



S0

Génie Logiciel

Programmation Objet

Auteurs :

M. Samuel RIEDO

M. Maïc QUEIROZ

30 septembre 2016

Question 1

Les deux classes principales sont Compte et Possesseur. Comme il y a plusieurs types de comptes, nous avons choisi de créer des sous-classes et de redéfinir certaines méthodes (@Override) pour garder le code le plus propre possible. De même pour les possesseurs de compte, les attributs communs sont dans la classe parente, et les spéciaux dans des sous-classes. Une classe Banque complète le tout, cette dernière est principalement utilisée pour stocker les comptes dans un tableau.

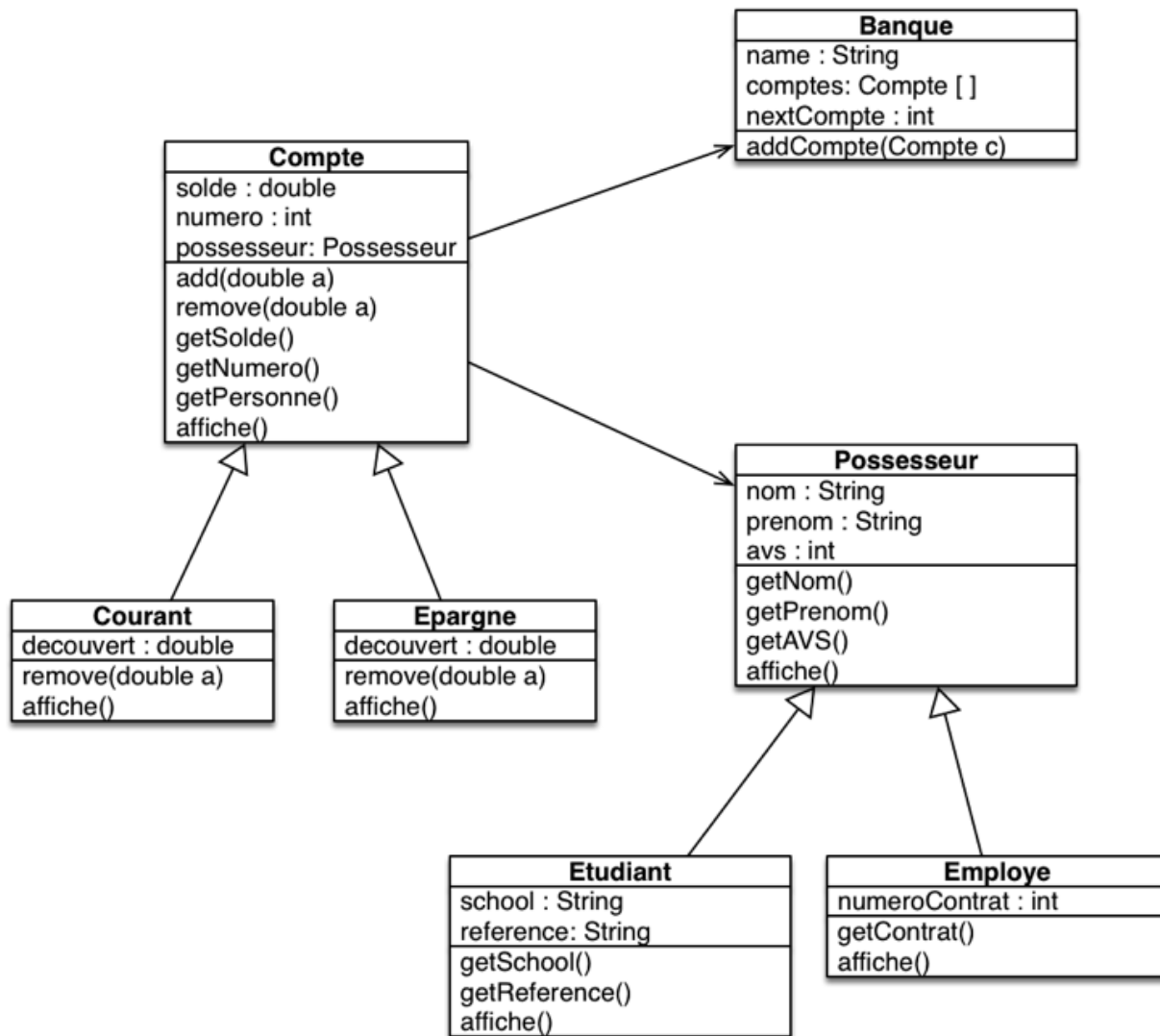


FIGURE 1 – Diagramme UML

Question 2

Les codes ci-dessus correspondent à la figure 1.

```
package tp0;

public class Main {

5   public static void main(String...args) {
        Etudiant max = new Etudiant ("Queiroz", "Max", 10, "HES-SO", "Son papa");
        Employe pierre = new Employe ("Caille", "Pierre", 11, 3786467);

        CompteEpargne compteMax = new CompteEpargne(-2000, 345345, max);
10   CompteCourant comptePierre = new CompteCourant (-1000, 4535345, pierre);

        compteMax.affiche();
        System.out.println("-----");
        comptePierre.affiche();
15   }
}
```

```
package tp0;

public class Banque {

5   private String name;
    Compte[] comptes;
    int nextcompte;

    public Banque (String n) {
10        name = n;
        comptes = new Compte[10];
        nextcompte=0;
    }

    public void addCompte(Compte c) {
15        comptes[nextcompte]=c;
        nextcompte++;
    }

20 }
```

