



Faculté des Sciences d'Agadir – Ibn Zohr

Licence d'excellence
Filière : Analytique de Données et Intelligence
Artificielle

TP 3

COMPT RENDU

Réalisé Par :
Elhillali Naoual

Création de la classe Agence :

```
Source History
9  */
10 public class Agence {
11     public final static int NB_MAX_COMPTES = 90;
12     public final static int NB_MAX_CLIENTS = 50;
13     private static int nbAgences = 0;
14     private static int nbClients = 0;
15     private static int nbComptes = 0;
16     private String numero;
17     private String adresse;
18     private Client[] lesClients;
19     private Compte[] lesComptes;
20     public Agence(String adresse){
21         numero = this.getClass().getName()+" "+ ++nbAgences;
22         this.adresse = adresse;
23         lesClients = new Client[NB_MAX_CLIENTS];
24         lesComptes = new Compte[NB_MAX_COMPTES];
25     }
26     public Compte getCompte(int numCompte) {
27         if(numCompte>=0 && numCompte<nbComptes){
28             return lesComptes[numCompte];
29         }
30         return null;
31     }
32     public void addCompte(Compte newCompte){
33         if(nbComptes<NB_MAX_COMPTES) lesComptes[nbComptes++] = newCompte;
34     }
35     public Client getClient(int numClient) {
36         if(numClient>=0 && numClient<nbClients){
37             return lesClients[numClient];
38         }
39     }
```

Création de la classe Client :

```
Source History
13 public class Client {
14     public final static int NB_MAX_COMPTES = 4;
15     private int nbComptes = 0;
16     private static int nbClients = 0;
17     private String code;
18     private String nom;
19     private String prenom;
20     private String adresse;
21     private Agence monAgence;
22     private Compte[] mesComptes;
23
24     public Client(Agence agence){
25         code = this.getClass().getName()+" "+ ++nbClients;
26         monAgence = agence;
27         mesComptes = new Compte[NB_MAX_COMPTES];
28     }
29     public Client(Agence agence, String nom, String prenom, String adresse){
30         code = this.getClass().getName()+" "+ ++nbClients;
31         monAgence = agence;
32         this.nom = nom;
33         this.prenom = prenom;
34         this.adresse = adresse;
35         mesComptes = new Compte[NB_MAX_COMPTES];
36     }
37     public void addCompte(Compte newCompte){
38         if(nbComptes < NB_MAX_COMPTES) mesComptes[nbComptes++] = newCompte;
39         else System.out.println("Erreur");
40     }
41     public boolean depenser(int numCompte, double som){
```

Création de la classe Compte :

```
Source History
6  /**
7   *
8   * @author dell
9   */
10 public class Compte {
11     private String code;
12     protected double solde;
13     protected Agence lAgence;
14     protected Client proprietaire;
15     public Compte(Client leProprioetaire, Agence agence){
16         this.solde = 0;
17         this.proprietaire = leProprioetaire;
18         this.lAgence = agence;
19     }
20     public Compte(double leSolde, Client leProprioetaire, Agence agence){
21         this.solde = leSolde;
22         this.proprietaire = leProprioetaire;
23         this.lAgence = agence;
24     }
25     public void setCode(String code) {this.code = code;}
26
27     public String getCode() {return code;}
28     public void retirer(double som) {if(som<solde) solde -= som;}
29     public void deposer(double som) {solde += som;}
30
31 }
32
```

Création de la classe ComptePayant :

```
Source History
8  * @author dell
9  */
10 public class ComptePayant extends Compte{
11     public static int nbComptesP = 0;
12     private final static double TAUX_OPERATION = 5;
13     public ComptePayant(Client leProprioetaire, Agence agence){
14         super(leProprioetaire, agence);
15         String code = this.getClass().getName()+" "+ ++nbComptesP;
16         setCode(code);
17     }
18     public ComptePayant(double leSolde, Client leProprioetaire, Agence agence){
19         super(leSolde, leProprioetaire, agence);
20         String code = this.getClass().getName()+" "+ ++nbComptesP;
21         setCode(code);
22     }
23     public void retirer(double som) {super.retirer(som+TAUX_OPERATION);}
24     public void deposer(double som) {super.deposer(som-TAUX_OPERATION);}
25     public String toString() {
26         String s =
27         "\t| "+getCode()+" | Solde: "+solde+" DH | Montant de l'operation:"+TAUX_OPERATION + " | " +"\n\t----
28         return s;
29     }
30 }
```

