

## LO43 : Programmation Orientée Objet - Final Printemps 2010

Durée : 2h. Documents non autorisés, Traducteurs autorisés, Les réponses de la partie 1 se feront sur le sujet.

NOM :

Signature :

PRENOM :

### Partie 1 : Programmation Orientée Objet

#### 1 Questions de cours :

Question 1 :

Pourquoi le profil de la méthode main est-il :  
`public static void main(String[] arg)` ?

Réponse :

public :

static :

void :

Question 2 :

Qu'est ce que l'autoboxing des types primitifs ?

Réponse :

---

**Question 3 :**

Sur quelle notion se basait la définition et l'utilisation des conteneurs (List, Vector,...) avant Java 1.5 ? Comment est-ce fait maintenant ?

**Réponse :**

## 2 Exercice : Statiquement abstrait

Qu'affiche exactement le programme suivant ?

```
class AbstractStatic {
public static void main(String[] args) {
N un = new N();
N deux = new N(true);
Z trois = new Z();
Z quatre = new Z(false);
Q cinq = new Q();

un.m1();
deux.m1();
deux.m3();
trois.m1();
trois.m3();
quatre.m3();
cinq.m2();

System.out.println(un.n);
System.out.println(un.b);
System.out.println(un.i);
System.out.println(deux.n);
System.out.println(deux.b);
System.out.println(deux.i);
System.out.println(trois.n);
System.out.println(trois.b);
System.out.println(trois.o);
System.out.println(quatre.n);
System.out.println(quatre.b);
System.out.println(quatre.o);
System.out.println(cinq.n);
System.out.println(cinq.d);
} }
```

```
abstract class C {
static int n = 0;

C() {
n++; }

C(boolean b) {
if (b) {
n+=2; } } }

abstract class R extends C {
boolean b = false;

R() {
super(); }

R(boolean b) {
super(b);
n++; }

abstract boolean m1();
}
```

```
class Q extends C {
    int d = 0;
```

```
    Q() {
        super(true); }

    boolean m1() {
        if (d == 1) {
            return true; }
        d++;
        return false; }

    void m2() {
        if (d == 2) {
            d++; } }
    }
```

```
class N extends R {
    int i = 1;
```

```
    N() {
        super();
        b = false; }

    N(boolean b) {
        super(b);
        n++;
        i++; }

    boolean m1() {
        if (b) {
            b = false; } else { b = true; }
        return !b; }

    void m3() {
```

```
        i++; }
    }
```

```
class Z extends R {
    int o;
```

```
    Z() {
        super();
        o++; }

    Z(boolean b) {
        super(b);
        o = 4; }

    boolean m1() {
        return !b; }

    void m3() {
        o++; } }
```

```
    boolean m1() {
        return !b; }

    void m3() {
        o++; } }
```

```
    void m3() {
        o++; } }
```

```
    void m3() {
        o++; } }
```

---

---

**Réponse :**

---

---

## Partie 2 : Conception Orientée Objet

### Exercice 1 : Eh hop ! dans l'avion pour les vacances

On souhaite gérer les réservations de vols effectués dans une agence. D'après les interviews réalisées avec les membres de l'agence, on sait que :

- Les compagnies aériennes proposent différents vols
- Un vol est ouvert à la réservation et refermé sur ordre de la compagnie
- Un client peut réserver un ou plusieurs vols, pour des passagers différents
- Une réservation concerne un seul vol et un seul passager
- Une réservation peut être confirmée ou annulée
- Un vol a un aéroport de départ et un aéroport d'arrivée

- Un vol a un jour et une heure de départ, et un jour et une heure d'arrivée
- Un vol peut comporter des escales dans un ou plusieurs aéroport(s)
- Une escale a une heure de départ et une heure d'arrivée
- Chaque aéroport dessert une ou plusieurs villes

**Question :**

1. **A partir des éléments qui vous sont fournis ci-dessus, élaborer le diagramme de classes (en y ajoutant tout attribut que vous jugez pertinent et qui n'a pas été décrit ci-dessus).**

## **Exercice 2 : Un petit peu de (Viti)Culture pour le voyage**

Dans le cadre d'un projet de recherche en viticulture, on désire collecter les temps de travaux sur des exploitations agricoles pilotes, pour travailler en particulier sur les opérations phytosanitaires. Pour ce faire, un glossaire des opérations culturales types a été mis en place (afin que tout le monde ait le même cadre analytique). Des contraintes assez fortes sont apparues sur le projet : les ouvriers agricoles des exploitations pilotes n'ont pas accès aux outils informatiques et la lourdeur d'enregistrement des temps de travaux a donc de fait écarté l'utilisation d'un outil informatique.

