TP : Prise en main de Swift : compilation, entrées-sorties, itérateurs, etc...

Vincent Berry, Christophe Fiorio, Elisa Henrion-Gueneau

1 Introduction

L'objectif de ce TP est de prendre en main la programmation swift. Il vous préparera également à lire et écrire des fichiers textes, à compiler un programme et à faire des modules.

À l'issu de ce TP, vous serez armés pour réaliser le projet.

Mais d'abord, il est nécessaire de configurer votre environnement.

2 Prise en main de la programmation Swift

Normalement vous avez déjà du écrire des fonctions en swift en utilisant le playground de l'Ipad. Nous allons maintenant voir comment programmer en swift en utilisant le compilateur en ligne de commande. Ce qui suit est réalisable sur Mac mais aussi sous Ubuntu.

* Exercice 1 Premières fonctions

À l'aide de votre éditeur de texte favori, ouvrez un fichier `tpswift.swift` et implémentez les fonctions suivantes de la feuille 3 de TD d'algorithmique :

— `sommeEntiers` de l'exercice 1

— `estPremier` et `sansDiviseurs` de l'exercice 5

Vérifiez votre syntaxe en compilant votre fichier. Pour cela exécutez la commande : `swiftc tpswift.swift`.

* Exercice 2 Résultat des fonctions

En fin de votre fichier `tpswift.swift`, rajoutez des instructions pour tester vos fonctions et afficher les résultats. Les paramètres seront fixés dans le programme pour l'instant.

Compilez à nouveau votre fichier et exécutez le pour vérifier vos résultats.

Pour récupérer une entrée au clavier de l'utilisateur, il faut utiliser la fonction `readLine()`. Cette fonction renvoie une chaine de caractères. Il faudra donc convertir le résultat de la saisie et vérifier si la conversion s'est bien passée.

Modifiez votre fichier `tpswift.swift` pour faire en sorte, lors des instructions de test du résultat de vos fonctions, de demander à l'utilisateur les données d'entrée avant d'exécuter la fonction.

Compilez et testez le résultat.

3 Entrée/sortie de fichiers texte

Pour lire ou écrire dans un fichier texte, il faut pouvoir :

— accéder à un fichier donner

— écrire un texte dans ce fichier, ou lire le texte contenu dans ce fichier

Le type `String` offre des fonctions pour créer une chaîne à partir du contenu d'un fichier, et permet également d'écrire dans un fichier. Tout cela peut se faire à partir d'une Url décrivant le chemin du fichier.

3.1 Création d'Url à partir d'un chemin

Pour créer une Url décrivant un chemin de fichier, 2 solutions s'offre à vous :

1. Utiliser la classe
   `FileManager` afin de récupérer un chemin standard

`FileManager.default.urls(for: .documentDirectory, in: .userDomainMask).first` permet de récupérer une URL du répertoire standard de stockage des documents.

la fonction `.appendingPathComponent("nom du fichier")` permet d'ajout à l'URL le nom du fichier

2. Utiliser la classe `URL` avec la classe `FileManager` pour construire une URL à partir d'une chaîne de caractères du chemin du fichier

   `FileManager.default.currentDirectoryPath` permet de récupérer une chaine de caractères décrivant le chemin courant ; il suffit d'y concaténer le nom du fichier.

   `URL(fileURLWithPath: String)` permet de créer une URL à partir d'une chaine de caractères.

Une fois l'URL créée, on peut écrire ou lire le fichier correspondant.

Pour écrire dans le fichier, il suffit d'utiliser la fonction `.write(to: URL, atomically: false, encoding: .utf8)` du type `String` pour écrire dans le fichier le contenu de la chaîne de caractère.

Pour lire le fichier, il suffit de créer une chaîne de caractères à partir de l'URL : `String(contentsOf: URL, encoding: .utf8)`.

Si l'on veut récupérer un tableau des composants d'une chaîne de caractères du contenu du fichier, par exemple, le tableau des lignes, il suffit d'utiliser la fonction `components(separatedBy:)` du type `String`. Par exemple `var lignes = fileContent.components(separatedBy:"\n")` récupère dans la variable tableau `lignes` les lignes du texte.

4 Dictionnaires d'indicatifs d'aéroport

Remarque liminaire : les fonctions demandées ne sont pas spécifiées dans les exercices, c'est à vous de le faire avant de les coder.

* Exercice 3 Type indicateurs aéroport

Définissez un type valeur `IndicatifAeroport` qui comprend, pour l'instant, comme seule fonction l'initialisation à partir des données suivantes :

MPL Montpellier

DUB Dublin

CDG Paris Charles-de-Gaule

ORY Paris Orly

YYZ Toronto

LHR London

LAX Los Angeles

* Exercice 4 Sauvegarde des données

Rajoutez au type une fonction `save` de sauvegarde des données dans un fichier texte au format CSV. Le nom de fichier est passé en paramètre et le fichier est sauvegardé dans le répertoire courant ;

* Exercice 5 Lecture des données

Rajoutez une fonction d'initialisation qui crée une instance à partir d'un nom de fichier. Le fichier est considéré comme étant dans le répertoire courant.

* Exercice 6 Modification des données

On a fait une erreur dans la liste précédente : `LHR` est l'indicatif de l'aéroport de *London Heathrow* et pas de Londres. En outre il faudrait rajouter l'autre aéroport de Londres, l'aéroport *London City* qui a pour indicatif `LCY`.

1. Rajoutez une fonction `aeroport` qui permet d'obtenir le nom d'un aéroport à partir d'un indicatif

2. Rajoutez une fonction `changeName` qui permet de modifier le nom d'un aéroport associé à une clef.
3. Rajoutez une fonction `add` qui permet de rajouter une nouvelle donnée (code, nom).

* Exercice 7 Recherche d'indicatif

Rajoutez une fonction `searchIndicatif` qui retourne l'indicatif d'un aéroport.

* Exercice 8 Supression d'une entrée

1. Rajoutez une fonction `remove` qui supprime un aéroport de la collection d'après son indicatif.
2. Rajoutez une fonction `remove` qui supprime un aéroport de la collection d'après son nom.

* Exercice 9 Itérateur sur les indicatifs

En vous inspirant de la documentation sur le protocole IteratorProtocol, définir un type itérateur sur le type `IndicateursAeroport` qui permet de parcourir les aéroports dans l'ordre alphabétique des indicatifs.

* Exercice 10 Iterateur sur les noms d'aéroport

Même chose que précédemment mais dans l'ordre alphabétique des noms d'aéroport.