



Département économique IESN

1^{re} baccalauréat en
Informatique de Gestion

Langage de programmation avancé

Dossier de programmation

Année académique 2016-2017

L'**objectif principal** de ce dossier est de traduire un diagramme d'actions¹ correspondant à la problématique posée tout en respectant les règles du *Clean code* et les consignes imposées.

La **thématique** est la gestion des inscriptions à une compétition nationale de bowling. Cette compétition se déroule dans un centre donné.

Un centre de bowling possède un nombre maximum de pistes (par exemple, 16 à Namur). Ce nombre sera fixé dans votre programme sous forme de constante symbolique.

La compétition se joue en doublettes (équipe de 2 joueurs) ; chaque doublette occupe une piste.

Un même joueur ne peut s'inscrire dans plusieurs doublettes.

Pour permettre la participation d'un nombre maximum de joueurs, plusieurs shifts (sessions) seront organisés sur une période d'une à deux semaines. Une doublette ne peut s'inscrire qu'à un seul shift : lors de ce shift, chaque équipe présente joue 6 jeux.

Au terme de la compétition, chaque doublette inscrite aura donc joué 6 jeux. Le classement de la compétition se fera sur base du résultat obtenu au terme de ces 6 jeux (ordre décroissant du nombre total de quilles tombées). Il y aura un classement différent pour chaque « catégorie ». La catégorie à laquelle appartient la doublette est déterminée par la moyenne arithmétique des moyennes officielles (voir fichier ci-dessous) des 2 joueurs de l'équipe selon le schéma suivant :

Catégorie	Moyenne
1	200 et plus
2	190 à 199
3	180 à 189
4	170 à 179
5	160 à 169
6	159 et moins

Les données de départ sont :

- Un fichier **DatesOrgShifts.dat**² reprenant un enregistrement par shift. Chaque enregistrement est de longueur fixe et comprend :
 - la date du shift (AAAAMMJJ) sous forme d'entier ;
 - l'heure du shift (HHMM) sous forme d'entier.

Ce fichier est classé par date et, pour une même date, par heure. Vous créez ce fichier par un éditeur de texte tel que *NotePad* ; les données seront triées manuellement lors de l'encodage.

- Un fichier **MembresFSBB.dat** reprenant les membres de la fédération sportive belge de bowling (FSBB) en ordre de cotisation.
Chaque enregistrement de ce fichier (longueur fixe) concerne un membre et comprend :

¹ Ce diagramme a été construit par l'ensemble des professeurs et doit être respecté. Néanmoins, il doit être complété dans le même sens que les modules déjà écrits.

² Ce fichier est donné par les professeurs ainsi que le fichier des membres ci-dessous.

- le numéro de matricule (nombre < 999999) ;
- le nom (80 caractères tout compris) ;
- le prénom (30 caractères tout compris) ;
- la moyenne officielle de l'année précédente (entier représentant le nombre moyen de quilles tombées par jeu sur l'ensemble de l'année, nombre moyen arrondi à l'unité inférieure).

Ce fichier est classé par numéro de matricule croissant.

- Deux fichiers **Francais.txt** et **English.txt** de type texte dans lesquels on lira les messages à afficher suivant la langue (français ou anglais) choisie par l'utilisateur du programme. Vous encodez vos fichiers via un éditeur de texte.

La structure de chaque enregistrement sera la suivante : un code entier lié au message à afficher puis le message.

Exemple :

1001MENU PRINCIPAL

1002Ajouter une inscription

1003Supprimer une inscription

1004Imprimer les doublettes par shift

1005Quitter

2001Date du shift choisi (format AAAAMMJJ) :

2002Heure du shift choisi (format HHMM) :

2003Numero de matricule :

2004Recommencer ? (1. Oui | 0. Non) :

2005Choix :

5001Ce choix n'est pas disponible !

5002Membre inconnu...

5003Ce membre est inscrit dans une autre doublette !

5004Ce shift est complet !

5005Aucun shift propose !

5006Ce shift n'existe pas !

5007Ce n'est pas une doublette valide...

