



Rédigé par :

Amine NABOULSI

Encadré par :

Youssef ES-SAADY

Réalisation de :

TP POO Java

Sommaire

CHAPITRE 1 : TP2	3
1.EXERCICE 1	3
1.1. CODE	3
1.2. EXECUTION	3-4
2.EXERCICE 2	4
2.1. CODE	4
2.2. EXECUTION	5
3.EXERCICE 3	5
2.2. CODE	5-8
2.3. EXECUTION	8-10
4.EXERCICE 4	16
2.2. CODE	16-19
2.3. EXECUTION	20

Chapitre II : TP1

1. Exercice 1.

1.1. Code.

```
//class Compte

public class Compte {
    int Numero;
    float Solde;
    String Propriete;
    public static int count_account=0;
    public int getCount_account() {
        return count_account;
    }

    public Compte() {
        this.Numero=0;
        this.Solde=0;
        count_account++;
    }

    public Compte(int numero, float solde) {
        Numero = numero;
        Solde = solde;
        count_account++;
    }
    public void setSolde(float solde){
        this.Solde=solde;
    }
    public void setNumero(int numero){
        this.Numero=numero;
    }
    public void Retirer(float M){
        if(M > Solde){
            new Exception("Solde insuffisant");
        }
        else{
            this.Solde-=M;
        }
    }
    public void Deposer(float M){
        this.Solde+=M;
    }

    @Override
    public String toString() {
        return "Numero :" + Numero +"\n"+
            "Solde :" + Solde +"\n"+
            "Propriete='" + Propriete +"\n";
    }
    public boolean VirerVers(int M , Compte destination){
        if(destination.Numero == 0){
            System.out.println(Colors.Bg_RED+new Exception("Ce compte n'a pas un
numero")+Colors.End_Color+"\n");
            return false;
        }
        else{
```

```

        this.Retirer(M);
        destination.Deposer(M);
        return true;
    }
}
import java.text.ParseException;
import java.util.Scanner;

//class Main

public class Main {
    public static void main(String[] args) throws ParseException {

        Compte c1 = new Compte();
        Compte c2 = new Compte();
        Compte c3 = new Compte();
        System.out.println("Nombre de compte creer :"+c3.getCount_account());
        Compte c4 = new Compte(1,400);
        Compte c5 = new Compte(2,200);
        System.out.println(c4.toString());
        System.out.println(c5.toString());
        System.out.println("Transition"+"\\n");
        if(c5.VirerVers(100,c4)){
            System.out.println(Colors.Bg_GREEN+"Virement a ete bien
envoyer"+Colors.End_Color+"\\n");
        }
        System.out.println(c4.toString());
        System.out.println(c5.toString());
    }
}

```

1.2. Exécution.

```

Nombre de compte creer :3
Numero :1
Solde :400.0
Propriete=null

Numero :2
Solde :200.0
Propriete=null

Transition

Virement a ete bien envoyer

Numero :1
Solde :500.0
Propriete=null

Numero :2
Solde :100.0
Propriete=null

Process finished with exit code 0

```

2. Exercice 2.

2.1. Code.

```
//class Animal

public class Animal {
    int nbPattes;

    public Animal(int nbPattes) {
        this.nbPattes = nbPattes;
    }

    @Override
    public String toString() {
        return "un bon animal";
    }
    public String affiche() {
        return "information d'animal";
    }
}

//class Autruche

public class Autruche extends Animal{

    public Autruche(int nbPattes) {
        super(nbPattes);
    }

    @Override
    public String toString() {
        return "un bon Autruche";
    }
}

public class Lapin extends Animal{

    public Lapin(int nbPattes) {
        super(nbPattes);
    }

    @Override
    public String toString() {
        return "un bon Lapin";
    }
}

import java.text.ParseException;
import java.util.Scanner;

//class Main

public class Zoo {
    public static void main(String[] args) throws ParseException {

        Lapin lapin = new Lapin(2);
        System.out.println(lapin.toString());
        Autruche autruche = new Autruche(2);
        System.out.println(autruche.toString());
    }
}
```

