



14/08/2021

Solution élaborée

Projet « Gestion des frais »

Jean-Sébastien CAMBIER la196912
1BI2D

Table des matières

Agencement des données	2
Agencement des résultats	2
Agencement des formulaires	3
UF_Start	3
UF_Budget (accessible via UF_Start)	3
UF_ComparaisonBudget (accessible via UF_Budget)	3
UF_Factures (accessible via UF_Start)	3
UF_NewFacture (accessible via UF_Factures)	3
UF_Fournisseurs (accessible via UF_Start)	3
UF_NewFournisseur (accessible via UF_Fournisseurs)	3
UF_ModFournisseur (accessible via UF_Fournisseurs)	4
UF_Enseignants (accessible via UF_Start)	4
UF_NewEnseignant (accessible via UF_Enseignants)	4
UF_ModEnseignant (accessible via UF_Enseignants)	4
Méthodes de programmation utilisées	4
Modules de classes	4
Budget	4
Enseignant	4
Facture	4
Fournisseur	4
Modules	5
Start	5
FonctionsFeuilles	5
Dictionnaires	5

Solution élaborée

Agencement des données

Les données sont agencées dans les feuilles de calcul suivantes, chacune contient les données nécessaires au bon fonctionnement du programme :

- Fournisseurs : la liste des fournisseurs
- Liste Facture Type : données nécessaires pour générer une liste de factures pour une année.
- Type Budget : données nécessaires pour générer une feuille de bilan pour une année.
- Type Frais : données nécessaires pour alimenter les combo box pour encoder une facture selon les catégories et types de frais.
- Enseignants : la liste des enseignants
- Années : les années d'activité

Les feuilles « Type » serviront à générer automatiquement les feuilles de résultats, qui seront à la base des copies de celles-ci, cela facilitera la vie de l'utilisateur, s'il faut modifier ces données pour les années à venir.

J'ai choisi de cacher ces feuilles pour éviter les fausses manipulations de l'utilisateur, et donc éviter une potentielle corruption des données.

Agencement des résultats

Les résultats sont stockés dans des feuilles générées grâce au code. Ces feuilles sont nommées :

- 20XX : la liste des factures relative l'année 20XX.
- Bilan20XX : cette feuille contient le budget alloué et les dépenses triées par catégorie et types de frais de l'année 20XX.
- NomPrenom : la liste des factures encodées par l'enseignant NomPrenom.
- Factures : toutes les factures, toute années confondues.
- FactureFiltrage : la liste des factures selon le critère de recherche.
- ComparaisonBudget : la comparaison du budget de deux années différentes.

Tout comme pour les données, les feuilles de résultats sont cachées afin de ne pas corrompre les données et pour raison de sécurité. Cependant, elles seront visibles et modifiables via certains formulaires.

Agencement des formulaires

UF_Start

Il s'agit du premier formulaire affiché quand l'utilisateur ouvre le programme, il sert de menu et permet de choisir ce que l'utilisateur décide de gérer :

- Le Budget
- Les fournisseurs
- Les enseignants
- Les factures

UF_Budget (accessible via UF_Start)

Il s'agit du formulaire lié à la gestion du budget, en choisissant une année via un combo box, l'utilisateur pourra visualiser et modifier le budget, ainsi que visualiser les dépenses de l'année sélectionnée. La répartition du budget sera aussi visible graphiquement dans le formulaire. L'utilisateur peut exporter le budget (le bilan) de l'année sélectionnée en PDF et/ou l'imprimer.

UF_ComparaisonBudget (accessible via UF_Budget)

Ce formulaire permet à l'utilisateur d'avoir une comparaison chiffrée et graphique de budgets entre deux années sélectionnées via deux combos box.

L'utilisateur peut exporter cette comparaison en PDF et/ou l'imprimer.

UF_Factures (accessible via UF_Start)

Ce formulaire permet à l'utilisateur d'afficher la liste des factures pour chaque année ainsi que pour toutes les années confondues, il est également possible d'effectuer des recherches par critères avec des mots clés. L'utilisateur peut également imprimer ou exporter une liste de factures en PDF.

UF_NewFacture (accessible via UF_Factures)

Ce formulaire sert à encoder une nouvelle facture, Les critères à encoder pour celle-ci sont :

- La date (doit respecter le format date sinon erreur)
- Le montant (doit respecter le format numérique et être positif sinon erreur)
- Le fournisseur
- La catégorie de frais
- Le type de frais
- L'objet
- La section concernée
- l'éventuel enseignant

UF_Fournisseurs (accessible via UF_Start)

Affiche la liste des fournisseurs présents dans la base de données, ils peuvent être ajoutés, supprimés, ou modifiés. Possibilité d'exporter en PDF et/ou imprimer.

