

Correction de l'examen Java

Exercice 1 (05 points)

1. Traitement (0.5 points)
2. Setters et Getters de Nom et Prénom (01 points)
3. Méthode ajouter (Float note) (0.5 points)
4. Méthode modifier (int i, Float note) (0.5 points)
5. Méthode Float get (int i) (0.5 points)
6. Méthode Afficher et Moyenne OÙ méthode size() (0.5 points)
7. Modification sur la méthode main () (1.5 points)

```
public class Etudiant {
    private String nom;
    private String prenom;
    private ArrayList<Float>notes;
    public Etudiant1(){
        this.nom = "";
        this.prenom = "";
        this.notes = new ArrayList<Float>();
    }
    public String getNom() { return nom;}
    public void setNom(String nom) { this.nom = nom;}
    public String getPrenom() { return prenom;}
    public void setPrenom(String prenom) { this.prenom = prenom;}
    public void ajouter(Float note){ this.notes.add(note);}
    public void modifier(int i, Float note){ this.notes.set(i, note);}
    public Float get(int i){ return this.notes.get(i);}
    public int size(){ return this.notes.size();}
}

public class Principale {
    public static void main(String[] args) {
        Etudiant etudiant2 = new Etudiant();
        etudiant2.setNom("nom2");
        etudiant2.setPrenom("prenom2");
        etudiant2.ajouter(new Float(12.0));
        etudiant2.ajouter(new Float(14.5));
        etudiant2.ajouter(new Float(8.25));
        etudiant2.modifier(1, new Float(14.0));
        Float moyenne = new Float(0.0);
        for(int i=0; i<etudiant2.size(); i++) moyenne = moyenne + etudiant2.get(i);
        moyenne = moyenne / etudiant2.size();
        System.out.println("Les notes de "+etudiant2.getNom()+" "+etudiant2.getPrenom()+": ");
        for(int i=0; i<etudiant2.size(); i++) System.out.println( etudiant2.get(i));
        System.out.println("L'etudiant "+etudiant2.getNom()+" "+etudiant2.getPrenom()+" a la moyenne: "+moyenne);
    }
}
```

Exercice 2 (09 points)

1. 6 classes et leurs attributs + héritage (02 points)
 2. Constructeurs (super dans deux classes) (01 points)
 3. Setters et Getters (1.5 points)
 4. Méthode void ajouter (Animal pet) (0.5 points)
-

5. Méthode void traiter (Animal pet)
 - instance of **(0.5 points)**
 - l'affichage **(0.5 points)**
6. Méthode void afficher ()
 - TreeSet et remplissage **(01 points)**
 - Affichage **(0.5 points)**
7. Classe Principale2
 - Création des objets **(0.5 points)**
 - Setters entre animaux et propriétaires **(01 points)**

```
public class Animal {
    private String nom;
    private int id;
    private String race;
    private Proprietaire pros;
    public Animal(String nom, int id, String race, Proprietaire pros){
        this.nom = nom;
        this.id = id;
        this.race = race;
        this.pros = pros; }
    public String getNom() { return nom;}
    public void setNom(String nom) { this.nom = nom;}
    public int getId() { return id;}
    public void setId(int id) { this.id = id;}
    public String getRace() { return race;}
    public void setRace(String race) { this.race = race;}
    public Proprietaire getPros() { return pros;}
    public void setPros(Proprietaire pros) { this.pros = pros;}
}
public class Chat extends Animal {
    private Boolean pelage;
    public Chat(String nom, int id, String race, Proprietaire pros, Boolean pelage){
        super(nom, id, race, pros);
        this.pelage = pelage;
    }
    public Boolean getPelage() { return pelage;}
    public void setPelage(Boolean pelage) { this.pelage = pelage;}
}
public class Chien extends Animal {
    private String taille;
    private String couleur;
    public Chien(String nom, int id, String race, Proprietaire pros, String taille, String couleur){
        super(nom, id, race, pros);
        this.taille = taille;
        this.couleur = couleur;
    }
    public String getTaille() {return taille;}
    public void setTaille(String taille) { this.taille = taille;}
    public String getCouleur() { return couleur;}
    public void setCouleur(String couleur) { this.couleur = couleur;}
}
public class Proprietaire {
    private String nom;
    private String adresse;
    private ArrayList<Animal> pets;
    public Proprietaire(String nom, String adresse){
        this.nom = nom;
        this.adresse = adresse;
        this.pets = new ArrayList<Animal>();
    }
    public void ajouter(Animal pet){ this.pets.add(pet);}
}
public class Veterinaire {
```

```
private String nom;
private String permis;
public Veterinaire(String nom, String permis){
    this.nom = nom;
    this.permis = permis;
}
public void traiter(Animal pet){
    if (pet instanceof Chat){
        System.out.println("Dr "+this.nom + " traite le chat "+ pet.getId());
    }
    if(pet instanceof Chien){
        Chien chien = (Chien)pet;
        System.out.println("Dr "+this.nom + " traite le chien "+ pet.getId()+ "au pelage "+chien.getCouleur()); } }
    public String getNom(){ return this.nom; }
}
public class Hopital {
    private ArrayList<Veterinaire> vets;
    private ArrayList<Proprietaire>pro;
    public Hopital(){
        this.vets = new ArrayList<Veterinaire>();
        this.pro = new ArrayList<Proprietaire>();
    }
    public void afficher(){
        TreeSet<String> st = new TreeSet<String>();
        for(int i=0; i<this.vets.size(); i++){
            st.add(this.vets.get(i).getNom());
        }
        System.out.println(st);}
}
public class Principal2 {
    public static void main(String[] args) {
        Animal chat = new Chat("chat1", 1, "siamois", null, true);
        Animal chien = new Chien ("chien2", 2, "labrador", null, "grand", "blanc");
        Proprietaire personne1 = new Proprietaire("nom1", "adresse1");
        Proprietaire personne2 = new Proprietaire("nom2", "adresse2");
        chat.setPros(personne1);
        chien.setPros(personne2);
        personne1.ajouter(chat);
        personne2.ajouter(chien);
    }}
}
```

Exercice 03 (03 points)

1. Définir les classes avec le mot clé *implement* (0.5 points)
2. Définir les attributs (0.5 points)
3. Redéfinition des méthodes (1.5 points)
4. Définition des constructeurs (0.5 points)

```
public class Rectangle implements Forme {
    private double a, b;
    public Rectangle(double a, double b){
        this.a = a;
        this.b = b;
    }
    public double surface() { return (a*b);}
    public double perimetre() { return 2 * (a +b);}
}
public class Cercle implements Forme {
    private double diametre;
```

```
public Cercle(double diametre){ this.diametre = diametre;}  
public double surface() { return (3.14 * this.diametre * this.diametre);}  
public double perimetre() { return (2 * 3.14 * this.diametre);}  
}
```

Exercice 04 (03 points)

Selon le programme. Un exemple de solution

```
public class Exercice {  
  
    public boolean trie(int [] a, int[] b){  
        int i;  
        for (i=0; i<b.length; i++){  
            if (!existe(a, b[i])) return false;  
        }  
        for (i=0; i<b.length-1; i++){  
            if (b[i] > b[i+1]) return false;  
        }  
        return true;  
    }  
  
    public boolean existe(int [] tab, int a){  
        for (int i = 0; i<tab.length; i++){  
            if (tab[i] == a) return true;  
        }  
        return false;  
    }  
}
```
