

TD 1 : Classes en JAVA

Exercice 1 :

Cet exercice fait suite à l'exemple présenté en cours : la classe étudiant.

La classe Module va nous permettre de simuler les notes des étudiants pour un module donné.

La classe Module va donc être définie par un **nom** et par une **note**.

Question 1 : Créez la classe Module avec les attributs, les constructeurs adéquats et les méthodes nécessaires.

Question 2 : Ajoutez à la classe Etudiant un tableau de modules qui permettra de stocker pour chaque étudiant leurs notes acquises à chaque module. Pensez à modifier les constructeurs pour permettre la création du tableau de modules (la taille maximale du tableau sera de 14 modules).

Question 3 : Créez la classe main.

Question 4 : modifiez le Tableau de modules par un vecteur de modules (Vector).

Exercice 2 :

Il s'agit de définir une (des) classe(s) JAVA permettant de modéliser des comptes bancaires. Un compte bancaire est identifié par un numéro de compte (RIB) et le solde. Chaque compte est affecté à un client. Un client est caractérisé par son nom, son cin, son salaire et un ou plusieurs comptes qui peuvent lui être affectés.

Question 1 : Créez la classe compte et client avec les attributs, les constructeurs adéquats et les méthodes nécessaires.

Question 2 : Créez la classe main qui permet de :

1. Créer un client
2. Créer un compte et de l'affecter à un client
3. d'effectuer une opération de retrait ou de dépôt d'argent sur un compte bancaire
4. d'afficher la liste des comptes d'un client donné

Question 3 : Modifiez la classe main en intégrant un menu qui permet d'effectuer plusieurs actions jusqu'à ce que l'utilisateur choisisse d'arrêter le traitement.



Matière : Programmation orientée objet

Chargée de cours : Jihène Tounsi

Chargée de TD : Emna Ammar et Asma Ben Cheikh

Correction TD1

Exercice 1 :

```
/* Question 1 */
import java.util.*;
public class Module{
private String nom;
private float note;
//constructeur parametré
public Module( String nom, float note){
this.nom = nom; //ou bien this.setNom(nom);
this.note = note; //ou bien this.setNote(note);
}
//constructeur non parametré
public Module( ){
System.out.println("Entrer le nom du module: ");
Scanner sc = new Scanner(System.in);
String str = sc.nextLine( );
this.nom = str; // ou bien this.setNom(str);
System.out.println("Entrer la note attribuée: ");
float f = sc.nextFloat();
this.note = f; // ou bien this.setNote(f);
}
//Les accesseurs: getter & setter
public String getNom( ){
return this.nom;
}
public float getNote( ){
return this.note;
}
public void setNom(String nom){
this.nom = nom;
}
public void setNote(float note){
this.note = note;
} }
/* Question 2 */
public class Etudiant
{
private String nom;
private String prenom;
//... suite d'attributs
private Module[] tabModules;
public Etudiant(String nom, String prenom){
this.nom = nom;
this.prenom = prenom;
tabModules = new Module[14]; //instancie l'objet tabModules et non le contenu
}
// ...
public Module[]
getModules(){
return this.tabModules;
}
public void setModules(int nbreModules){ //spcifier le nombre de modules <= 14
```

```

for (int i=0; i<nbreModules; i++){
this.tabModules[i] = new Module(); //instancie un objet Module ^ chaque case
}
}
}
/*Question 3 */
public class Main{
public static void main(String[] args) {
Etudiant etd1= new Etudiant("Ben Cheikh","Asma");
etd1.setModules(2);
}
}
/*Question 4*/
import java.util.Vector;
public class Etudiant {
private String nom;
private String prenom;
private Vector tabModules;
public Etudiant(String nom, String prenom){
this.nom = nom;
this.prenom = prenom;
tabModules = new Vector();
}
public Vector getModules(){
return this.tabModules;
}
public void setModules(int nbreModules){
for (int i=0; i<nbreModules; i++){
this.tabModules.add(i,new Module());
}
}
}
}

```

Exercice 2 :

