des commentaires pour expliquer vos pensées chaque fois que vous le trouvez utile.
- Commencez par mettre en œuvre la version minimale exploitable, puis concentrez-vous sur chaque partie pour l'améliorer et la rendre encore meilleure. (Faites en sorte que cela fonctionne, puis rendez-le beau, puis, si vous le devez vraiment, faites-le rapidement. Dans 90 % des cas, si vous le rendez beau, il sera déjà rapide. Alors, vraiment, faites-le simplement beau ! – Joe Armstrong)

Algorithme de recherche

Vous êtes libre d'implémenter la partie recherche de l'application selon la méthode de votre choix. Nous attendons de vous que vous décriviez le temps d'exécution de l'algorithme que vous utilisez pour rechercher les entrées de contact par nom ou par numéro de téléphone.

POINT BONUS. Il est recommandé d'essayer d'utiliser une structure de données qui rendra le processus de recherche plus rapide et plus efficace. Encore une fois, vous devez décrire et expliquer le temps d'exécution de l'algorithme.

DevOps

Écrivez un script shell qui supprime toutes les classes compilées et compile les fichiers sources Java.
Étendez le script shell qui exécutera une application Java que vous avez développée et qui la traitera en écrivant la sortie dans de nouveaux fichiers.

Utilisez votre script créé. Créez un Dockerfile dans lequel la commande du point d'entrée exécute le script mentionné. Nommez votre script docker.entrypoint.sh et créez un conteneur Docker, qui utilisera votre image Docker nouvellement créée. Ainsi, lorsque vous exécutez le conteneur Docker, cette sortie doit être la même que celle de votre script.

Question bonus

Structures de données et algorithmes

Machine Translated by Google

Implémentez une fonction récursive qui détermine si le tableau donné est trié par ordre décroissant ou non. Écrivez des commentaires pour expliquer votre code et indiquez la complexité de la fonction.

Entrée : tableau de nombres. Sortie : booléen.
Exemples : IsDescending([6, 4, 3, 3, 1]) = true, IsDescending([6, 4, 3, 4, 1]) = false

Les boucles sont interdites, vous devez utiliser la récursivité.