

IFT1020 - Session Été, Final

Mohamed Lokbani

**IFT1020 - FINAL**

Inscrivez tout de suite votre nom et code permanent.

Nom: \_\_\_\_\_ | Prénom(s): \_\_\_\_\_

Signature: \_\_\_\_\_ | Code perm: \_\_\_\_\_

Date : 28 juillet 2005

Durée: 3 heures (de 17h30 à 20h30) Local: 1360 du Pavillon A.-AISENSTADT.

**Directives:**

- Toute documentation permise.
- Calculatrice **non** permise.
- Répondre directement sur le questionnaire.
- Les réponses **doivent être brèves, précises et clairement présentées.**

1. \_\_\_\_\_/22 (1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7)

2. \_\_\_\_\_/12 (2.1, 2.2, 2.3)

3. \_\_\_\_\_/12 (3.1, 3.2, 3.3, 3.4)

4. \_\_\_\_\_/16 (4.1, 4.2, 4.3, 4.4)

5. \_\_\_\_\_/13 (5.1, 5.2, 5.3, 5.4)

6. \_\_\_\_\_/25 (6.1, 6.2, 6.3, 6.4)

Total: \_\_\_\_\_/100

**Directives officielles**

\* Interdiction de toute communication verbale pendant l'examen.

\* Interdiction de quitter la salle pendant la première heure.

\* L'étudiant qui doit s'absenter après la première heure remet sa carte d'étudiant au surveillant, l'absence ne devant pas dépasser 5 minutes.

\* Un seul étudiant à la fois peut quitter la salle.

\* Tout plagiat, copiage ou fraude constitue une infraction et l'étudiant qui commet ce délit se voit attribuer la note F par le professeur.

F.A.S

**Question 1 (22 points)**

**1.1** Dans Java, si une exception est de la catégorie « Checked », ceci signifie .....  
.....  
.....

**1.2** Quand la méthode « repaint() » est appelée dans une « Frame », ceci a comme incidence que la « Frame » est tout de suite redessinée (VRAI/FAUX) ..... car  
.....  
.....

**1.3** Donner deux exemples d'un « Layout Manager » en Java : 1-....., 2-.....

**1.4** Vous voulez réaliser une application de suivi de rendez-vous (calendrier électronique) pour votre agenda électronique. Cette application va vérifier en permanence l'état de vos rendez-vous, en vous alertant s'il y a lieu que vous avez des choses à faire. Parmi les collections suivantes, laquelle est la plus appropriée pour représenter les données de votre application de manière efficace: « **Liste, Table de hachage, Pile, Queue, Queue Prioritaire** » et dites pourquoi.

**1.5** Vous voulez construire une application de fouille pour collecter des adresses de courriel. Cette application va scanner votre disque dur à la recherche de toutes les adresses de courriel se trouvant sur votre disque. Par la suite elle va se charger d'envoyer les courriels recueillis à une tierce personne. Parmi les collections suivantes « **List, Hashtable, HashSet, Array, Map** », laquelle est la plus appropriée pour contenir les données collectées et dites pourquoi.

**1.6** Après compilation du fichier « `examen.java` », parmi les instructions suivantes, laquelle va permettre de « l'exécuter » et pourquoi (expliquez aussi pourquoi les autres possibilités sont incorrectes)

- a) `java Examen`
- b) `javac examen`
- c) `java examen.java`
- d) `java examen`
- e) `java examen.main()`

**1.7** Expliquez l'intérêt de définir la méthode « `toString` » dans une classe.

**Question 2 (12 points)** Soit le programme suivant qui **compile correctement**

```
public class Q2 {
    public static void main(String[] args){
        if( args.length > 0 ) {
            convert( args[0] );
            System.out.print("Bon ");
        }
        else {
            System.out.print("Mauvais ");
        }
        System.out.println( nombre );
    }
    static float nombre = 0 ;
    static boolean convert( String s ){
        try {
            nombre = Float.parseFloat( s );
            return true ;
        }catch (NumberFormatException e ){
            nombre = Float.NaN ;
            return false ;
        }
    }
}
```

**Tout en justifiant votre réponse**, dites ce qui sera affiché en sortie si le programme est exécuté avec les commandes suivantes :

2.1) `java Q2`

2.2) `java Q2 1-2`

2.3) `java Q2 7.5 1-2`

**Question 3 (12 points)** « **DataVitaless** » est le nom d'une classe faisant partie d'une série de classes que vous avez rassemblées dans le paquetage « Unis ». Ce paquetage sera utilisé par d'autres programmeurs dans votre compagnie. Vous ne voulez pas que ces programmeurs puissent étendre par héritage la classe « **DataVitaless** ». Par ailleurs, vous voulez autoriser aussi l'accès à « **DataVitaless** » aux classes se trouvant dans d'autres paquetages. **Comment doit-on déclarer cette classe?**

Traiter chaque sous question **à part de manière indépendante**. Les explications données pour une sous question ne doivent pas dépendre des réponses données dans les autres sous questions.

**3.1) protected final class DataVitaless**

Correct  
Pourquoi

Incorrect

**3.2) public final class DataVitaless**

Correct  
Pourquoi

Incorrect

**3.3) public abstract class DataVitaless**

Correct  
Pourquoi

Incorrect

**3.4) public static class DataVitaless**

Correct  
Pourquoi

Incorrect

**Question 4 (16 points)** « C1 » et « C2 » sont des références à deux collections. Pour chacune des instructions suivantes, les affirmations associées (signalées en commentaire) sont-elles correctes?

