

Compte rendu du projet 2

But

Le but du projet est de créer un programme Java qui permet de gérer ses comptes: courant et épargne tout en prenant compte les limites et seuils.

Il sera question pour l'utilisateur de s'identifier premièrement et de pouvoir ainsi gérer son compte et ses bénéficiaires.

A l'aide d'un menu comportant plusieurs fonctions et classes que nous développerons juste après, le client aura l'opportunité d'effectuer plusieurs tâches comme faire des virements, des retraits.

Structure du compte :

Notre programme est constitué d'une boucle principale contenant différents blocs. Chaque bloc ayant une fonction spécifique.

Schéma de la structure de l'algorithme :

```
Int bloc1 = 1;  
Int bloc2 = 0;  
Int bloc3 = 0;  
Int bloc4 = 0;
```

```
While(True)  
{  
    If (bloc1 == 1)  
    {  
        ...  
    }  
  
    If (bloc2 == 1)  
    {  
        ...  
    }  
  
    If (bloc3 == 1)  
    {  
        ...  
    }  
  
    If (bloc4 == 1)  
    {  
        ...  
    }  
}
```

}

Il n'y a toujours qu'un seul bloc actif (egale à 1) à la fois. Ici le premier bloc activé sera bloc1.

Il existe 9 blocs:

- Menu entrer compte : Il demande à l'utilisateur d'entrer le code de son compte, vérifie dans le fichier texte contenant tous les compte, s'il existe bien un compte avec le code entré. S'il n'existe, pas on lui affiche un message d'erreur puis on retourne au début du bloc. S'il existe, on stock les informations du compte contenu dans le fichier texte dans une variable de type Compte (construite pour stocker toutes les informations d'un compte) puis on se place dans le menu compte.
- Menu compte : Il propose différentes options à l'utilisateur (obtenir les informations du compte, effectuer un versement, etc...). On se place ensuite dans le bloc choisit par l'utilisateur.
- Information compte : Affiche sur la console les informations stockées de la variable de type compte.
- Versement : Demande à l'utilisateur un montant à ajouter à son compte, vérifie, pour les comptes épargnes, s'il ne dépasse pas sa limite maximum.
- Retrait : Demande à l'utilisateur un montant à retirer à son compte, vérifie, s'il ne dépasse pas sa limite minimum.
- Virement : demande à l'utilisateur a quel bénéficiaire il veut envoyer de l'argent puis demande le montant du virement en vérifiant si la limite minimum et la limite maximum du bénéficiaire n'est pas dépassé.
- Ajouter compte : ajoute un compte au fichier texte.
- Supprimer compte : supprime un compte au fichier texte.

Explication et fonctionnement du code:

Tout d'abord on demande à l'utilisateur de rentrer son numéro de compte et le programme vérifie s'il existe dans le fichier texte que nous avons au préalable définie. Tous les comptes sont stockés dans un fichier texte. Si le numéro de compte est dans le fichier texte alors l'exécution du programme continue ou sinon on redemande un code.

On peut accéder au menu principal qui comporte toutes les options et informations du compte que l'on peut afficher de la manière suivante et dont la sélection du choix se fait grâce à un « switch ».

```

if( menu_compte == 1 )
{
    choix_menu_compte_valide = 0;

    while( choix_menu_compte_valide == 0 )
    {

        System.out.println("\n\n\nMENU\n");

        System.out.println("1 : Changer de compte");
        System.out.println("2 : Information sur votre compte");
        System.out.println("3 : Versement d'argent");
        System.out.println("4 : Retrait");
        System.out.println("5 : Virement");
        System.out.println("6 : Ajouter un nouveau compte");
        System.out.println("7 : Supprimer votre compte");

        System.out.print("\nEntrez votre choix : ");
        choix_menu_compte = sc.nextInt();

        switch(choix_menu_compte)
        {
            .

```

Avec l'option versement par exemple, on peut transférer ou ajouter une certaine somme, vérifier qu'on dispose bien de cette somme dans son compte.

On a rajouté des conditions réalisées avec des if pour chaque variables déclarées dans le menu pour que l'on puisse retourner au menu plus simplement comme au peut le voir juste après:

```

if (versement == 1)
{
    System.out.print("\n\nVERSEMENT\n\n");

    System.out.print("Entrez le montant que vous souhaitez déposer sur votre compte : ");
    montant = sc.nextInt();

    if(montant < 0)
    {
        System.out.print("\nVersement impossible car le montant entré est négatif.");
    }

    else if( (compte.type_de_compte).equals("épargne") && (compte.argent + montant > compte.argent_max) )
    {
        System.out.print("\nVersement impossible en raison d'un dépassement de votre limite maximum d'argent.");
    }

    else
    {
        ajouter_argent_compte(nom_fichier, recherche_code_fichier_txt(nom_fichier, code), montant );
        System.out.print("\nVersement effectué !");

        compte = initialise_compte(nom_fichier, recherche_code_fichier_txt(nom_fichier, code));
    }

    versement = 0;
    menu_compte = 1;
}