UF_NewFournisseur (accessible via UF_Fournisseurs)

Permet d'ajouter un fournisseur dans la base de données, les critères à encoder :

- Nom de la société (obligatoire)
- Numéro de téléphone
- Adresse électronique (doit respecter le format d'une adresse mail)
- Domaine d'activité (obligatoire)

[UF_ModFournisseur \(accessible via UF_Fournisseurs\)](#)

Permet de modifier tous les critères d'un fournisseur, si une fausse manipulation dans la modification, l'utilisateur peut réinitialiser les données.

[UF_Enseignants \(accessible via UF_Start\)](#)

La première listbox affiche la liste des enseignants (l'utilisateur peut ajouter ou modifier un enseignant) la deuxième listbox permet d'afficher la liste des frais effectués par l'enseignant sélectionné dans la première listbox (s'il en a), exportable en format pdf.

[UF_NewEnseignant \(accessible via UF_Enseignants\)](#)

Permet d'ajouter un enseignant à la base de données, selon le nom et le prénom

[UF_ModEnseignant \(accessible via UF_Enseignants\)](#)

Permet de modifier le nom et/ou le prénom d'un enseignant

Méthodes de programmation utilisées

Modules de classes

Budget

Cette classe permet d'utiliser dans le programme des objets de type Budget, cette classe a pour attribut : entretiens, telecom, autresFourn, retrib, infos, assurances et autres (tous ces attributs sont des réels).

Pourquoi avoir créé cette classe ? Tout simplement pour pouvoir encoder le budget sur les feuilles de calculs en numérique et non en chaîne de caractères

Enseignant

Cette classe a un seul attribut : nomPrenom (chaîne de caractères), cette classe permet d'utiliser des objet Enseignant dans le programme, ces objets sont composés du nom et du prénom d'un enseignant.

Pourquoi avoir créé cette classe ? Pour pouvoir stocker les enseignant dans une collection de type dictionnaire.

Facture

Cette classe est composée des attributs suivants : num (numero de la facture), montant, fournisseur, categorieFrais, typeFrais, objet, concerne, fichier (en String) et montant (en double)

Cela permettra d'utiliser dans le programme des objets de type facture, et pouvoir les stocker dans une collection de type dictionnaire.

Fournisseur

Cette classe possède quatre attributs : societe, telephone, mail, domaine (en String)

Elle servira pour l'encodage de fournisseurs dans le programme, et pour le stocker dans une collection de type dictionnaire.

Modules

Start

Ce module permet d'utiliser une macro qui lancera le programme lors de l'ouverture du classeur Excel, ou en cliquant sur un bouton « Lancer le programme » sur la feuille d'accueil.

FonctionsFeuilles

Ce module a pour objectif de regrouper les fonctions du programme liées aux feuilles de calcul, par exemple elle contient :

- Une fonction pour trouver le nombre d'élément sur une feuille
- Une fonction pour vérifier si une feuille existe dans le classeur
- Une procédure pour créer les feuilles relatives aux factures d'une année
- Une procédure pour ajouter une facture, a une feuille d'une année spécifique
- Une procédure pour générer une année d'activité sur la feuille « Années »
- Une procédure pour mettre à jour les combobox grâce aux données des feuilles de calculs
- Une procédure pour créer un graph sur la feuille des budgets
- Une procédure pour exporter une feuille au format PDF
- Une procédure pour imprimer une feuille

Dictionnaires

Ce module a pour but de déclarer les collections de type dictionnaires et des méthodes permettant de les initialiser.

Il y a trois dictionnaires :

- ensDictionnary : dictionnaire contenant les objets de type Enseignant
- factDictionnary : dictionnaire contenant les objets de type Facture
- fournDictionnary : dictionnaire contenant les objets de type Fournisseur

Les trois procédures computeEnsDisctionnary, computeFactDictionnary et computeFournDictionnary permettent d'initialiser les dictionnaires en récupérant les données récupérées dans les feuilles de données, elles sont exécutées en lancement de programme.

J'ai choisi d'utiliser des dictionnaires pour l'ajout, la modification ou la suppression de données, pour éviter des doublons et donc de potentiels bugs.