**4.1) `C1.retainAll(C2)`; // Elle ne modifie pas le contenu de « C1 »?**

Oui  
Pourquoi

Non

**4.2) `C1.removeAll(C2)`; // Elle ne modifie pas le contenu de « C1 »?**

Oui  
Pourquoi

Non

**4.3) Pour: `C2.retainAll(C1)`; `C1.containsAll(C2)`; // La 2<sup>e</sup> instruction va retourner « true »?**

Oui  
Pourquoi

Non

**4.4) Pour: `C2.addAll(C1)`; `C1.retainAll(C2)`; // La 2<sup>e</sup> instruction n'aura aucun effet sur « C1 »?**

Oui  
Pourquoi

Non

**Question 5 (13 points)** Soit une description partielle d'une hiérarchie d'exceptions qui pourront être levées durant des opérations E/S sur des fichiers.

```
Exception
+----IOException
+-----FileNotFoundException
```

Soit la méthode « **X** » qui est supposée ouvrir un fichier à partir de son nom pour lire son contenu. Sachant que la méthode « **X** » n'a aucun bloc « **try-catch** », dites si les options suivantes sont correctes ou incorrectes, tout en expliquant votre choix de réponse.

**5.1)** La méthode « **X** » doit être déclarée comme levant l'exception « **IOException** » ou bien « **Exception** ».

Correct  Incorrect  
Pourquoi

**5.2)** La méthode « **X** » doit être déclarée comme levant l'exception « **FileNotFoundException** ».

Correct  Incorrect  
Pourquoi

**5.3)** N'importe quelle méthode faisant appel à la méthode « **X** » doit utiliser « **try-catch** » pour spécifiquement capturer l'exception « **FileNotFoundException** ».

Correct  Incorrect  
Pourquoi

**5.4)** Aucune précaution n'est nécessaire lors de la déclaration ou l'utilisation de la méthode « **X** ».

Correct  Incorrect  
Pourquoi

**Question 6 (25 points)** On suppose pour cette question que la classe « **Item** » a été définie dans le fichier « **Question6.java** » comme suit :

```
class Item {
    String nom;
    int CodeBarre;
    public Item(String x, int y){nom=x;CodeBarre=y;}
    // + d'autres membres (attributs, méthodes, constructeurs etc.)
}
```

Les informations relatives aux items ne sont pas nécessairement uniques, sauf le code barre. En effet, à chaque item est associé un code barre unique représenté dans la classe « **Item** » par « **CodeBarre** ». Vous pouvez supposer que deux items (instance de « **Item** »), ayant le même code barre, ont forcément le même nom. Par contre, deux items ayant le même nom, peuvent avoir des codes barre différents.

**Pour la suite de cet exercice, chaque sous question dépend des précédentes**

**6.1)** Écrire le code de la méthode « **equals** » associée à la classe « **Item** ».

**6.2)** Quelque part dans la méthode « **main** » se trouvant dans la classe « **Question6** », il y a ce fragment de code :

```
Set MesItems = new TreeSet();
Item A = new Item("a",1);
Item B = new Item("b",2);
MesItems.add(A);
MesItems.add(B);
```

Pour que le précédent fragment de code puisse s'exécuter correctement :

**a)** Énumérer les changements, si nécessaire, que vous devez apporter à la classe « **Item** ».

**b)** Est-ce que la classe « **Item** » doit avoir une méthode additionnelle (autre que « **equals** »)? Si oui donner la signature d'une telle méthode?



c) Si vous avez répondu « **oui** » à la précédente question, écrire le code de cette méthode.

6.3) Plus loin dans la même méthode « **main** » de la classe « **Question6** », il y a ce fragment de code :

```
Set AutresItems = new HashSet();
AutresItems.add(A); // -A- instance créée dans 6.2
AutresItems.add(B); // -B- instance créée dans 6.2
```

Pour que le précédent fragment de code puisse s'exécuter correctement :

a) Est-ce que la classe « **Item** » doit avoir une méthode additionnelle (autre que « **equals** »)? Si oui laquelle?

b) Si vous avez répondu « **oui** » à la précédente question, écrire le code de cette méthode.

**6.4)** Plus loin dans la même méthode « **main** » de la classe « **Question6** », il y a ce fragment de code :

```
Set BisItems = new TreeSet(new ItemCompareur());  
BisItems.add(A);  
BisItems.add(B);
```

Avec les mêmes hypothèses de départ .....

**a)** Énumérer les changements, si nécessaire, que vous devez apporter à la classe « **Item** ».

**b)** Définir la classe « **ItemCompareur** » (écrire le code complet de la classe).

c) Tout en justifiant votre réponse, est-ce que vous avez besoin de modifier la définition de la méthode « `equals` » de la classe « `Item` »? Si oui comment?

d) La classe « `Question6` » contient la méthode « `Affiche` » qui sera utilisée dans la méthode « `main` ». Cette méthode permet d'afficher en sortie les noms des items ainsi que le code barre associé, dans l'ordre établi par la collection utilisée. Un exemple d'appel de « `Affiche` » est donné dans le fragment de code suivant :

```
Set AutresItems = new HashSet();
AutresItems.add(A); // -A- instance créée dans 6.2
AutresItems.add(B); // -B- instance créée dans 6.2
Affiche(AutresItems);
```

Écrire le code de la méthode « `Affiche` » en utilisant les nouvelles propriétés de Java 5 (comme le « `foreach` » etc.), sachant que l'avertissement suivant ne doit pas apparaître suite à la compilation de votre programme :

```
« uses unchecked or unsafe operations. Note: Recompile with -Xlint:unchecked for details ».
```

---

**Bonnes vacances!**