Avant de créer ces fichiers, **réfléchissez** à la structure du code ! Dans notre exemple, tous les messages concernant le menu principal commence par 1, les messages pour des obtentions par 2, les messages d'erreur par 5.

Dans un premier temps, établissez une liste des messages, regroupez-les judicieusement et encodez.

Le **programme** demande tout d'abord de choisir la langue souhaitée par l'utilisateur puis :

- La 1^{re} fois qu'il est exécuté, lire le fichier DatesOrgShifts.dat et garnir une liste chaînée simple reprenant les shifts organisés.

Chaque chaînon comprendra :

- la date du shift,
- l'heure du shift,
- le nombre de doublettes déjà inscrites (0 au départ),
- un pointeur vers une liste des inscriptions pour ce shift (NULL au départ),
- un pointeur vers le chaînon suivant.

Cette liste doit respecter l'ordre chronologique (même ordre que les enregistrements du fichier).

La sous-liste des inscriptions comprendra un chaînon par doublette. Chaque chaînon sera composé :

- du numéro de matricule du 1^{er} joueur,
- du numéro de matricule du second joueur,
- de la catégorie attribuée à la doublette,
- du pointeur vers le chaînon suivant.

- Afficher un menu proposant à l'utilisateur :

- d'ajouter une inscription (afficher les shifts organisés non complets, entrer le choix, réaliser l'inscription à partir des numéros de matricule ; l'inscription se fait par équipe complète)
- de supprimer une inscription (à partir des numéros de matricule)
- [de déplacer une inscription (d'un shift à un autre)]³
- d'imprimer par shift, les doublettes inscrites (par shift, date, heure, noms, prénoms et catégorie des doublettes ; le message sera adapté en cas de shift sans inscription)
- [d'imprimer les doublettes inscrites par catégorie (noms et prénoms)]⁴
- de quitter et sous-entendu, de sauvegarder dans un fichier Inscriptions.dat, les inscriptions effectives de manière à pouvoir regarnir les listes à l'exécution suivante du programme.

Tous les cas d'erreur seront envisagés (mémoire insuffisante, fichier inexistant, matricule déjà inscrit dans une autre doublette, ...). L'erreur sera remontée à la fonction principale ; celle-ci se chargera d'appeler un module qui affichera le message correspondant.

³ Facultatif. Points bonus si réalisé et fonctionnel.

⁴ Facultatif. Points bonus si réalisé et fonctionnel.

Consignes par rapport au langage

1. Votre programme sera de type multifichiers⁵.
2. Les noms des fichiers de données, les dimensions des tableaux seront sous forme de constantes symboliques dans un fichier de type *header file*.
3. Aucune variable ne sera globale.
4. Clean Code⁶...

Quelques rappels :

- a. Pour toute énumération, vous devrez employer l’instruction *enum*.
- b. Noms de variables et noms de fonctions adéquats et compréhensibles
- c. Présentation du code
- d. Synonymes aux structures
- e. Etc...

Consignes générales

Le programme (codes : source, compilé et exécutable) – sous le nom LangC_IG1_Groupe_NomPrenom – sera remis à votre professeur de laboratoires pour **une date précisée ultérieurement. Aucun retard ne sera accepté sauf cas de force majeure.** En cas de non remise à la date indiquée, une cote nulle sera attribuée.

Nous n’accepterons pas non plus des travaux qui ne seraient pas réalisés en laboratoires (votre présence est requise pour que le professeur voit l’évolution de votre travail) ou qui seraient des plagiats⁷.

La cotation sera attribuée en fonction des points suivants :

- votre programme s’exécute correctement au niveau des fonctionnalités attendues,
- votre code source est de qualité ou non.

Dans la cotation de votre travail, des bonus seront attribués pour :

- les champs des enregistrements dans les fichiers de longueur variable;
- ou des fonctionnalités supplémentaires dans cet énoncé.

Toutefois, ces bonus ne seront attribués que si la note du travail de base atteint les 80%.

⁵ Un document de référence de construction du projet est donné. Il est à respecter.

⁶ N’oubliez pas de consulter le syllabus.

⁷ Cf. Règlement des études : accès via le site Henallux