Création de la classe CompteEpargne :

```
Source History
9  */
10 public class CompteEpargne extends Compte {
11     private double taux;
12     public static int nbComptes = 0;
13
14     public CompteEpargne(Client leProprietaire, Agence agence, double taux) {
15         super(leProprietaire:leProprietaire, agence);
16         this.taux = taux;
17         String code = this.getClass().getName() + ":" + (++nbComptes);
18         setCode(code);
19     }
20
21     public CompteEpargne(double leSolde, Client leProprietaire, Agence agence, double taux) {
22         super(leSolde, leProprietaire:leProprietaire, agence);
23         this.taux = taux;
24         String code = this.getClass().getName() + ":" + (++nbComptes);
25         setCode(code);
26     }
27
28     public double getTaux() {
29         return taux;
30     }
31
32     public void setTaux(double taux) {
33         this.taux = taux;
34     }
35
36     public void calculerInteret() {
37         double interet = solde * taux / 100;
```

Création de la classe ApplicationBancaire(main) :

```
Source History
10 public class ApplicationBancaire {
11     public final static int NB_CLIENTS = 4;
12     public static void main(String[] args){
13         /*(a) Création de l'agence */
14         Agence lAgence = new Agence( adresse: "N 5, Rue banafsej, Hay salam, Agadir");
15
16         /*(b) Création des clients et de leurs comptes*/
17         Client[] lesClients = new Client[NB_CLIENTS];
18         String nom, prenom, adresse;
19         for(int i=0; i<NB_CLIENTS; i++){
20             nom = "NOM_" + (i+1);
21             prenom = "Prenom_" + (i+1);
22             adresse = "Rue N " + (i+1) + ", saada " + (i+1);
23
24
25             lesClients[i] = new Client( agence: lAgence, nom, prenom, adresse);
26         }
27         for(int i=0; i<lesClients.length; i++)
28             lesClients[i].toString();
29
30         /* Client 1 */
31         lesClients[0].addCompte (new CompteEpargne( leSolde:1200, lesClients[0], agence:lAgence, taux:5.0));
32         /* Client 2 */
33         lesClients[1].addCompte(new ComptePayant( leSolde:3500, lesClients[1], agence:lAgence));
34         /* Client 3 */
35         lesClients[2].addCompte(new ComptePayant(lesClients[2], agence:lAgence));
36         lesClients[2].addCompte(new ComptePayant( leSolde:5000, lesClients[2], agence:lAgence));
37
38         /* Client 4 */
```

L'exécution de code :

```
Output - Gestiondecomptbancaire (run)

run:
--- Liste des differents clients avec leursdifferents comptes ---

-----
| Client:1 | NOM_1 | Prenom_1 | @: Rue N 1, saada 1 |
-----
| CompteEpargne:1 | Solde: 1365.0 DH | Montant de l'operation:5.0 |
-----

-----
| Client:2 | NOM_2 | Prenom_2 | @: Rue N 2, saada 2 |
-----
| ComptePayant:1 | Solde: 3500.0 DH | Montant de l'operation:5.0 |
-----

-----
| Client:3 | NOM_3 | Prenom_3 | @: Rue N 3, saada 3 |
-----
| ComptePayant:2 | Solde: 0.0 DH | Montant de l'operation:5.0 |
-----
| ComptePayant:3 | Solde: 4795.0 DH | Montant de l'operation:5.0 |
-----

-----
| Client:4 | NOM_4 | Prenom_4 | @: Rue N 4, saada 4 |
-----
| CompteEpargne:2 | Solde: 2060.0 DH | Montant de l'operation:3.0 |
-----
| ComptePayant:4 | Solde: 0.0 DH | Montant de l'operation:5.0 |
-----

--- Liste des comptes d epargne de l agence ---

-----

--- Liste des comptes d epargne de l agence ---
| CompteEpargne:1 | Solde: 1365.0 DH | Montant de l'operation:5.0 |
-----
| CompteEpargne:2 | Solde: 2060.0 DH | Montant de l'operation:3.0 |
-----

--- Liste des comptes payants de l agence ---
| ComptePayant:1 | Solde: 3500.0 DH | Montant de l'operation:5.0 |
-----
| ComptePayant:2 | Solde: 0.0 DH | Montant de l'operation:5.0 |
-----
| ComptePayant:3 | Solde: 4795.0 DH | Montant de l'operation:5.0 |
-----
| ComptePayant:4 | Solde: 0.0 DH | Montant de l'operation:5.0 |
-----

BUILD SUCCESSFUL (total time: 0 seconds)
```

J'utilise le IDE Apache NetBeans.