**ALGO.3 : Procédures & Fonctions**

|  |
| --- |
| Objets d’un traitement informatique |

Contexte :

L'association sportive CNC, Club Nautique de la Chevreuse, utilise une application informatique pour la gestion des activités proposées par ce club. L'une des fonctionnalités de cette application permet la gestion des adhérents.

L'application informatique utilise des procédures et des fonctions.

# Introduction

Lorsque l'on progresse dans la conception d'un algorithme, ce dernier peut prendre une taille et une complexité croissante. De même des séquences d'instructions peuvent se répéter à plusieurs endroits.

Un algorithme écrit d'un seul tenant devient difficile à comprendre et à gérer dès qu'il dépasse deux pages. La solution consiste alors à découper l'algorithme en plusieurs parties plus petites. Ces parties sont appelées des sous-algorithmes.

Le sous-algorithme est écrit séparément du corps de l'algorithme principal et sera appelé par celui-ci quand ceci sera nécessaire.

Il existe deux sortes de sous-algorithmes : les procédures et les fonctions.

# Les procédures

Syntaxe création de le procédure (identique à ALGO) :

**PROCEDURE** nom\_de\_la\_procedure (*liste de paramètre : type*)

DÉBUT

VARIABLES

action(s)

FIN

Syntaxe appel de la procédure dans l'algo

**ALGO** nom\_de\_lalgo (*liste de paramètre : type*)

DÉBUT

VARIABLES

action(s)

**NOM\_DE\_LA\_PROCEDURE(*liste de paramètre*)**

FIN

Contexte :

Le site internet de l'association a des dizaines de page en php. Sur plusieurs pages, le site affiche la liste des adhérents : nom, prénom, âge, et catégorie en fonction de l'âge.

Rappel des catégories :

* Mineur
* Senior
* Vétéran (à partir de 40 ans)

Voici l'affichage qu'il faut avoir pour la liste des adhérants :

"<nom> <prenom> : <age> ans (cat : <lacategorie>)" 🡪 "Hall Gorithme : 65 ans (cat : Vétéran)"

Voici une représentation de la page d'accueil

ACCUEIL

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

téléphone : 01.02.03.04.05

Voici la liste des adhérents :

Hall Gorithme : 65 ans (cat : Vétéran)

Hall Gorithme : 65 ans (cat : Vétéran)

Hall Gorithme : 65 ans (cat : Vétéran)

Hall Gorithme : 65 ans (cat : Vétéran)

Hall Gorithme : 65 ans (cat : Vétéran)

Hall Gorithme : 65 ans (cat : Vétéran)

N'hésitez pas à nous contacter

**Q1 : Faire l'algorithme de la page d'accueil**

*🡪l'algo reçoit en paramètre un tableau de nom, de prénom, d'âge ainsi qu’un nombre d’adhérent. On supposera que chaque indice des tableaux correspond au même adhérent*

**Q2 : Faire la procédure AFFICHAGE\_ADHERENT d'affichage des adhérents** (qui reçoit en paramètre les 3 tableaux et le nombre d’adhérent)

**Q3. Utiliser la procédure d'affichage dans l'algorithme de la page d'accueil.**

L'association souhaite afficher en haut de toutes ses pages :

* 40 caractères \*
* téléphone : 01.02.03.04.05
* 40 caractères \*

**Q4. Ecrire un exemple d'algorithme permettant de faire cet affichage**

🡪 Vous optimiserez votre code en utilisant une ou plusieurs procédures.

**Q5. Proposez une définition d'une procédure**

# Les fonctions

Syntaxe création de la fonction :

**FONCTION** nom\_de\_la\_fonction (*liste de paramètre : type*) **: type**

DÉBUT

VARIABLES

action(s)

**retourner** unevariable

FIN

Syntaxe appel de la fonction (doit se faire dans un endroit qui attend une valeur)

**ALGO** nom\_de\_lalgo (*liste de paramètre : type*)

DÉBUT

VARIABLES

action(s)

**~~NOM\_DE\_LA\_FONCTION(~~*~~liste de paramètre~~*~~)~~**

*//appel dans une affectation*

a <- **NOM\_DE\_LA\_FONCTION(*liste de paramètre*)**

*//appel dans un affichage*

AFFICHER ("ma valeur =", **NOM\_DE\_LA\_FONCTION(*liste de paramètre*)**)

*//appel dans une condition*

SI (mavariable > **NOM\_DE\_LA\_FONCTION(*liste de paramètre*)**) ALORS

action(s)

FINSI

FIN

**Q6 : Faire la fonction qui renvoie la catégorie en fonction d'un âge.**

**Q7 : Utiliser cette fonction dans la procédure AFFICHAGE\_ADHERENT effectuée en Q2.**

**Q8 : Proposez une définition d'une fonction.**

**Q9. Quelles sont les avantages des procédures et fonctions ?**

# Utilisation en java

En java, on ne désigne pas "procédure" ou "fonction" mais "méthode", en effet, les fonctions (ou procédures) seules n'existent pas. Elles font parties d'un objet (appelé une classe) et définissent le comportement de cet objet.

Syntaxe :

