

Corrigé de l'interrogation écrite du 15 avril 2016

Exercice 1. Classe Chien – V 1.0.0

12 points

1) Combien de paramètres formels dans le constructeur normal ?	Nom, race, sexe, jour, mois, année, catégorie : 7
2) Comment déclarer et initialiser un tableau de 3 entiers (pour la date de naissance par exemple) ?	int [] tab = new int[3] ; tab[0] = jour ; tab[1] = mois ; tab[2] = annee ;
3) Pourquoi la méthode clone nécessite-t-elle un bloc try/catch ?	Il y a des exceptions dans le constructeur normal, appelé par la méthode clone
4) Enumérer les contrôles de validité des paramètres nécessaires dans le constructeur normal.	Vérifier le nom et la race : non null et non vide Vérifier sexe égal à F ou M Jour compris entre 1 et 31 Mois entre 1 et 12 Année <= année en cours
5) Que donne la ligne de code suivante : Calendar.getInstance().getTimeInMillis() ?	Donne la date du jour en ms
6) Donner le prototype de la méthode equals.	public boolean equals (Object o)
7) Soit la séquence de code suivante : Chien c1 = new Chien("Médor", "Chihuahua", 'M', 14, 3, 2016); Chien c2 = new Chien("Médor", "Chihuahua", 'M', 14, 3, 2016); Que donne le prédicat (c1 == c2) ?	False, car == compare les références
8) Donner le prototype et le code de la méthode toStringDate.	public static String toStringDate(int j, int m, int a) { return String.format("%1\$02d/%2\$02d/%3\$04d", j, m, a); }
9) Donner une séquence de code permettant de « cloner » un chien.	Chien c0= new Chien(); Chien cc0 = (Chien) c0.clone(); Chien c1 = new Chien("Medor", "Cocker", 'M', 3, 5, 2015, Catégorie.DEFENSE); Chien cc1 = (Chien) c1.clone();
10) Donner le prototype et le code de la méthode accouplementPossible.	public boolean accouplementPossible(Chien c) { if (!this.race.equals(c.race)) return false; if (this.sexe.equals(c.sexe)) return false; return true; }

```
public class Droite
{
    private Point A, B;

    public Droite ()
    {
        A = new Point();
        B = new Point(0.,1.);
    }

    public Droite (Point p1, Point p2) throws Throwable
    {
        if (p1.equals(p2)) throw new Throwable ("-1");
        A = (Point)p1.clone();
        B = (Point)p2.clone();
    }

    public boolean equals(Object o)
    {
        if (!(o instanceof Droite)) return false;
        Droite d = (Droite) o;
        return (d.appartient(this.A) && d.appartient(this.B));
    }

    public String toString()
    {
        return "[ " + A.toString() + ", " + B.toString() + " ]";
    }

    public Object clone()
    {
        Droite d = null;
        try { d = new Droite((Point)this.A.clone(), (Point)this.B.clone()); }
        catch(Throwable t){}
        return d;
    }

    public double pente() throws Throwable
    {
        double deltax = B.getAbscisse() - A.getAbscisse();
        double deltax = B.getOrdonnee() - A.getOrdonnee();
        if (deltax == 0) return 0;
        if (deltax == 0) throw new Throwable ("-1");
        return deltax/deltax;
    }
}
```