```

Explication des fonctions:

recherche_code_fichier_tx : Cette fonction a pour but de vérifier le code rentrée au clavier par l'utilisateur dans le fichier texte existant ce qui permet ensuite d'accéder au menu principal dans le cas ou le code est pré définie.

```
public static int recherche_code_fichier_txt(String nom_fichier, String code)
{
    int i=0;
    String ligne = new String();

    try
    {
        File f = new File (nom_fichier);
        FileReader fr = new FileReader (f);
        BufferedReader br = new BufferedReader (fr);

        try
        {
            do
            {
                ligne = br.readLine();//code

                if ( ligne != null && ligne.equals(code) )
                {
                    br.close();
                    fr.close();
                    return i;
                }
                while( !(ligne.equals("fin")) )
                {
                    ligne = br.readLine();//fin
                }
                ligne = br.readLine();//\n

                i++;
            }while(ligne != null);

            br.close();
            fr.close();
        }
    }
}
```

Compte initialise compte: Cette fonction permet de déclarer les variables qui sont nécessaires à l'affichage des données de l'utilisateur

```
public static Compte initialise_compte(String nom_fichier, int indice)
{
    int i=0;
    int j=0;

    String ligne = new String();

    String code = new String();
    String nom_du_proprietaire = new String();
    String prenom_du_proprietaire = new String();
    String type_de_compte = new String();
    int argent = 0;
    int argent_max = 0;
    int argent_min = 0;
    String[] beneficiaires = new String[10];

    for(i=0; i<beneficiaires.length; i++)
    {
        beneficiaires[i] = new String();
    }

    i = 0;

    try
```

afficher compte: Cette fonction permet de consulter et afficher les données personnelles du compte c'est-à-dire : nom et prénom du propriétaire, son code et la somme présente sur le compte ainsi que le type de compte comme présenté ci-joint:

```
public static void afficher_compte(Compte compte)
{
    int i=0;

    System.out.println ("Code : " + compte.code);
    System.out.println ("Nom du proprietaire : " + compte.nom_du_proprietaire);
    System.out.println ("Prenom du proprietaire : " + compte.prenom_du_proprietaire);
    System.out.println ("Type de compte : " + compte.type_de_compte);
    System.out.println ("Argent : " + compte.argent);
    System.out.println ("Argent max : " + compte.argent_max);
    System.out.println ("Argent min : " + compte.argent_min);

    System.out.println ();
    for(i=0; i<compte.beneficiaires.length && !(compte.beneficiaires[i].equals("")) ; i++)
    {
        System.out.println ("Beneficiaire[" + i + "] : " + compte.beneficiaires[i]);
    }
}
```

ajouter argent compte: Cette fonction permet de réaliser un versement d'une somme d'argent disponible sur le compte à un autre compte du même utilisateur.

```
public static void ajouter_argent_compte(String nom_fichier, int indice, int montant)
{
    int i=0;

    String ligne = new String();

    try
    {
        File f = new File (nom_fichier);
        FileReader fr = new FileReader (f);
        BufferedReader br = new BufferedReader (fr);

        try
        {
            File fichier_tempo = new File("fichier_tempo.txt");
            BufferedWriter writer = new BufferedWriter(new FileWriter(fichier_tempo));

            while(i < indice)
            {
                while( !(ligne.equals("fin")) )
                {
                    ligne = br.readLine();//fin

                    writer.write(ligne + '\n');
                }
                ligne = br.readLine();//\n

                writer.write('\n');

                i++;
            }

            writer.write(br.readLine() + '\n');//code
            writer.write(br.readLine() + '\n');//nom
            writer.write(br.readLine() + '\n');//prenom
            writer.write(br.readLine() + '\n');//type de compte
        }
    }
}
```

ajouter compte fichier: Cette fonction permet d'ajouter un compte en modifiant le fichier texte.


```

public static void ajouter_compte_fichier(String nom_fichier, Compte compte)
{
    String ligne = new String();

    try
    {
        File f = new File (nom_fichier);
        FileReader fr = new FileReader (f);
        BufferedReader br = new BufferedReader (fr);

        try
        {
            File fichier_tempo = new File("fichier_tempo.txt");
            BufferedWriter writer = new BufferedWriter(new FileWriter(fichier_tempo));

            writer.write(compte.code + '\n');
            writer.write(compte.nom_du_proprietaire + '\n');
            writer.write(compte.prenom_du_proprietaire + '\n');
            writer.write(compte.type_de_compte + '\n');
            writer.write(String.valueOf(compte.argent) + '\n');
            writer.write(String.valueOf(compte.argent_max) + '\n');
            writer.write(String.valueOf(compte.argent_min) + '\n');
            writer.write("fin" + "\n\n");

            while(ligne != null)
            {
                ligne = br.readLine();

                if(ligne != null)
                {
                    writer.write(ligne + '\n');
                }
            }

            fichier_tempo.renameTo(f);
        }
    }
}

```

supprimer_compte_fichier : Cette fonction permet de sélectionner le compte et de le supprimer

```

public static void supprimer_compte_fichier(String nom_fichier, int indice)
{
    int i=0;

    String ligne = new String();

    try
    {
        File f = new File (nom_fichier);
        FileReader fr = new FileReader (f);
        BufferedReader br = new BufferedReader (fr);

        try
        {
            File fichier_tempo = new File("fichier_tempo.txt");
            BufferedWriter writer = new BufferedWriter(new FileWriter(fichier_tempo));

            while(i < indice)
            {
                while( !(ligne.equals("fin")) )
                {
                    ligne = br.readLine();//fin

                    writer.write(ligne + '\n');
                }
                ligne = br.readLine();//\n

                writer.write('\n');

                i++;
            }

            ligne = br.readLine();

            while( !(ligne.equals("fin")) )
            {

```