

Module et IO

1 Exercice 1

Une librairie enregistre les contacts de ses clients dans un fichier sous le format `<email> <numéro de téléphone> <adresse> <type> <nom>`. Où `<email>`, `<numéro de téléphone>` et `<adresse>` sont des chaînes de caractères sans espace. `<email>` a au maximum 50 caractères. `<numéro de téléphone>` a au maximum 9 caractères. Quant à `<adresse>` elle a au maximum 100 caractères. `<type>` est le caractère I ou E (I est mis pour individu et E est mis pour entreprise) `<nom>` est une chaîne pouvant être constituée de plusieurs mots et de taille maximale 150 caractères. Un exemple de fichier est le suivant.

```
d.nkalla@idea-cm.club 691179152 Rue_lottin_Same I Nkalla Ehawe
epoto.douglas@yahoo.fr 685656699 Bepanda_tonnere I Epoh Douglas Landry
info@iadnet.cm 699050505 Rue_lottin_Same E Africa Development Network
iservices@iservices.cm 671747568 Dragage_Yaounde E iServices
contact@iadnet 699050505 Rue_lottin_Same E Africa Development Network
ossambo.ledrole@paradi.qlq 660000001 Rue_des_Palmiers I Ossambo Le Drole
maurice.tchueto@inivers.com 662252404 Payage_Batoufam I Tchuente Maurice Pierre
maurice.tchueto@inivers.com 675057317 Payage_Batoufam I Tchuente Maurice Pierre
d.nkalla@idea-cm.club 691179152 Rue_des_manguiers I Nkalla Ehawe didier Junior
```

Le nombre d'enregistrement dans le fichier n'est pas connu a priori. Deux enregistrements sont identiques s'ils ont la même adresse email et le même numéro de téléphone. Par exemple:

`<d.nkalla@idea-cm.club 691179152 Rue_lottin_Same I Nkalla Ehawe>` et `<d.nkalla@idea-cm.club 691179152 Rue_des_manguiers I Nkalla Ehawe didier Junior>` sont identiques

1. Ecrire en Haskell la fonction `readFileContent` qui prend en paramètre le nom du fichier et restitue son contenu dans une structure de donnée appropriée qu'on définira avant la définition de la fonction.
2. Ecrire la fonction `displayFileContent` qui permet d'afficher le contenu du fichier sans les doublons (les doublons sont les enregistrements identiques)
- 3- Les doublons dans le fichier sont considérés comme des mises à jour du contact de départ. Le dernier enregistrement est le plus à jour. Ecrire la fonction `noRepeat` qui permet d'éliminer les doublons dans les enregistrements en ne gardant que le plus récent (dernier enregistrement du même contact dans l'ordre de lecture du fichier)
3. Ecrire la fonction `displayById` qui prend en paramètre une chaîne de caractères et affiche tous les enregistrements ayant pour mail ou numéro de téléphone la chaîne spécifiée
- 5- Ecrire la fonction `displayByAddress` qui prend en paramètre une chaîne de caractères et affiche tous les enregistrements ayant pour adresse la chaîne spécifiée

6- Ecrire la fonction `validateRegistration` qui prend en entrée un enregistrement et retourne `true` si l'adresse email et le numéro de téléphone sont valides; et `false` si non.

Une chaîne de caractère est considérée comme une adresse email valide si:

a- la chaîne ne commence pas par le caractère `@`

b- Contient exactement une seule fois le caractère `@`

c- Contient exactement une seule fois le caractère `.` après le caractère `@`. Le caractère `@` et `.` sont séparés d'au moins un caractère.

Un numéro de téléphone est valide s'il possède exactement neuf(9) caractères numériques et s'il commence par 69 ou 67 ou 68 ou 65 ou 66

Indication: On pourra d'abord écrire une fonction qui valide les numéros de téléphone, puis une seconde qui valide les adresses email et combiner les deux fonctions.

7- Ecrire une fonction qui enregistre dans un fichier `nome contacts.txt` tous les enregistrements valides

2 Exercice 2

Gestion de conférence

Développer une application de gestion de conférence en langage Haskell.

2.1 Conférences

Le système de gestion est implémenté via un Module `SystemManagement`. Les conférences sont définies par la fonction `newConference`, qui reçoit comme paramètres le nom, le lieu de déroulement, l'année, les dates de début et de fin (mois et jour, dans le format `mm/jj`) et le montant de l'inscription. La fonction renvoie l'objet `Conference` créé. Les conférences sont identifiées de façon univoque par un acronyme (en majuscules) composé des lettres initiales des mots qui composent le nom, suivi de l'année (par exemple, pour une conférence en 2018 intitulée `Wireless sensor networks`, l'acronyme serait `WSN2018`). Le type `Conference` fournit des fonctions appropriées pour accéder aux informations susmentionnées. Cherchant à définir deux conférences avec le même nom et la même année, la fonction `newConference` se limite à la mise à jour des informations de la première.

Grâce à la fonction `addSponsor`, un sponsor peut être ajouté à la conférence. En essayant d'ajouter un sponsor déjà spécifié pour la conférence, la fonction n'a aucun effet. La fonction `ListSponsor` renvoie une collection contenant les sponsors dans l'ordre dans lequel ils ont été ajoutés. Un sponsor est caractérisé par une chaîne de caractères.

