

TD 2

1. Fonctions héritées

Implémenter la fonction **equals** pour les classes **Point**, **Segment** et **Cercle** vues en TD.

2. Cours

- Quelle est la différence entre une classe et un objet ?
- En quoi l'héritage est-il un outil puissant pour le développement d'applications ?
- Quelle est la différence entre
 - o `Point p[][] = new Point[5][4] ;`
 - o `Point p = new Point(5,4) ;`

3. Analyse de code

Soit

```
public abstract class Animal {
    public abstract void afficher();
}
public interface peutNager {
    public abstract void nage() ;
}
public class Poisson extends Animal implements peutNager{
    public void afficher() { System.out.println("Je suis un poisson");
    public void nage() {System.out.println("Je nage je suis un poisson") ;}
}
public class Mamifere extends Animal{
    public void afficher() { System.out.println("Je suis un mamifère");
}
Public class Baleine extends Mamifere implements peutNager() {
    public void nage() {System.out.println("Je nage je suis une baleine") ;
}
```

Parmi les appels suivant indiquez ceux qui sont corrects et ceux qui ne le sont pas, donnez le résultat de ce qui est correct.

Animal A1 = new Animal() ; Baleine B = new Mamifere() ;	Animal A2 = new Poisson() ; Poisson B = new Baleine() ;
Mamifere M1 = new Baleine() ; M1 = new Mamifere() ;	M1.afficher(); M1.afficher() ;
peutNager PN = new Poisson();	PN.afficher();
Poisson P = new peutNager();	peutNager Pp = new peutNager();
peutNager PN2[] = new peutNager[2] ; PN2[0] = new Poisson() ; PN2[0].afficher() ; PN2[1].afficher() ;	PN2[1] = new Baleine() ; PN2[0].nage() ; PN2[1].nage() ;

4. Analyse de code 2

Soit le code java suivant

```
Public PolygoneSegment(Point[] t) {
    segments = new Segment[t.length];
    for (int i=0; i<segments.length; i++) {
        segments[i] = new Segment(t[i], t[(i+1) % t.length]);
    }
}
```

Commentez le plus précisément possible le code (type de méthode, type des variables, fonctionnement, rôle des appels de fonction, ...).

Indiquez comment doit être déclaré l'objet « segments ».

5. Cours

- Expliquez les deux types de clonage que vous connaissez
- Expliquez la notion de polymorphisme

6. Clonage et liste chaînée

Soit A une classe implémentant l'interface Cloneable, et B une classe n'implémentant pas l'interface Cloneable, le code suivant est extrait de la méthode main de la classe test

```
LinkedList Liste = new LinkedList ();
```

```
Liste.add(new A());
```

```
Liste.add(new B());
```

```
LinkedList Liste2 = (LinkedList) Liste.clone();
```

1. Sachant que ce code ne renvoie pas d'erreur ni à la compilation ni à l'exécution, expliquez son fonctionnement et l'état de la mémoire.
2. Comment s'appelle le clonage réalisé par la classe LinkedList ?
3. Si on voulait réaliser l'autre type de clonage sur les LinkedList, quelle condition serait nécessaire sur les objets passés à la liste ?
4. Comment s'appelle le (LinkedList) devant Liste.clone() et pourquoi est-ce nécessaire dans le cas de la method clone().

7. Cours

Expliquez la notion d'Exception en java (fonctionnement, déclaration, ...).

8. Les exceptions

1. Définir une exception ExceptionNegatif. Cette classe aura simplement un attribut entier valeur et un constructeur avec un paramètre entier (correspondant à valeur).
2. Ecrire le code permettant de lever l'exception dans une méthode appelée *factoriel* si le paramètre passé à la méthode est négatif.
3. Ecrire le code permettant de récupérer l'exception et d'afficher la valeur (négative) ayant conduit à sa levée. On considère que la méthode *factoriel* se trouve dans la classe *Calcul*.

9. Segments et Exceptions

On veut protéger la classe Segment pour éviter que les deux points ne soient confondus (Segment réduit à un point).

Pour cela on définit une exception : PointsConfondusException.

1. Ecrire la classe PointsConfondusException
2. Ecrire un constructeur de la classe segment qui lève une exception si les points passés en paramètre sont confondus.
3. Ecrire la méthode « setP1 » qui modifie le point P1
4. Ecrire l'appel au constructeur de Segment (déclaration et instantiation d'un objet de type Segment