PROJET C 2015-2016:

GESTION DE BUDGET

CHEN Yutian

-Groupe 5-

YESUFU Sarah

PLAN

Introduction

- Explication globale du travail
- Fonctions
- Difficultés
- Répartition du travail

Conclusion

INTRODUCTION

La gestion budgétaire est un plan ou un état prévisionnel des recettes et de dépenses présumées qu'une personne aura à encaisser et à effectuer pendant une période donnée.

Notre travail consiste à permettre à un utilisateur quelconque de gérer et planifier son budget personnel. Il s'appuie sur le relevé bancaire de ce dernier donné en entrée. Les relevés bancaires utilisés sont en format texte (.txt) et correspondent au format AFB120 (utilisé en France).

EXPLICATION GLOBALE DE NOTRE PROJET

- *En première partie :*

  A partir du relevé bancaire de l'utilisateur, nous devons classifier les entrées et sorties d'argent. La classification se fait en fonction des revenus et des dépenses regroupés selon les catégories LOISIRS, QUOTIDIEN, FACTURE, IMMOBILIER, TRAITEMENT, TRANSPORT et NON CLASSE pour les libellés non reconnus. Ces différentes catégories regroupent des sous-classes telles que BANQUE, LOYER, SALAIRE …

- *En seconde partie :*

  Cette partie consiste à faire fournir des statistiques sur les dépenses/revenus mensuelles et des graphiques pour voir leur évolution. On utilisera l'outil GNUPLOT.

- *En troisième partie :*

  Le programme renvoie une alerte lorsqu'on approche d'un seuil fixé par catégorie par l'utilisateur lui-même.

- *En quatrième partie :*

  Cette partie concerne la sauvegarde de toutes les données traitées dans des fichiers.

FONCTIONS ET DIFFICULTES

- Les structures :
  - Dépense : elle contient la date, le libellé, le montant et la classification d'une dépense dans le relevé.
  - Revenu : elle contient les mêmes attributs que la structure Dépense adapté au type revenu (entrée d'argent).
- Fichier FileReaders :
  - FileReader : lit le fichier ligne par ligne, effectue le découpage de chaque ligne à partir du caractère « , » et les enregistre dans un tableau.
  - FileReaderClassification : lit les mots clés de classification et les enregistre dans un tableau.
- Fichier RelevéProcess :
  - classification() : classifie tous les informations du relevé selon les différentes catégories.
  - Les fonctions « set » : récupère les données "date", "libellé" et "montant" du relevé et les mémorise.
  - *getClassification : classifie les informations en fonction du libellé.
  - setAllOfLigne : crée et sépare une dépense ou un revenu avec un numéro de ligne du relevé donné.

- Fichier Somme :
  - SumXxxD : Fonction qui calcule les sommes des dépenses par catégorie.
  - SumXxxR : Fonction qui calcule les sommes des revenus par catégorie.
- Fichier print :

  Il contient la plupart des fichiers permettant les affichages sur la console (affichage des tableaux de revenus/ dépenses et de leur détails).
- Fichier seuils :
  - setSeuil: Partant sur la base de seuils prédéfinis, cette fonction assure la mise à jour d'un seuil entré par l'utilisateur.
  - afficheAlertes: Fonction affichant les messages d'alerte selon les cas.
- GNUPLOT :

  Nous avons choisi d'afficher un histogramme modélisant la somme des montants par catégorie. On récupère les sommes calculées précédemment et on les écrit dans des fichiers texte externes. A partir de ces fichiers, on trace le graphique correspondant.
- L'interface de communication :

  Cette interface visible dès l'exécution du programme, présente à l'utilisateur les différentes options de gérance de son monde bancaire: *afficher le relevé, afficher la classification, les sommes dépensées et reçues par catégorie, définir des seuils d'alerte…*

DIFFICULTES

La difficulté majeure de ce projet réside dans la concision de l'énoncé du sujet. En effet, contrairement aux précédents projets de groupe que nous avons effectué, celui-ci ne comportait aucunes indications sur le squelette du programme à réaliser.

Dans la globalité, les fonctions n'ont causé de problèmes majeurs. Par contre, étant confrontés pour la première fois à l'outil GNUPLOT, il nous a demandé beaucoup de temps pour la compréhension de son utilisation (et surtout pour l'intégrer dans notre fichier c).

Par ailleurs, comme tout nouveau programmeur, la notion de pointeurs nous a mené des bâtons dans les roues. En effet, nous étions confrontés à des tableaux de 3 dimensions. Il fallait distinguer, sinon maîtriser la compréhension des pointeurs et tableaux, surtout les pointeurs doubles ou même triples, et les tableaux de tableaux... Par exemple, la différence entre char **p, char *p[], char (*p)[], char p[][]…

L'autre difficulté est l'attention dont il fallait faire montre vu que nous avons codé sur Code::Blocks ; cet ADE ne désignant pas les erreurs commises.

REPARTITION DU TRAVAIL

	Recherche générale	Programmation et Tests	Finitions et Rapport
Yutian	10h	55h	1h
Sarah	10h	45h	8h

CONCLUSION

Service proposé en ligne par toutes les banques, la gestion budgétaire que nous avons ainsi réalisée nous permet de proposer notre version d'utilisation.

Plus subjectivement, ce projet nous a permis de consolider les acquis en cours, d'accroître notre esprit de gestion de projet en nous faisons concevoir de A à Z une solution à un problème donné.