

**EXAMEN FINAL****02 heures-Documents interdits****Exercice 1 (3 pts) : Qu'affichent les codes suivants ?**

1/

```
class X {
public double g(double x) {
    return f(x) * f(x);
}
public double f(double x) {
    return x + 1.0;
}
}
class Y extends X {
public double f(int x) { return x + 2.0; }
public double f(double x) {
    return x + 3.0;
}
}
public class Exo1_1{
public static void main(String args[]){
    X x = new X();
    System.out.println(x.f(3.0));
    System.out.println(x.f(3));
    Y y = new Y();
    System.out.println(y.f(3.0));
    System.out.println(y.f(3));
    System.out.println(y.g(3.0));
    x = y;
    System.out.println(x.f(3));
    System.out.println(x.g(3.0));
}
}
```

2/

```
public class Exo1_2 {
    public static void uneMethode() { throw new RuntimeException(); }
    public static void main(String [] args) {
        try {
            uneMethode();
            System.out.print("A");
        }
        catch (Exception ex){
            System.out.print("B");
        }
        finally{ System.out.print("C"); }
        System.out.print("D");
    }
}
```

**Exercice 2 (5 pts) :**

La figure Fig.1 représente le fichier Test.java contenant un programme java. Lisez-le attentivement et examinez les instructions soulignées afin de déterminer pour chacune d'elles si elle comporte des erreurs ou pas. Justifiez vos réponses.

```
1  import java.util.*;
2  public final class A {
3      private final int r;
4      protected A (int i) { r = i; }
5      public static A create(int i) {
6          if (i < 0) return null;
7          return new A(i);
8      }
9      protected Byte getR() {return r;}
10 }
11 abstract final class B extends A {
12     private B(int i){super(i);}
13     public static B create(int i, int j) {
14         int k = i + j;
15         return new A(k);
16     }
17 }
18 public class Test {
19     public int somme(List<A> liste) {
20         int s = 0;
21         for(A a: liste) s+= a.getR();
22         return s;
23     }
24     public static void main (String[] args) {
25         Set<A> s = new ArrayList<A>();
26         s.add(A.create(1));
27         s.add(new B());
28         int somme = somme(s);
29     }
30 }
```

Fig1. Test.java

**Exercice 3 (5 pts) :**

Une entreprise souhaite créer une boîte à idées électronique dans le but de recueillir les idées et suggestions de ses salariés. Pour cela, elle met à leur disposition un formulaire sur son site internet contenant trois champs : nom du salarié, objet du message (son titre) et texte du message (son contenu). Le salarié peut aussi ajouter des pièces jointes (photos, documents, vidéos...etc.). Après la validation du formulaire, un message est créé et ajouté à la liste des messages contenus dans la boîte. Pour réaliser ce programme, nous proposons les deux classes suivantes :

**Classe Boite.** Elle est identifiée par le nom de l'entreprise et la liste des messages qu'elle contient triés selon l'ordre de leur réception.

**Classe Message.** Elle encapsule les attributs et comportement d'un objet de type message. Son code java incomplet est donné ci-dessous.



1. Quelle collection est la plus appropriée pour contenir la liste des messages de sorte à garantir l'affichage des messages non lus en premier puis les messages lus, et en affichant dans les deux cas les plus récents en premier grâce à leurs identifiants? Donnez l'instruction permettant de la déclarer.
2. Complétez le code de la classe Message suivante :

```
import java.util.*;

public class Message implements [redacted] { // Elément 1 à compléter

    // Elément 2 à compléter : Liste des attributs
    [redacted]

    public Message(String nomSalarie, String titre, String texte) {
        this.objet=titre;
        this.contenu=texte;
        this.emetteur=nomSalarie;
        this.lu=false ;
        this.piecesJointes=new HashMap<String, Double>();
        this.id=i;
        i++;
    }

    public void lire(){
        this.lu=[redacted]; // Elément 3 à compléter
        this.afficher();
    }

    public void afficher(){
        // Elément 4 à compléter
        [redacted];
    }

    /* Elément 5 à compléter : Méthode permettant d'ajouter un fichier
    aux pièces jointes du message et de lancer une erreur si la taille
    totale des pièces jointes dépasse 100Mo */

    public void ajouterPieceJointe (String nomfichier, double
    taille) [redacted] {
        double tailleTotal=this.calculerTailleTotale() ;
        if ((tailleTotal+taille)>100) [redacted];
        else piecesJointes.[redacted]; //
    }

    /* Elément 6 à compléter : Donnez les deux autres méthodes
    nécessaires au bon fonctionnement de la classe*/
    [redacted]
}
```



**Exercice 4 (7 pts) :**

Pour assurer les déplacements de ses employés chez eux, une compagnie pétrolière dispose d'un bureau de voyages qui s'occupe de la réservation d'avions chez une compagnie aérienne.

Les agents du bureau de voyages ont accès à la liste des employés à partir de laquelle ils sélectionnent les employés ayant terminé leur période de travail de 4 semaines et résidant dans la ville de destination du vol. A la fin du processus, la liste des employés sélectionnés est extraite et communiquée à la compagnie aérienne. En plus de ce processus d'extraction des listes de vols, les agents du bureau de voyages peuvent consulter les informations des vols en sélectionnant un vol pour voir s'afficher ses informations (Numéro du vol, date et heure, destination et liste des employés voyageurs). Le bureau de voyages est géré par un chef de service qui intervient en cas de changement des informations des vols. Ce changement est communiqué par la compagnie aérienne au chef de service qui est le seul à avoir le droit de mettre à jour les informations. En plus de ce droit de modification, le chef de service possède tous les droits que possèdent les agents du bureau (la consultation des vols et l'extraction des listes de vol). Il peut également consulter pour un employé donné, l'historique de tous les vols qu'il a pris depuis son embauche dans l'entreprise.

Afin de faciliter le travail du bureau de voyages, on désire concevoir une application permettant d'effectuer les différentes tâches. L'application peut être utilisée par le chef de service, les agents du bureau et les employés à travers un compte qui donne accès à chaque utilisateur aux fonctionnalités auxquelles il a droit. A travers leurs comptes, les employés peuvent introduire les dates de début et de fin de leur période de travail courante et s'informer sur leur prochain vol. Le compte d'un utilisateur de l'application est décrit par un nom de compte et un mot de passe et chaque employé est caractérisé par son nom, son prénom, sa ville de résidence et la date de début de sa période de travail actuelle.

**Question.** Proposez une conception orientée objet de votre solution. Pour cela il faut :

1. Tracer un diagramme des classes illustrant les relations entre les classes.
2. Décrire chacune des classes à l'aide d'un tableau en définissant les différentes classes, leurs attributs et leurs méthodes.

Nom de la classe			
Liste des attributs			
Modificateur d'accès	Type de l'attribut	nom de l'attribut (donnez un nom significatif)	
Liste des méthodes			
Modificateur d'accès	Valeur de retour	Signature de la méthode	description