

**Exercice 1 ( 10 points)**

Afin de modéliser en java, l'affichage de noms de fichiers et de dossiers, écrivez les classes suivantes :

- 1) L'interface « **FichierAbstrait** » possédant une méthode afficher() ;
- 2) La classe « **Fichier** » implémente «FichierAbstrait» et possède :
  - a. deux attribut privés nom de type String et retrait de type Retrait.
  - b. un constructeur avec deux paramètres String et Retrait
  - c. une méthode afficher(), qui affichera d'abord un retrait puis le nom du fichier
- 3) La classe « **Dossier** » implémente aussi FichierAbstrait et possède :
  - a. deux attribut privés nom de type String et retrait de type Retrait.
  - b. un attribut fichiers de type « ArrayList », pouvant contenir des dossiers et des fichiers.
  - c. un constructeur avec deux paramètres String et Retrait
  - d. une méthode ajouter (...), qui permettra d'ajouter au tableau « fichiers » des fichiers et des dossiers.
  - e. une méthode afficher(), qui affichera d'abord un retrait puis le nom du dossier. Ensuite, elle augmentera le retrait puis elle affichera tous les fichiers et les dossiers du tableau « fichiers ». Enfin elle diminuera le retrait.
- 4) La classe « **Retrait** » qui permettra de gérer le retrait selon le niveau du fichier ou du répertoire.

```
class Retrait {  
    private StringBuffer sbRetrait = new StringBuffer();  
    public String getRetrait () { return sbRetrait.toString(); }  
    public void augmenterRetrait () { sbRetrait.append("___"); }  
    public void diminuerRetrait () {  
        if (sbRetrait.length() >= 3) { sbRetrait.setLength(sbRetrait.length() - 3); }  
    }  
}
```

- 5) La classe principale « **Program** », qui permettra :
  - a. de créer un objet Retrait commun à tous les fichiers et dossiers
  - b. de créer cinq (05) objets « Fichier », et trois objets « Dossier » tels que :

Dossier	Contient
d1	f1
d2	f2, f3, d1
d3	f4, d2, f5

f1, f2, f3, f4 et f5 sont des objets « Fichier » dont les noms sont respectivement :  
a, b, c, d et e  
d1, d2 et d3 sont des objets « Dossier » dont les noms sont respectivement :  
dos1, dos2 et dos3

- c. d'afficher le dossier d3 :

```
dos3  
___d  
___dos2  
____b  
____c  
____dos1  
____a  
___e
```

## Exercice 2 (05 points)

Soient les classes suivantes :

```
class Etudiant {
    private int note ;           // 0 <= note <= 20
    public Etudiant ( int note ) {
        this.note = note ;
    }
}
```

```
class TestNote {
    public static void main ( String [ ] args ) {
        int n=0;
        n = Integer.parseInt(args[0]);
        Etudiant etudiant = new Etudiant ( n ) ;
        if (n>=10) System.out.println("Admis(e)");
        else System.out.println("Ajourné(e)");
    }
}
```

Complétez le programme ci-dessus pour que les erreurs susceptibles de se produire soient gérées.

- n** n'est pas une note valide (n entier, compris entre 0 et 20):
  - Créez la classe "**NoteException**"
  - Dans la classe Etudiant, levez (throw) NoteException
  - Dans la classe TestNote, lorsque par exemple n=22, attrapez (catch) cette exception et affichez le message suivant : **NoteException: 22 n'est pas une note valide** en redéfinissant toString() dans la classe NoteException.
- Aussi, gérez l'erreur s'il n'y a pas de paramètre sur la ligne de commande (args[0] n'existe pas) en précisant le type de l'exception et en affichant : **java.lang.ArrayIndexOutOfBoundsException: 0**

## Exercice 3 (05 points)

- Ecrivez une classe **MaFenetre** qui hérite **JFrame** et qui implémente **ActionListener**
  - Ajoutez dans cette classe deux (02) **JTextField** qui permettront de saisir du texte
  - Ajouter un **JButton** avec l'intitulé "Concaténer"
  - Ajoutez un **JLabel** qui permettra l'affichage de la concaténation
  - Fixez la taille de la fenêtre, par exemple à (400,100)
  - Alignez ces composants avec **new FlowLayout()**
  - En cliquant sur le bouton, les deux textes des JTextField seront concaténés et affichés par le JLabel
- Ecrivez la classe principale qui permettra d'instancier MaFenetre et de l'afficher.