2.2. Execution.

```
un bon Lapin  
un bon Autruche
```

```
Process finished with exit code 0
```

3. Exercice 3.

3.1. Code.

```
//class employe
```

```
public class employe {  
    //Attributs  
    String Nom;  
    int Age;  
    float salaire;  
  
    //Constructeurs  
  
    public employe() {  
        this.salaire = 0;  
    }  
  
    public employe(employe e) {  
        this.Nom = e.Nom;  
        this.Age = e.Age;  
        this.salaire = e.salaire;  
    }  
  
    public employe(String nom, int age, float salaire) {  
        this.Nom = nom;  
        this.Age = age;  
        this.salaire = salaire;  
    }  
  
    //Methodes  
  
    public void Augmentation(float m) {  
        this.salaire += m;  
    }  
  
    @Override  
    public String toString() {  
        return "Nom : " + Nom + "\n" +  
            "Age : " + Age + "\n" +  
            "Salaire : " + salaire ;  
    }  
    public String affiche() {  
        return "Nom : " + Nom + "\n" +  
            "Age : " + Age + "\n" +  
            "Salaire : " + salaire ;  
    }  
    public void calculesalaire() {  
    }  
}
```

```

public class technicien {

    //Attributs
    String Nom;
    int Age;
    float salaire;
    int grade ;

    //Constructeurs
    public technicien() {
        this.salaire =0;
    }

    public technicien(String nom, int age, float salaire) {
        Nom = nom;
        Age = age;
        this.salaire = salaire;
    }

    @Override
    public String toString() {
        return "Nom :" + Nom + "\n" +
            "Age : " + Age + "\n" +
            "salaire :" + salaire + "\n" +
            "grade :" + grade ;
    }

    public technicien(technicien t) {
        Nom = t.Nom;
        Age = t.Age;
        this.salaire = salaire;
    }

    public void Augmentation(int grade){
        if(grade == 1){
            salaire+=100;
        }
        else if (grade == 2) {
            salaire+=200;
        }
        else if (grade == 3){
            salaire+=300;
        }
    }

    public int prime(){
        if(grade == 1){
            return 100;
        }
        else if (grade == 2) {
            return 200;
        }
        else if (grade == 3){
            return 300;
        }
        else{
            return 0;
        }
    }

    public String affiche(){
        return "Nom : " + Nom + "\n" +
            "Age : " + Age + "\n" +
            "Salaire : " + salaire + "\n" +

```

```

        "Grade"+grade;
    }
    public void calculesalaire(){
    }
}

import java.text.ParseException;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) throws ParseException {

        employe employe = new employe("ahmed",25, 1000.0f);
        technicien technicien = new technicien("saad",28,1500.0f);
        System.out.println("###employe ahmed###");
        System.out.println(employe.toString()+"\n");
        System.out.println("###technicien saad###");
        System.out.println(technicien.toString()+"\n");

        employe.Augmentation(100.0f);
        System.out.println("auhementation du salaire d'ahmed par 100dh");
        employe.calculesalaire();
        employe.affiche();

        technicien.Augmentation(1);
        System.out.println("auhementation du salaire de saad par 100dh");
        technicien.calculesalaire();
        technicien.affiche();

        System.out.println("\n"+"###employe ahmed###");
        System.out.println(employe.toString()+"\n");
        System.out.println("###technicien saad###");
        System.out.println(technicien.toString());
    }
}

```

3.2. Exécution.

```

###employe ahmed###
Nom : ahmed
Age : 25
Salaire : 1000.0

###technicien saad###
Nom :saad
Age : 28
salaire :1500.0
grade :0

auhementation du salaire d'ahmed par 100dh
auhementation du salaire de saad par 100dh

###employe ahmed###
Nom : ahmed
Age : 25
Salaire : 1100.0

###technicien saad###
Nom :saad
Age : 28
salaire :1600.0
grade :0

```


4. Exercice.

4.1. Code.

//class employe

```
import javax.swing.*;
import java.util.ArrayList;

public class employe {

    //attributs
    int matricule;
    String nom ;
    String adresse ;
    float salaire ;
    static ArrayList<Integer> li =new ArrayList<>();
    public Boolean check =false;

    //methode d'accès
    public int getMatricule() {
        return matricule;
    }

    public String getNom() {
        return nom;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public String getAddress() {
        return adresse;
    }

    public void setAddress(String address) {
        this.adresse = address;
    }

    public float getSalaire() {
        return salaire;
    }

    public void setSalaire(float salaire) {
        this.salaire = salaire;
    }

    //3 constructeur
    public employe() {

    }
    public employe(int matricule, String nom, String adresse) {

        this.matricule = matricule;
        this.nom = nom;
        this.adresse = adresse;
    }
}
```

```

public employe(int matricule, String nom, String adresse, float salaire) {
    if (li.contains(matricule))
    {
        check=true;
    }else {
        this.matricule = matricule;
        this.nom = nom;
        this.adresse = adresse;
        this.salaire = salaire;
        li.add(matricule);
    }
}

//methode toString
@Override
public String toString() {
    return "employe{" +
        "matricule=" + matricule +
        ", nom='" + nom + '\'' +
        ", adresse='" + adresse + '\'' +
        ", salaire=" + salaire +
        '\'';
}
}

