

TD 3 — Héritage, polymorphisme

Objectif(s)

- ★ mieux comprendre le concept de l'héritage et du
- ★ du polymorphisme

Exercice 1 – La notion d'héritage

Que fait le code suivant ?

```
class A {
    int numero;
    A(int n) {
        numero=n ;
    }
    public String toString() {
        return "mon numero est : "+numero;
    }
}

class B extends A {
    String nom;
    B(int n, String nom) {
        super(n);
        this.nom=nom;
    }
    public String toString() {
        return super.toString()+" et mon nom est "+nom ;
    }
}

class testAB {
    public static void main(String args[]) {
        A objet_a=new A(5);
        B objet_b=new B(6, "objet b");
        System.out.println(objet_a);
        System.out.println(objet_b);
    }
}
```

Exercice 2 – methodes de la classe Animal et Mammifere

On dispose de la classe Animal suivante :

```
class Animal {
    private float poids;
    private float taille;
    private boolean dents;
    private String nom;
    Animal(float p, float t, boolean d, String nom) {
        poids=p; taille=t; dents=d; this.nom=nom;
    }
}
```

```

    }
    public void set(float p, float t, Boolean d, String nom) {
        poids=p; taille=t; dents=d; this.nom=nom;
    }
    public void mange() {
        System.out.println(nom+" mange");
    }
    public String toString() {
        String res="Animal (" +nom+") de "+poids+"kg, de " +taille+" cm";
        if (dents) res+=" avec des dents\n";
    return res ;
    }
}

```

Et d'une classe Mammifere :

```

class Mammifere extends Animal
{
    private int nbreMamelles;
    private float rythmeCardiaque;
    ...
}

```

Question 1

La classe Mammifere a combien attributs ?

Question 2

Écrire le code du constructeur de la classe Mammifere dont la signature est :

```

    public Mammifere(float p, float t, boolean d, String nom, int n, float r)

```

p, t, d, nom, n et r correspondent respectivement au poids, la taille, la présence de dents, le nom, le nombre de mamelles et le rythme cardiaque d'un mammifère.

Question 3

Est-il nécessaire d'ajouter des getter et des setter dans la classe Animal pour permettre à la classe Mammifère d'accéder aux attributs nom, poids, taille, dents ? Justifiez votre réponse. Une solution fonctionnant sans getter et setter est-elle possible ?

Question 4

Redéfinir les méthodes clone et equals pour ces deux classes afin qu'elles réalisent leur rôle respectif.

Question 5

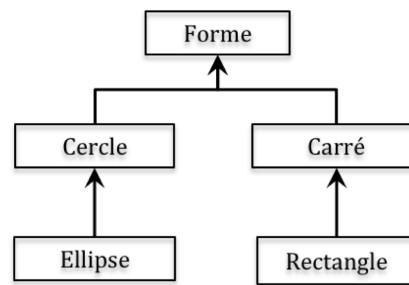
Ecrire une méthode pour tester les classes Animal et Mammifere en :

- créant un tableau d'animal contenant des animaux et des mammifères.
- ajoutant dans le tableau la copie d'un animal et la copie d'un mammifère

(À faire en TP) Tester ensuite votre méthode equals.

Exercice 3 – Formes géométriques

Réaliser les classes correspondant à l'arbre d'héritage suivant :



Une forme est décrite par les coordonnées réelles de son centre de gravité. Vous disposez d'une classe `Point2D`, avec un constructeur `Point2D(double x, double y)`, les getter et les setter, et la méthode `equals`. Les attributs supplémentaires sont pour :

- Un cercle : un rayon.
- Une ellipse : deux rayons
- Un carré : une longueur
- Un rectangle : une longueur et une largeur.

Question 1

Donnez la description des 5 classes en utilisant le minimum requis d'attributs par classe. Pour toutes les classes vous écrirez deux constructeurs (sans et avec paramètres) et une méthode `toString()`. Vous ajouterez ensuite une méthode permettant de calculer la surface de chaque forme. Pour finir, vous redéfinirez la méthode `equals`.

Vous donnerez aussi les accesseurs (getter) nécessaires.

Question 2

L'héritage indiqué ci-dessus (ellipse comme spécialisation de cercle et rectangle comme spécialisation de carré) n'est mathématiquement pas très logique. On souhaite donc inverser le rôle de Carré (Cercle) et Rectangle (Ellipse) dans l'arbre d'héritage ci-dessus, proposez les modifications nécessaires de ces deux classes pour que le résultat soit fonctionnel. Si la classe `Rectangle` avait des setters, la classe `Carré` pourrait les directement utiliser ou elle devrait les redéfinir ? Justifiez votre réponse !