

## Contrôle Intermédiaire

Durée 02 heures

### Documents interdits

#### Questions de cours :

1. Un étudiant inscrit en première année à l'ESI vous demande de lui expliquer brièvement la différence entre la programmation procédurale et la programmation orientée objet. Que lui répondez-vous ?
2. Quelle est la différence entre surdéfinition et redéfinition ?
3. Dans le cadre des collections Java, quelles sont les classes utilitaires ?

#### Exercice 1. Conception d'un agenda électronique

On désire concevoir un agenda électronique permettant à son utilisateur de gérer son emploi du temps et ses contacts.

L'utilisateur caractérisé par son nom et son prénom peut introduire trois types d'éléments dans son agenda: une tâche, un rendez-vous ou un contact. Un élément de l'agenda peut être ajouté, supprimé ou modifié et il doit être toujours possible d'afficher ses propriétés.

Une tâche peut être régulière ou occasionnelle. Elle est définie par un nom, une heure de début et une durée. Si la tâche est régulière, l'utilisateur spécifie le jour de la semaine à laquelle elle se déroule ainsi que la date de début et la date de fin (Exemple : Cours de POO tous les lundis à 13h00 du 12/02/2012 au 27/05/2012). Dans le cas où la tâche est occasionnelle, l'utilisateur doit fournir la date et l'heure précise à laquelle elle doit être effectuée (Exemple : Contrôle intermédiaire de POO le 14/04 à 14h00).

Un rendez-vous est défini par un libellé, une date, une heure de début, une durée et les contacts avec lesquels le rendez-vous est convenu.

Tous les contacts sont caractérisés par un nom et peuvent être de deux types : Personne ou organisation. Dans le cas où le contact est une personne, l'utilisateur fournit une adresse email et spécifie si ce contact est de type privé ou professionnel. Pour un contact de type privé, l'utilisateur ajoute son numéro de portable et pour un contact de type professionnel, il saisit le nom de l'entreprise dans laquelle il travaille. Si le contact est une organisation, celle-ci est caractérisée par une adresse et son site internet.

#### Questions :

1. Proposer une modélisation en indiquant les différentes classes (Préciser si une classe est abstraite), leurs attributs et leurs méthodes, et indiquer les relations entre les classes (héritage ou utilisation).
2. Nous supposons que certaines tâches occasionnelles ou rendez-vous peuvent être retardés d'un certain nombre de jours fourni par l'utilisateur. Proposer une solution sans ajouter d'attributs aux classes existantes.
3. Nous supposons qu'une tâche possède un niveau de priorité. Celui-ci est un entier égal à 1 si la priorité est élevée, à 2 si la priorité est moyenne et à 3 si elle est basse. Proposer une solution permettant de classer les tâches selon leur priorité. Ecrire son code Java.

## UEF4.3. Programmation Orientée Objet

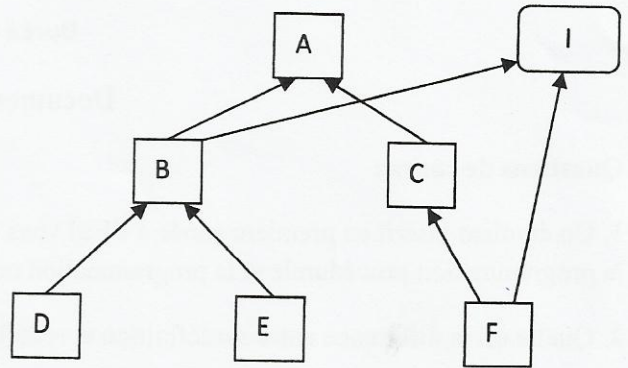
**Exercice 2.** Soit la hiérarchie de classes illustrée dans la figure ci-dessous:

A, B, C, D, E et F sont des classes alors que I est une interface implémentée par les classes B et F.

**Questions :**

A/ Dans la suite d'instructions suivante, indiquez pour chacune d'elles:

- 1) si elle est correcte ou si elle provoque une erreur à la compilation.
- 2) si on peut résoudre le problème de la compilation, et si oui, s'il reste une erreur à l'exécution ou non.



```

A a1 = new A();
I a2 = new E();
I a3 = new F();
I a4 = new C();
C a5 = new F();
a5 = a1;
a1 = a5;
B a6 = new B();
a5 = a6;
E a7 = a6;
  
```

**NB.** Répondre en remplissant le tableau suivant :

Instruction	Compilation Ok/ Erreur	Correction	Exécution Ok/Erreur
.....	.....	.....	.....

**B/** Nous ajoutons les classes Alpha et Beta suivantes :

```

Class Alpha{
public void f(A a) {...}
}
class Beta extends Alpha{
public void f(A a) {...}
public void f(B b) {...}
public void g(float x, B b) {...}
public void g(int n, A a) {...}
}
  
```

```

Alpha o1 = new Alpha(); Beta o2 = new Beta();
A a = new A(), B b = new B(); int i; float y;
  
```

```

o1.f(b);
o2.f(b);
o1 = o2;
o1.f(b);
o2.g(i, a);
o2.g(i, b);
o2.g(y, b);
o2.g(y, a);
  
```

Parmi les instructions suivantes, indiquer celles qui provoquent une erreur à la compilation et préciser pour les instructions correctes la méthode qui sera appelée.