```

//class employe horaire

```

public class employe_horaire extends employe{

    int number_heure;

    public int getNumber_heure() {
        return number_heure;
    }

    public void setsalaire(float s){
        super.setSalaire(s);
    }

    public employe_horaire(int matricule , String nom , String adresse){
        super(matricule , nom ,adresse);
    }

    public employe_horaire(employe e , int number_heure) {
        super(e.matricule , e.nom , e.adresse , e.salaire*number_heure);
        this.number_heure= number_heure;
    }
}

```

//class employe journée

```

public class employee_jours extends employe{
    int number_jours;

    public void setsalaire(float s){
        super.setSalaire(s);
    }

    public employee_jours(int matricule , String nom , String adresse){
        super(matricule , nom ,adresse);
    }

    public employee_jours(employe e , int number_jours) {
        super(e.matricule , e.nom , e.adresse , e.salaire*number_jours);
    }
}

```

```

        this.number_jours= number_jours;
    }
}

//class employe semaine

public class employee_semaine extends employe{
    int number_semain;

    public void setsalaire(float s){
        super.setSalaire(s);
    }
    public employee_semaine(int matricule , String nom , String adresse){
        super(matricule , nom ,adresse);
    }

    public employee_semaine(employe e , int number_semain) {
        super(e.matricule , e.nom , e.adresse , e.salaire*number_semain);
        this.number_semain= number_semain;
    }
}

//class Main
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        //
        employe[] table_employe = new employe[7];
        //employe_horaire 1
        employe e11 =new employe_horaire(14, "A7mad" , "adresse 1");
        e11.setSalaire(200f);
        employe e1 = new employe_horaire(e11,46);
        if (!e1.check){
            table_employe[0]=e1;
        }
        //employe_horaire 2
        employe e22 =new employe_horaire(17, "amine" , "adresse 2");
        e22.setSalaire(200f);
        employe e2 = new employe_horaire(e22,15);
        if (!e2.check){
            table_employe[1]=e2;
        }
        //employee_jours 1
        employe e33 =new employee_jours(18, "sa3id" , "adresse 1");
        e33.setSalaire(450f);
        employe e3 = new employee_jours(e33,1);
        if (!e3.check){
            table_employe[2]=e3;
        }
        //employee_jours 2
        employe e44 =new employee_jours(71, "anwar" , "adresse 2");
        e44.setSalaire(450f);
        employe e4 = new employee_jours(e44,2);
        if (!e4.check){
            table_employe[3]=e4;
        }

        //employee_semaine 1
        employe e55 =new employee_semaine(57, "lida" , "adresse 1");
        e55.setSalaire(1200f);
        employe e5 = new employee_semaine(e55,2);
    }
}

```

```

        if (!e5.check){
            table_employe[4]=e5;
        }
        //employee_semaine 2
        employe e66 =new employee_semaine(74, "3issa" , "adresse 2");
        e66.setSalaire(1200f);
        employe e6 = new employee_semaine(e66,2);
        if (!e6.check){
            table_employe[5]=e6;
        }
        //employee_semaine 3
        employe e77 =new employee_semaine(321, "samiya" , "adresse 3");
        e77.setSalaire(1200f);
        employe e7 = new employee_semaine(e77,2);
        if (!e7.check){
            table_employe[6]=e7;
        }

        //appeler methode afficher salaire
        afficher_salaire(table_employe);
    }

    public static void afficher_salaire(employe[] item){
        float somme =0;
        for (int i =0 ; i<item.length;i++){
            if(item[i] instanceof employe_horaire){
                System.out.println("L'employé horaire "+item[i].nom+" a comme salaire
"+item[i].salaire+"\n");
                somme+=item[i].salaire;
            }
            if(item[i] instanceof employee_jours){
                System.out.println("L'employé jours "+item[i].nom+" a comme salaire
"+item[i].salaire+"\n");
                somme+=item[i].salaire;
            }
            if(item[i] instanceof employee_semaine){
                System.out.println("L'employé semaine "+item[i].nom+" a comme salaire
"+item[i].salaire+"\n");
                somme+=item[i].salaire;
            }
        }
        System.out.println("Somme de tous les employe : "+String.valueOf(somme));
    }
}

```

4.2. Exécution.

```

L'employé horaire A7mad a comme salaire 9200.0
L'employé horaire amine a comme salaire 3000.0
L'employé jours sa3id a comme salaire 450.0
L'employé jours anwar a comme salaire 900.0
L'employé semaine lida a comme salaire 2400.0
L'employé semaine 3issa a comme salaire 2400.0
L'employé semaine samiya a comme salaire 2400.0
Somme de tous les employe : 20750.0
Process finished with exit code 0

```

Bien Cordialement