Objectifs :

- Construire des programme java en utilsant les structures de bases
- Utiliser les notions POO de java(classe , heritage ,interface..)
- Construire des applications multithreads

Exercice 1

Q1- Ecrire un programme java qui permet de calculer le salaire à partir du nombre d'heures et le prix par heure et lui ajoute une prime qui est de 5% de chiffre d'affaire si le salaire calculé est supérieur à 5000 et 3% dans les autres cas.
Q2- Ecrire un programme java qui demande un nombre de départ, et qui calcule le factoriel.
8 !=1 x 2 x 3 x 4 x 5 x 6 x 7 x 8
Q3- Ecrire un programme C# qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres :
Entrez le nombre numéro 1 : 12
Entrez le nombre numéro 2 : 14 etc.
Entrez le nombre numéro 20 : 6
Le plus grand de ces nombres est : 14
Modifiez ensuite l'algorithme pour que le programme affiche quelle position avait été saisie ce nombre : C'était le nombre numéro 2

Q4-

a- Écrire un programme qui effectue une division par zéro et ne contient aucun traitement d'exception. Que se passe-t-il? Pourquoi? Quel est le type de l'exception générée.

b- Cette fois, ré-écrire le programme pour capturer l'exception.

Exercice 2

1. Définir une classe **Elèves** caractérisée par : **Nom**, **Prénom** et **un tableau de trois notes**.
2. Définir les accésseurs
3. Définir un constructeur par défaut permettant d'initialiser le nom et le prénom par "Anonyme" et les trois notes par 0.
4. Définir un constructeur d'initialisation qui prend deux paramètres de type string et un tableau de réels permettant d'initialiser le nom, le prénom et le tableau des notes.
5. Définir un constructeur de recopie, permettant de cloner l'objet en cours.
6. Définir la méthode **Moyenne( )** qui retourne la moyenne de l'objet en cours.
7. Ajouter la méthode **Afficher( )** permettant d'afficher les informations d'un élève comme suit :

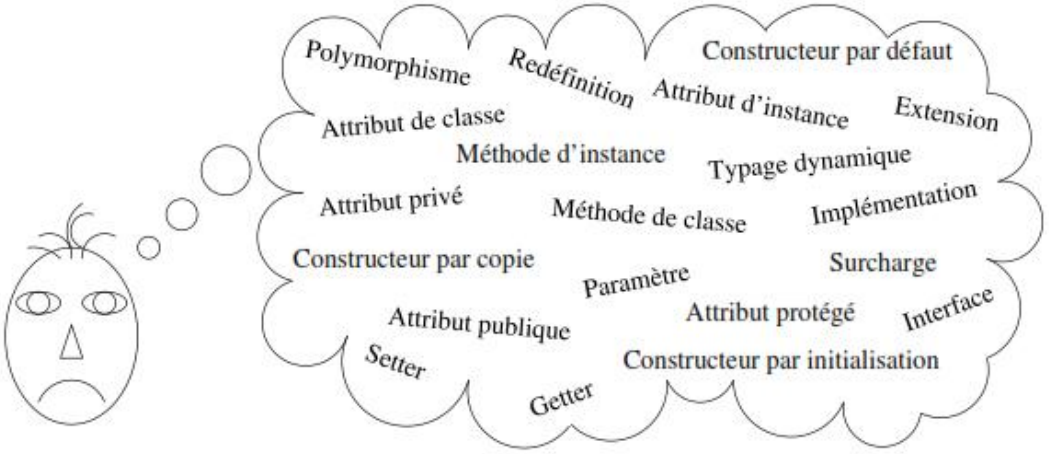
```
- Nom complet : [Nom Prénom] - Notes : - Note 1 : […] - Note 2 :
[…] - Note 3 : […] - Moyenne : […]
```

8. Écrire une classe permettant d'instancier 3 objets de la classe Élèves, le premier objet est

créé à l'aide du constructeur par défaut, les caractéristiques du deuxième sont demandées à l'utilisateur, le troisième est une copie du deuxième,stocker ces objet sur une liste de type Eleve et Afficher ensuite les informations de chaque élève.

Exercice 3

Un étudiant est un peu perdu avec toutes les notions vues en cours. En particulier, il confond les différents termes relatifs à la programmation objet.