La procédure suivante a ainsi été définie :

- Chaque ouvrier agricole saisit ses temps de travaux sur un cahier au format prédéfini. (dans ce cahier, il peut consulter en annexes le glossaire afin d'identifier l'opération culturale type<sup>1</sup>).
- En fin de mois, le chef d'exploitation vérifie la saisie effectuée sur le cahier et apporte d'éventuelles corrections.
- Il saisit ensuite les opérations du mois sur une application internet connectée à une base de données.
- Le chercheur en charge du projet reçoit automatiquement un mail qui lui indique que la saisie mensuelle a été effectuée.
- Après avoir vérifié la pertinence de la saisie, il notifie au chef d'exploitation que tout s'est bien passé et que les données intégrées dans la base de données sont valides et prêtes à être exploitées.
- Le chef d'exploitation imprime alors 2 documents sur le mois écoulé :
  - L'état mensuel des travaux pour chaque salarié (qui est remis à chaque salarié)
  - L'état des opérations phytosanitaires (état Terravitis)
- En fin d'année, le chercheur analyse toutes les opérations saisies et rédige une synthèse générale sur les temps de travaux dans les différentes exploitations.
- Cette synthèse est alors transmise à tous les chefs d'exploitation.

**Question :**

1. **Proposez les diagrammes nécessaires (cas d'utilisation, diagramme de classe, diagramme de séquence,...) pour la description de cette application**

**Remarques :**

*N'oubliez pas de commenter vos diagrammes. La notation tiendra autant compte de la qualité de vos diagrammes et du respect du formalisme UML que des commentaires qui les accompagnent.*

---

1. Pour les opérations de type phytosanitaire, les informations complémentaires sont demandées : liste des maladies visées, stade phénologique, méthodes de traitements et observation.

RELEVÉ MENSUEL POUR Strubel Frédéric pour le mois de Avril 2006

Jour	Operations	Total heures
1		0 h
2		0 h
3	<a href="#">Taille</a> : 24.00 h	24 h
4	<a href="#">Taille</a> : 15.00 h	15 h
5	<a href="#">Taille</a> : 24.00 h	24 h
6	<a href="#">Tirage des bois</a> : 24.00 h	24 h
7	<a href="#">Phage/Liage</a> : 32.00 h	32 h
8		0 h
9		0 h
10	<a href="#">Phage/Liage</a> : 24.00 h	24 h
11	<a href="#">Phage/Liage</a> : 16.00 h	16 h
12	<a href="#">Decavaillonnage mecanique</a> : 5.00 h	5 h

FIGURE 1 – Exemple d'un état mensuel

LISTE DES INTERVENTIONS

No intervention	Parcelles	Operateur	Date	Operation	Duree	Matériel	Maladies visées (stade pheno)	Méthode de traitement	Observations
<a href="#">Modifier</a>	Merlots, Blancs, Petits verdoles et Cabernets Franc	Pichot Yann	11-05-2006	Traitement phytosanitaire	9.00	Enjambeur Bobard 896 Pulvé Hardy	Mildiou Oïdium (?)	Face / Face Localisé	5 rangs traités par passage
<a href="#">Modifier</a>	Cabernets Sauvignons	Strubel Frédéric	17-05-2006	Traitement phytosanitaire	8.00	Enjambeur Bobard 896 Pulvé Hardy	Mildiou Oïdium (?)	Face / Face	5 rangs traités par passage
<a href="#">Modifier</a>	Merlots, Blancs, Petits verdoles et Cabernets Franc	Pichot Yann	26-05-2006	Traitement phytosanitaire	8.00	Enjambeur Bobard 896 Pulvé Hardy	Mildiou Oïdium (?)	Face / Face	stade 21 / 5 rangs traités par passage
<a href="#">Modifier</a>	Cabernets Sauvignons	Pichot Yann	02-06-2006	Traitement phytosanitaire	8.00	Enjambeur Bobard 896 Pulvé Hardy	Mildiou Oïdium (?)	Face / Face	stade 21 / 5 rangs traités par passage
<a href="#">Modifier</a>	Parcelles de multiplication	Pichot Yann	06-06-2006	Traitement phytosanitaire	0.00	Enjambeur Bobard 896 Pulvé Hardy	Fd (?)	Face / Face	stade 21 / 5 rangs traités par passage

FIGURE 2 – Exemple d'impression Terravitis