Les attributs de cette classe sont protected, afin de les protéger des autres classes tout en les laissant accessibles aux enfants. L'usage de getter et setter est donc obligatoire.

```
package tp0;

public class Compte {

5   protected double solde;
   protected int numero;
   protected Possesseur possesseur;

   public Compte(double s, int n, Possesseur p) {
10      solde = s;
      numero = n;
      possesseur = p;
   }

15   public void add(double a) {
      this.solde+= a;
   }

   public void remove(double a) {
20      this.solde-=a;
   }

   public double getSolde() {
25      return this.solde;
   }

   public int getNumero() {
      return this.numero;
   }

30   public Possesseur getPersonne() {
      return this.possesseur;
   }

35   public void affiche() {
      System.out.println("Etat du compte No "+numero);
      System.out.println("Solde: "+solde);
   }

40 }
```

Les classes CompteCourant et CompteEpargne sont des sous-classes de Compte, avec des attributs spécifiques en plus. La méthode *affiche()* est surchargée dans les deux afin de s'adapter aux nouvelles descriptions.

```
package tp0;

public class CompteCourant extends Compte {

    5    final double decouvert = 5000;

    public CompteCourant(double s, int n, Possesseur p) {
        super(s, n, p);
    }

    10    @Override
    public void remove(double a) {
        if (this.solde-a<-decouvert)
            this.solde-=a;
    }

    15    @Override
    public void affiche() {
        System.out.println("Etat du compte No "+this.numero);
        System.out.print("Solde: "+this.solde);
        20        if (this.solde<-5000)
            System.out.println(" (blocked)");
        else
            System.out.println();
        25        this.posseceur.affiche();
    }
}
```

```
package tp0;

public class CompteEpargne extends Compte {

    5    final double decouvert = 1000;

    public CompteEpargne(double s, int n, Possesseur p) {
        super(s, n, p);
    }

    10    @Override
    public void remove(double a) {
        if (this.solde-a<-decouvert)
            this.solde-=a;
    }

    15    @Override
    public void affiche() {
        System.out.println("Etat du compte No "+this.numero);
        System.out.print("Solde: "+this.solde);
        20        if (this.solde<-1000)
            System.out.println(" (blocked)");
        else
            System.out.println();
        25        this.posseceur.affiche();
    }
}
```

Similairement à *Compte*, les attributs de *Possesseur* sont *protected* pour les mêmes raisons. Les sous-classes *Employé* et *Etudiant* redéfinissent également la méthode *affiche()*.

```
package tp0;

public class Possesseur {

    5    protected String nom;
        protected String prenom;
        protected int avs;

    public Possesseur(String n, String p, int av) {
    10        nom = n;
        prenom = p;
        avs = av;
    }

    15    public String getNom() {
        return this.nom;
    }

    public String getPrenom() {
    20        return this.prenom;
    }

    public int getAVS() {
    25        return this.avs;
    }

    public void affiche() {
        System.out.println("Possesseur: ");
        System.out.println("Nom: "+this.getNom());
    30        System.out.println("Prenom: "+this.getPrenom());
        System.out.println("Numero AVS: "+this.getAVS());
    }
}
```

```
package tp0;

public class Employe extends Possesseur {

    5    private int contrat;

    public Employe(String n, String p, int av, int c) {
        super(n, p, av);
        contrat = c;
    10    }

    public int getContrat() {
        return this.contrat;
    }

    15    @Override
    public void affiche() {
        System.out.println("Possesseur: (Employe)");
        System.out.println("Nom: "+this.getNom());
        System.out.println("Prenom: "+this.getPrenom());
    20        System.out.println("Numero AVS: "+this.getAVS());
    }
```

```
        System.out.println("Numero de contrat: "+this.getContrat());
    }
25 }
```

```
package tp0;

public class Etudiant extends Possesseur {

5     private String school;
    private String reference;

    public Etudiant(String n, String p, int av, String s, String r) {
        super(n, p, av);
10        school = s;
        reference = r;
    }

    public String getSchool() {
15        return this.school;
    }

    public String getReference() {
        return this.reference;
20    }

    @Override
    public void affiche() {
25        System.out.println("Possesseur: (Etudiant)");
        System.out.println("Nom: "+this.getNom());
        System.out.println("Prenom: "+this.getPrenom());
        System.out.println("Numero AVS: "+this.getAVS());
        System.out.println("Numero de contrat: "+this.getSchool());
        System.out.println("Referent: "+this.reference);
30    }
}
```

Fribourg, le 30 septembre 2016

Samuel Riedo

Maïc Queiroz