**NB.** : JTextField est un composant qui permet l'édition de texte. Comme JLabel, le texte peut être modifié par setText() et retourné par getText().

Bon courage !

## Corrigé

### Exercice 1 ( 10 points)

```

public class Retrait {
    private StringBuffer sbRetrait = new StringBuffer();
    public String getRetrait () {return sbRetrait.toString();}
    public void augmenterRetrait () {sbRetrait.append("___");}
    public void diminuerRetrait () {
        if (sbRetrait.length() >= 3) {
            sbRetrait.setLength(sbRetrait.length() - 3);
        }
    }
}

```

---

```

public interface FichierAbstrait {
    public void afficher();
}

```

---

```

class Fichier implements FichierAbstrait {
    private String name;
    private Retrait retrait;
    public Fichier(String name, Retrait retrait) {
        this.name = name;
        this.retrait = retrait;
    }

    public void afficher() {
        System.out.println(retrait.getRetrait () + name);
    }
}

```

---

```

import java.util.ArrayList;
public class Dossier implements FichierAbstrait {
    private String name;
    private ArrayList<FichierAbstrait> fichiers = new ArrayList<FichierAbstrait>();
    private Retrait retrait;

    public Dossier (String name, Retrait retrait) {
        this.name = name;
        this.retrait = retrait;
    }
    public void ajouter(FichierAbstrait f) {
        fichiers.add(f);
    }
    public void afficher() {
        System.out.println(retrait.getRetrait () + name);
        retrait.augmenterRetrait ();
        for (FichierAbstrait fichier : fichiers) {fichier.afficher();}
        retrait.diminuerRetrait();
    }
}

```

---

```

public class Program {
    public static void main(String[] args) {
        Retrait retrait = new Retrait ();
        Dossier d3 = new Dossier("dos3", retrait);
        Dossier d2 = new Dossier("dos2", retrait);
        Dossier d1 = new Dossier("dos1",retrait);
        Fichier f1 = new Fichier("a", retrait);
        Fichier f2 = new Fichier("b", retrait);
        Fichier f3 = new Fichier("c", retrait);
        Fichier f4 = new Fichier("d", retrait);
        Fichier f5 = new Fichier("e", retrait);
        d1.ajouter(f1);
        d2.ajouter(f2);
        d2.ajouter(f3);
        d2.ajouter(d1);
        d3.ajouter(f4);
        d3.ajouter(d2);
        d3.ajouter(f5);
        d3.afficher();
    }
}

```

## Exercice 2 ( 05 points)

```
class Etudiant {
    private int note ; // 0 <= note <= 20
    public Etudiant ( int note ) throws NoteException {
        if (note <0 || note>20)
            throw new NoteException(note);
        this.note = note ;
    }
}
```

---

```
class NoteException extends Exception
{
    int valeur;
    NoteException(int val) { valeur = val; }
    public String toString() { return super.toString()+
        ":"+valeur + " n'est pas une note valide"; }
}
```

---

```
class TestNote {
    public static void main ( String [ ] args ) {
        int n = 0;
        try {
            n=Integer.parseInt(args[0]);
            Etudiant etudiant = new Etudiant ( n ) ;
            if (n>=10) System.out.println("Admis(e)");
            else System.out.println("Ajourné(e)");
        }

        catch ( ArrayIndexOutOfBoundsException e ) {
            System.out.println(e.toString () ) ;
        }

        catch ( NoteException e ) {
            System.out.println(e.toString () ) ;
        }
    }
}
```

---

### Exercise 3 ( 05 points)

```
import javax.swing.*; import java.awt.event.* ; import java.awt.* ;           0.5
class MaFenetre extends JFrame implements ActionListener
{
    JLabel l= new JLabel("");
    JButton b = new JButton ("Concaténer");
    JTextField t1=new JTextField("Bon");
    JTextField t2=new JTextField("jour");                                     0.5
    MaFenetre(){
        setTitle("Exercice 3");
        b.addActionListener(this);                                         0.5
        getContentPane().add(t1);
        getContentPane().add(t2);
        getContentPane().add(b);
        getContentPane().add(l);                                           0.5
        setSize (400, 100) ;                                               0.25
        getContentPane().setLayout(new FlowLayout());                     0.25
    }

    public void actionPerformed (ActionEvent ev){                          2
        l.setText(t1.getText()+t2.getText()) ;
    }
}

public class Principal{                                                    0.5
    public static void main (String args[]){
        MaFenetre fen = new MaFenetre () ;
        fen.setVisible(true) ;
    }
}
```