Aidez-le à retrouver les termes correspondant aux parties entourées dans les classes suivantes :

```
class C1 {
  private int a;
  public double b;

  public C1() {
    a=0;
    b=1;
  }
  public C1(int a, double b) {
    this.a = a;
    this.b = b;
  }
  public C1(C1 c) {
    a = c.a;
    b = c.b;
  }
}

class C2 extends C1 {
  private String c;
  public C2(String c) {
    super();
    this.c = c;
  }
  public void afficher(int a) {
    System.out.println(a + c);
  }
}

class C3 extends C2 {
  private static int d = 0;
  public C3() {
  }
  public String toString() {
    System.out.println("C3");
  }
  public void toto() {
    C1 o = new C2();
  }
  public static int getD() {
    return d;
  }
}
```

(chacun des termes envisagés n'est utilisé qu'une fois, mais les cases peuvent contenir plusieurs termes)

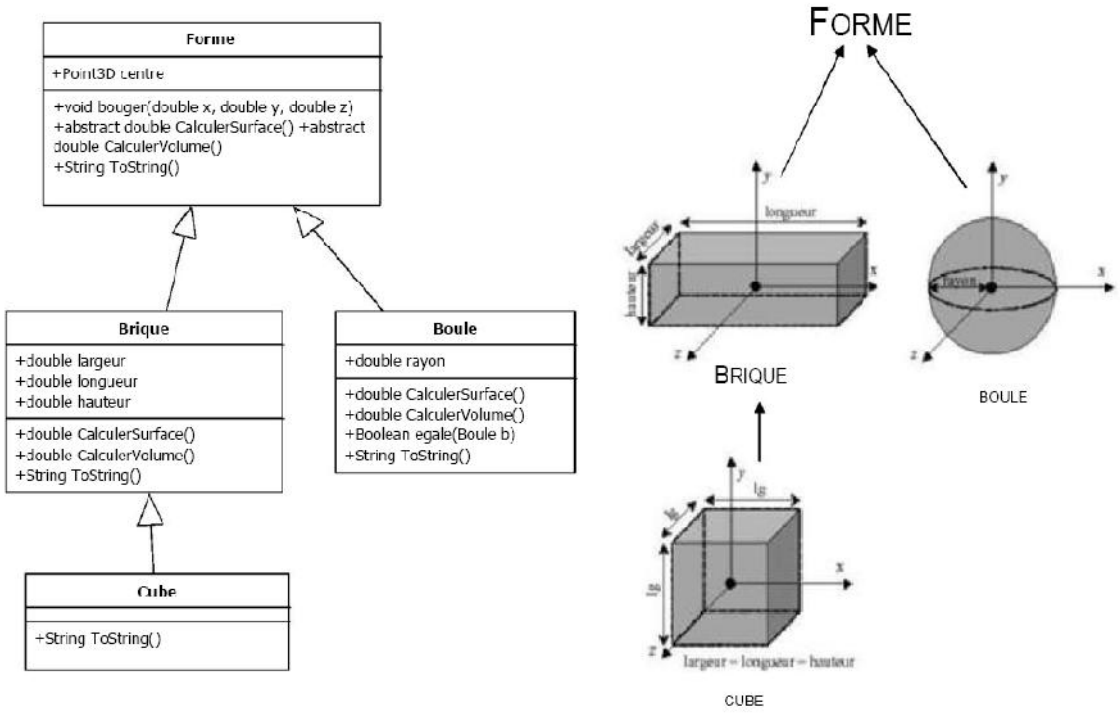
Exercice 5

Partie1

1- **Ecrire La classe Point3D caractériséé par 3 coordonnées x ,y et z .Définir les accesseurs ,et le constructeur d'initialisation et redéfinir les methodes ToString et Equals(égalité des coordonnés).(3 pts)**

2- **Ajouter à la classe une methode bouger prenant en paramètre troix réel dx ,dy et dz et qui permet de translater le point vers le point x+dx ,y+dy et z+dz. (1 pts)**

On va utiliser cette classe pour effectuer des dessins personnalisés avec des graphiques en 3D.



On souhaite disposer d'un ensemble de classes permettant de manipuler des formes tridimensionnelles. Pour cela on propose la hiérarchie de classes ci dessus.

- La classe Forme ne pourra pas être instanciée.
- La classe Cube ne pourra pas être dérivée.
- Chaque forme possède un attribut de type Point3D qui représente son centre de gravité et un attribut réel représentant sa densité.
- Un objet de type Boule est caractérisé par son centre de gravité,et son rayon.
- Un objet de type CylindrePlein est caractérisé par son centre de gravité, une hauteur et un rayon.
- Un objet de type Brique est caractérisé par son centre de gravité, une largeur, une longueur et une hauteur.
- Un objet de type Cube est une brique pour laquelle largeur = longueur = hauteur.