Grâce à la fonction `listConferences` du module `SystemManagement`, il est possible d'obtenir une collection contenant toutes les conférences gérées, triées par nom et année. Une variante de cette fonction reçoit comme paramètre deux années (par exemple, 2015 et 2018) et permet de restreindre la collection retournée à la période indiquée seulement, y compris les limites.

2.2 Les utilisateurs

Les utilisateurs du système sont identifiés de manière unique par l'adresse e-mail et sont représentés par des instances du type `User`. Pour définir un nouvel utilisateur, vous pouvez utiliser la fonction `newuser` du Module `SystemManagement`. La fonction reçoit comme paramètres le prénom, le nom, l'organisation et l'adresse électronique (e-mail) de l'utilisateur. Dans le cas où un utilisateur avec le même courrier électronique est déjà présent dans le système, une exception de type `UserDuplicatedException` doit être déclenchée.

La fonction `searchUser` reçoit une chaîne en paramètre et renvoie tous les utilisateurs trouvés pour lequel l'un des attributs contient, partiellement ou en totalité, la chaîne passée en paramètre (une liste vide si aucun utilisateur n'est trouvé). La fonction `listUser` renvoie une collection qui contient tous les utilisateurs du système dans l'ordre alphabétique (par nom et prénom).

2.3 Œuvres et auteurs

Les utilisateurs peuvent, en tant qu'auteurs, soumettre des travaux à la conférence pour examen et présentation ultérieure aux participants. Les œuvres sont représentées par le type `Work` et caractérisées par un titre, un type et une liste d'auteurs. Deux types d'œuvres sont possibles, à savoir des articles et des affiches (Poster), représentés par les types `Article` et `Poster`.

Pour soumettre un travail à une conférence, utilisez la fonction `submit` du module `SystemManagement`. La fonction reçoit comme paramètres l'acronyme de la conférence, le titre de l'œuvre, le type (article ou poster) et l'adresse e-mail de l'utilisateur qui agit en tant que premier auteur. La fonction renvoie l'identifiant attribué au travail, obtenu en combinant l'acronyme de la conférence, suivi du caractère '-' et d'un code progressif unique pour la conférence spécifique attribuée à partir de 1 (par exemple `WSN2018-1`). Supposons, pour simplifier, ne pas avoir à gérer les soumissions en double. Dans le cas où l'acronyme ou le courriel n'existe pas, la fonction n'a aucun effet. Le type `Work` fournit des fonctions pour accéder au titre et à l'identifiant affecté. La fonction `searchWork` du module `SystemManagement` reçoit l'ID du travail en tant que paramètre et renvoie les conférences à laquelle le travail a été soumis.

Pour ajouter d'autres auteurs à une œuvre (en plus du premier, spécifié au moment de la soumission), nous utilisons la fonction `addAuthor` du module `SystemManagement`, qui reçoit comme paramètre l'identifiant de l'œuvre et l'adresse mail de l'utilisateur à ajouter à la liste des auteurs. Dans le cas où l'œuvre ou l'utilisateur n'existe pas, la fonction n'a aucun effet. La fonction `listAuthors` du module `SystemManagement` reçoit l'ID d'une œuvre et renvoie une collection contenant tous les auteurs de l'œuvre dans l'ordre d'insertion.

2.4 Révisions et programmes

Pour chaque œuvre, un certain nombre de révisions doit être collectées des utilisateurs du système, sous la forme d'un commentaire et d'une notation. Pour créer une nouvelle révision, utilisez la fonction `newRevision` du module `SystemManagement`, qui reçoit l'identifiant de l'œuvre, l'email de l'utilisateur, le commentaire et la note (sur une échelle de 1 à 5). Si l'id ou l'email n'est pas défini ou si la note est en dehors de l'échelle indiquée, la fonction déclenche une exception `RevisionRefusedException`.

La fonction `CalculatePointMedium` de la classe `Work` vous permet de calculer la note moyen d'une œuvre spécifique.

La fonction `GenerateProgram` du module `SystemManagement` renvoie une chaîne contenant la liste de tous les articles soumis à la conférence dont l'acronyme est passé en paramètre, suivie de la liste de tous les posters. Les articles et les poster sont présentés par ordre de notation décroissant. Chaque ligne contient le titre d'un article / poster et se termine par une virgule, sauf la dernière, terminée par un point.

2.5 Inscriptions

Le système gère également les enregistrements lors de conférences. En particulier, pour inscrire un utilisateur à une conférence, on utilise la fonction `registration` du module `SystemManagement`, qui reçoit comme paramètres l'email de l'utilisateur à enregistrer et l'acronyme de la conférence. Dans le cas où l'utilisateur ou l'acronyme n'est pas défini, la fonction n'a aucun effet. La fonction `listRegistration` reçoit l'acronyme d'une conférence en tant que paramètre et renvoie une collection contenant les utilisateurs abonnés dans l'ordre d'enregistrement. Grâce à la fonction `revenu` du module `SystemManagement`, il est possible de calculer le montant encaissé pour la conférence dont l'acronyme est passé en paramètre. Le montant encaissé est considéré comme la somme des frais payés lors de l'inscription.