

TD I – Développement Mobile (iOS)

Objectifs de ces séances Labs

Les activités pratiques de la matière 1ADS se déroulent en **6 séances de deux heures**.

A l'issue de ces 6 séances, un étudiant devra :

- Connaître la logique de base de la programmation, en particulier en Swift.
- Maîtriser les concepts élémentaires : structures conditionnelles et itératives, sous-programmes.
- Savoir choisir une structure de données en fonction d'un problème : liste, tuple, dictionnaire, ensemble, etc.
- Etre capable d'implémenter les algorithmes les plus usuels.
- Savoir coder des petits programmes utilisant des interfaces graphiques.

Exercice 1

- ✓ Déclarez une valeur entière constante qui représente le nombre de secondes dans une heure en lui affectant ce nombre sur la même ligne.
Essayez ensuite d'attribuer une valeur différente à cette constante.
- ✓ Déclarez une variable entière avec un type explicite, puis sans type.
- ✓ Imprimez cette variable à l'aide de la fonction print ().
- ✓ Quelle est la valeur maximale pouvant être stockée dans un **Int16**? Écrivez le code qui imprime la valeur maximale du type Int16.
- ✓ Le type de la constante **p** dans l'exemple ci-dessous? Pourquoi?

```
let p = 3 + 0.141592654
```

- ✓ Que se passe-t-il si vous essayez le code suivant? Pourquoi?

```
let myNumber: UInt = -12  
let bigNumber: Int16 = 32767 + 1
```

Exercice 2

Ecrire un programme réalisant la permutation circulaire de trois variables x, y et z.

Exercice 3

Ecrire un programme calculant la somme de 1 à n, où n est un entier naturel donné par l'utilisateur.

Exercice 4

Ecrire un programme affichant une suite de nombres pairs dans une plage déterminée par l'utilisateur. Ce dernier donne une borne basse, ainsi qu'une borne haute, et le programme affiche dans un ordre croissant les nombres pairs compris dans ce domaine, sans inclure les bornes.

Exercice 5

Ecrire un programme qui détermine si une chaîne de caractère donnée est un palindrome. Rappelons qu'un palindrome est un mot qui se lit « dans les deux sens », comme par exemple « laval ».

Exercice 7

Ecrire une fonction qui prend en paramètres deux entiers et renvoie leur somme.

Exercice 8

Ecrire une fonction qui prend en entrée un tableau d'entiers et renvoie un tableau constitué des nombres positifs de ce tableau.

Exercice 9

Ecrire une fonction qui prend en entrée un tableau d'entiers et renvoie un tableau composé des nombres pairs de ce tableau.

Exercice 10

Ecrire une fonction qui prend en entrée un entier, et renvoie true si le nombre est divisible par 2 ou 3 et false sinon. Rappel : pour savoir si a est divisible par 2, le test est $a \% 2 == 0$.

Exercice 11

Ecrire une fonction calculant le nombre d'occurrences d'un élément dans une liste.

Exercice 12

Ecrire une fonction qui prend en entrée un tableau de String et renvoie un dictionnaire qui compte pour chaque valeur du tableau le nombre de fois quelle apparaît.

Exercice 13

1. Créer une structure Employé
 - ✓ Avec les propriétés : nom, âge, sexe, fonction
 - Valeurs possibles pour la fonction : technique, commerciale, financière , comptabilité, administrative.
 - Valeurs possibles pour le sexe : homme, femme
2. Créez quelques employés
3. Mettre ces employés dans un tableau employés
4. Ecrire une fonction qui prend en entrée un tableau d' employé et une fonction : et qui renvoie un tableau composé des employés de cette fonction
5. L'appliquer au tableau employés pour tester

Problème 1 :

On modélise une application devant servir à l'inventaire d'une bibliothèque. Elle devra traiter des documents de nature diverse : des livres, des dictionnaires, et autres types de documents qu'on ne connaît pas encore précisément mais qu'il faudra certainement ajouter un jour (articles, bandes dessinées...). Tous les documents possèdent un numéro d'enregistrement et un titre. A chaque livre est associé, en plus, un auteur et un nombre de pages, les dictionnaires ont, eux, pour attributs supplémentaires une langue et un nombre de tomes. On veut manipuler tous les articles de la bibliothèque au travers de la même représentation : celle d'un document.

1. Définissez les classes Document, Livre et Dictionnaire. Définissez pour chacune un constructeur permettant d'initialiser toutes ses variables d'instances.
2. Tester les classes précédentes.
3. Définissez la classe ListeDeDocuments permettant de créer une liste vide de documents, puis y adjoindre une fonction permettant d'ajouter un document.
4. Dans la classe ListeDeDocuments Définissez une méthode tousLesAuteurs() qui affiche la liste des numéros des documents de la liste avec, pour chacun, l'éventuel auteur.