Les formes disposent d'une méthode **bouger** prenant comme paramètres trois réels représentant les composantes x, y et z d'un vecteur de translation et des méthodes **calculerSurface, calculerVolume** et calculant respectivement la surface, le volume .

.

1-Ecrivez le code java des classes Forme, Brique et Cube et Boule. (8 pts)

2-Deux Boules sont égales si elles ont même centre de gravité et même rayon , rédefinir la methode Equals pour la classe Boules. (1pts)

3- De plus, toute forme est capable de donner sa représentation sous la forme d'une chaîne de caractères contenant le nom de sa classe et la description textuelle de chacun de ses attributs (3pts)

Exemple : la chaîne de caractères produite pour un objet de classe Brique :

```
[Brique
  centre de gravité : [Point3D x :10.0 , y : 4.0, z : 3.0]
   largeur : 10.5
   longueur : 14.3
   hauteur : 4.6
]
```

Formules permettant de calculer l'aire et le volume d'une brique

Surface 2×(largeur×longueur + largeur×hauteur + longueur×hauteur).

Volume largeur*hauteur*longueur;

Formules permettant de calculer l'aire et le volume d'une boule

Surface $4\pi R^2$ avec R : rayon de la boule et

Volume $\frac{4\pi R^3}{3}$ $\pi = 3.14$

Partie 2

Changer la classe Forme en Interface Iforme et refaire l'exercice.
2. Ajouter l'héritage de l'interface Iforme de l'interface Icomprable.
Implimenter la methode Compareto pour toutes les classe formes.
Comparaison de 2 cercle se fait par la compariason des rayons.
Comparaison de 2 rectangle se fait par la compariason des largeurs longueurs et hauteurs.
3. Créer une classe de Test dans laquelle vous instancier 2 boule , 2 cercles et 2 cube.
a. Calculer la surface et le perimètre pour chaque forme.
b. Afficher les formes par type de plus petites au plus grandes.

Exercice 6

Q1- Un "compteur" a un nom (Toto par exemple) et il compte de 1 à n (nombre entier positif quelconque). Il marque une pause aléatoire entre chaque nombre (de 0 à 5000 millisecondes par exemple).

Un compteur affiche chaque nombre (Toto affichera par exemple, "Toto : 3") et il affiche un message du type "*** Toto a fini de compter jusqu'à 10" quand il a fini.

Ecrivez la classe compteur et testez-la en lançant plusieurs compteurs qui comptent jusqu'à 10. Voyez celui qui a fini le plus vite.

Q2- Modifiez la classe Compteur pour que chaque compteur affiche son ordre d'arrivée : le message de fin est du type : "Toto a fini de compter jusqu'à 10 en position 3"

Q3- Voici 2 classes `Compte` (correspond à un compte bancaire) et `Operation` (thread qui effectue des opérations sur un compte bancaire).

1. Examinez le code et faites exécuter la classe `Opération`. Constatez le problème : opération effectue des opérations qui devraient laisser le sode du compte inchangé, et pourtant, après un moment, le solde ne reste pas à 0. Expliquez.
2. Modifiez le code pour empêcher ce problème.

Compte.java

```
public class Compte {
  private int solde = 0;

  public void ajouter(int somme) {
    solde += somme;
    System.out.print(" ajoute " +
somme);
  }

  public void retirer(int somme) {
    solde -= somme;
    System.out.print(" retire " +
somme);
  }

  public void operationNulle(int
somme) {
    solde += somme;
    System.out.print(" ajoute " +
somme);
    solde -= somme;
    System.out.print(" retire " +
somme);
  }

  public int getSolde() {
    return solde;
  }
}
```

Operation.java

```
public class Operation extends
Thread {
  private Compte compte;

  public Operation(String nom,
Compte compte) {
    super(nom);
    this.compte = compte;
  }

   public void run() {
    while (true) {
      int i = (int) (Math.random() *
10000);
      String nom = getName();
      System.out.print(nom);
//        compte.ajouter(i);
//        compte.retirer(i);
      compte.operationNulle(i);
      int solde = compte.getSolde();
      System.out.print(nom);
      if (solde != 0) {
        System.out.println(nom +
":**solde=" + solde);
        System.exit(1);
      }
    }
  }

  public static void main(String[]
args) {
    Compte compte = new Compte();
    for (int i = 0; i < 20; i++) {
      Operation operation = new
Operation("" + (char)('A' + i),
compte);
      operation.start();
    }
  }
}
```