Class Compte :

```

public class Compte {
    private int RIB;
    private float solde;
    private Client client;
    public Compte(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Entrer le RIB du compte: ");
        this.RIB = sc.nextInt( );
        System.out.println("Entrer le solde initial: ");
        this.solde= sc.nextFloat();
    }
    public Compte(int rib){
        this.RIB = rib;
        this.solde = 0;
    }
    public Compte(int rib, float solde){
        this.RIB = rib;
        this.solde = solde;
    }
    public Compte (int rib, float solde, Client clt){

```

```

        this.RIB = rib;
        this.solde = solde;
        this.client = clt;
    }
    public int getRIB(){
        return this.RIB;
    }
    public float getSolde(){
        return this.solde;
    }

    public void setSolde(float solde){
        this.solde = solde;
    }

    public Client getClient(){
        return this.client;
    }

    public void setClient(Client clt){
        this.client = clt;
        this.client.addCompte(this);
    }

    /*On peut enrichir les fonctionnalités et simplifier la main qui fera appel à
ces méthodes:
    public void debiterSolde(float montant){
        this.solde -= montant;
    }
    public void crediterSolde(float montant){
        this.solde += montant;
    }
    */
}

```

Class Client :

```

public class Client {
    private String nom;
    private int cin;
    private float salaire;
    private Vector<Compte> comptes;
    public Client(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Entrer le nom du client: ");
        this.nom = sc.nextLine();
    }
}

```

```

        System.out.println("Entrer la cin du client: ");
        this.cin = sc.nextInt();
        System.out.println("Entrer le salaire du client: ");
        this.salaire = sc.nextFloat();
        comptes = new Vector();
    }
    public float getSalaire(){
        return this.salaire;
    }
    public Client(String nom, int cin, float sal){
        this.nom = nom;
        this.cin = cin;
        this.salaire = sal;
        comptes = new Vector();
    }
    public void addCompte(Compte cpt){
        this.comptes.add(cpt);
    }
    public Compte getCompte(int RIB){
        for(int i=0; i<this.comptes.size(); i++){
            if (comptes.elementAt(i).getRIB()==RIB)
                return comptes.elementAt(i);
        }
        return null;
    }
    public void afficherComptes(){
        System.out.println("Informations du client: "+this.nom+" :");
        for(int i=0; i<this.comptes.size(); i++){
            System.out.println("Compte n° "+ i+1+" :\tRIB: "
                +comptes.elementAt(i).getRIB()+"\t Solde:
"+comptes.elementAt(i).getSolde());
        }
    }
}

```

Class Main :

```

public class Main {
    public static void main(String[] args){
        Client client1 = new Client();
        Client client2 = new Client("Asma Ben Cheikh", 01234567, 3000);
        //Création d'un compte avec un salolde initial non null:
        Compte cmptel = new Compte(1001, (float) 2000.550);
        cmptel.setClient(client1);
    }
}

```

```

//Création d'un compte avec solde initial vide
Compte cmpte2 = new Compte(1002);
cmpte2.setClient(client2);
//Afficher les informations initiales:
System.out.println("*****Initialement*****");
client1.afficherComptes();
client2.afficherComptes();
System.out.println("*****");
//Exemples d'opérations
Main.debiterSolde(cmpte1, (float) 500.00);
System.out.println("Infos après 1ere operation: débit 500 du compte 1");
client1.afficherComptes();
System.out.println("*****");
Main.crediterSolde(cmpte2, client2.getSalaire());
System.out.println("Infos après 2eme operation: dépôt de 1000 dans compte
2");

client2.afficherComptes();
System.out.println("*****");
//Ajouter un nouveau compte au client 2:
Compte cmpte3 = new Compte(1003, 5000);
cmpte3.setClient(client2);
//Afficher la MAJ:
System.out.println("*****Mise à jour*****");
client2.afficherComptes();
System.out.println("*****");
}

public static void debiterSolde(Compte cmpt, float montant){
    float solde = cmpt.getSolde();
    solde = solde - montant;
    cmpt.setSolde(solde);
    //cmpt.setSolde(cmpt.getSolde() - montant);
    //
    //Ou invoquer cmpt.debiterSolde(montant)
}

public static void crediterSolde(Compte cmpt, float montant){
    float solde = cmpt.getSolde();
    solde = solde + montant;
    cmpt.setSolde(solde);
    //cmpt.setSolde(cmpt.getSolde() + montant);
}
}

```