

TP numéro 5

Remarques :

- Tous les attributs des différentes classes doivent être privés.
- Evitez de dupliquer une partie du code.
- Si nécessaire, ajoutez des attributs et des méthodes.

Exercice 1

1. Définissez la classe « **Personne** » qui possède :
 - deux attribut privé, nommés « **nom** » et « **prenom** » de types String ;
 - un attribut privé, nommé « **DateNaissance** » de type String (qui doit être saisie sous la forme 23/12/1980) ;
 - un constructeur qui permet d'initialiser les différents attributs ;
 - une méthode **abstraite** « **description()** » qui servira à afficher le statut de la personne (Etudiant, Professeur, autre).
 - redéfinissez la méthode « **toString()** » pour retourner les différentes informations concernant une personne.
2. Définissez la classe « **Professeur** » qui hérite de la classe « **Personne** ». Ajoutez à la classe « **Professeur** » :
 - un attribut privé « **coursEnseigne** » (cours enseigné par le professeur, par exemple : POO-Java).
3. Définissez la classe « **Etudiant** » qui hérite de la classe « **Personne** ». Ajoutez à la classe « **Etudiant** » :
 - un attribut privé « **cne** ».
 - un tableau de type **double** qui contient 6 notes.
 - un tableau de type **boolean** prévu pour contenir les inscriptions aux modules.
 - une méthode « **moyenne()** » qui permet de calculer la moyenne des notes selon l'inscription aux modules.
 - une méthode « **mention()** » qui permet d'afficher la mention. Si l'étudiant est inscrit dans 6 modules, la méthode affichera la mention, sinon elle affichera le message : « Etudiant non inscrit dans 6 modules ».
4. Définissez la classe « **EtudiantTranger** » qui hérite de la classe « **Etudiant** ». Ajoutez l'attribut privé « **nationalité** ».
5. Redéfinissez dans les différentes sous classes, la méthode « **toString()** »
6. Définissez la classe « **SmiS6** » qui contient le tableau statique **modules**. Le tableau contient les noms des 6 modules.
7. Utilisez la classe « **GestionSMI** » pour les tests :
 - (a) Créez 5 objets de type « Personne » :
 - 4 étudiants , dont un étranger ;
 - 1 professeur (responsable d'un de module de S6).

Initialisez les différents attributs.

- (b) Pour les 4 étudiants :
 - le premier et le deuxième (étranger) sont inscrits dans 6 modules;
 - le troisième est inscrit dans les modules M1 et M2;
 - le quatrième est inscrit dans les modules M2, M3 et M5.
- (c) Saisissez les notes des différents étudiants et affichez la moyenne correspondante.
- (d) En utilisant la boucle (for - pour chaque élément) affichez les différentes notes des étudiants.
- (e) Affichez l'âge des différents étudiants. Pour obtenir l'année en cours, vous pouvez utiliser la classe « **GregorianCalendar** »

```
//Pour utiliser la classe GregorianCalendar
import java.util.GregorianCalendar;

//methode age
int age() {
    GregorianCalendar cal = new GregorianCalendar();
    //Annee en cours
    int anneeEnCours = cal.get(GregorianCalendar.YEAR);

    //A completer
}
```

- (f) Pour un module donné, listez les étudiants inscrit dans ce module.
- (g) Affichez les étudiants qui ont validé le semestre (affichés par ordre alphabétique du nom).

Exercice 2

1. Quelles sont les méthodes que doit implémenter une classe qui hérite de la classe **Calendar** et qui implémente l'interface **CharSequence** du package **java.lang**.

```
import java.util.Calendar;
import java.lang.CharSequence;

public class Exercice1 extends Calendar implements CharSequence{
    //declarations
}
```

2. Quel-est le problème avec la déclaration suivante :

```
public interface Interf {
    void Methode(int x){
        System.out.println("Test");
    }
}
```

3. Quel-est le problème avec la déclaration suivante :

```

interface A {
    void m();
}
class B implements A {
    void m() {
        System.out.println("m");
    }
}

```

4. Est-ce-que l'interface suivante est valide :

```

public interface Test {
}

```

Exercice 3

Mettez chaque classe et chaque interface dans un fichier.

1. Définissez les deux interfaces **Forme3D** et **FormeSymetrique** comme suit :

```

public interface Forme3D {
    double volume();
}

public interface FormeSymetrique{
    boolean axeX();
    boolean axeY();
    boolean axeXY();
}

```

2. Soient la classe **Point** et la classe abstraite **FormeGeometrique**, définies comme suit :

```

public class Point {
    private int x, y;
    // Ajoutez 2 constructeurs : un constructeur par default
    // et un autre de la forme Point(int,int)
    // Ajoutez des setters et des getters
    // Ajoutez une methode de copie (copie un point dans un autre)
}

public abstract class FormeGeometrique{
    private Point centre = new Point();
    //Ajoutez un constructeur de la forme FormeGeometrique(Point centre)
    abstract double surface();
}

```

3. Ajoutez à la classe **FormeGeometrique** une méthode qui permet déplacer une forme. Le centre (**x**, **y**) sera (**x+dxy**, **y+dxy**) avec **dxy** un entier.

4. Ajoutez la classe **Rectangle** :

- qui hérite de **FormeGeometrique**;
- implémente l'interface **FormeSymetrique**;

- aura 2 attributs privés (largeur et longueur) et un constructeur pour initialiser les attributs.
- 5. Ajoutez la classe **Cercle** qui :
 - hérite de **FormeGeometrique** ;
 - implémente l'interface **FormeSymetrique** ;
 - aura 1 attribut privé (**rayon**) et un constructeur pour initialiser cet attribut ;
 - aura la méthode « **diametre()** » qui permet de retourner le diamètre d'un cercle.
- 6. Ajoutez la classe **Carre** qui hérite de **Rectangle**. Elle aura 1 attribut privé (largeur) et un constructeur pour initialiser cet attribut.
- 7. Ajoutez une classe **BoiteCarre** qui hérite de la classe **Carre** et implémente l'interface **Forme3D**.
- 8. Ajoutez une classe **Cylindre** qui hérite de la classe **Cercle** et implémente l'interface **Forme3D**.
- 9. Redéfinissez la méthode « **toString()** » aux différentes classes pour afficher le type d'objet ainsi que ses attributs ; par exemple, pour le cercle, elle affichera :
 Cercle de centre (1, 5) et de rayon 10
 Diamètre : 20
 Surface : 314.15926
 Symétrique par rapport aux axes x et y.
 Pour les classes **BoiteCarre** et **Cylindre**, elle affichea aussi le volume.
- 10. Utilisez la classe **TestForme** pour les tests :

```

public class TestForme {
    public static void main(String[] args) {
        FormeGeometrique[] formes = new FormeGeometrique[3];
        //formes[0] <— contient un rectangle de largeur 7
        // et de longueur 8
        //formes[1] <— contient un carre de largeur 7,5
        //formes[2] <— contient un cercle de rayon 3
        //deplacez le rectangle de 2
        //deplacez le carre de 3
        //deplacez le cercle de 4
        //affichez les surfaces des differentes formes
        //affichez le diametre du cercle
        //creez deux objets (BoiteCarre et Cylindre)
        //affichez les informations concernant tous les objets
        //(2D et 3D)
    }
}

```

Exercice 4

Reprenez le TP3 et réorganisez les classes de la façon suivante :

1. Créez le package **banque.personnes** et mettez dans ce package les classes **Personne** et **Client**.
2. Créez le package **banque.comptes** et mettez dans ce package la classe **CompteBancaire**.
3. Dans le package **banque**, mettez la classe **Banque**.