IFT1170 - Session Été, Intra

Mohamed Lokbani

# IFT1170 - INTRA

Nom:_________________________________| Prénom(s):____________________________|

Signature:______________________________| Code perm:____________________________|

Date:14 juin 2001

Durée: 2 heures (de 20h30 à 22h30) Local: Z-110

## Directives:

- Il vous est permis d'utiliser un livre de votre choix.
- Les documents de cours ne sont pas autorisés.
- Ordinateurs personnels prohibés.
- Calculatrice (simple) permise.

- Répondre directement sur le questionnaire.

1.____________________/24

2.____________________/09

3.____________________/17

4.____________________/20

5.____________________/30

Total:__________________/100

## Question 1 (24 points)

**-Q1.1-** Quels sont les avantages et les inconvénients de la programmation orientée objet? Expliquez brièvement.

**-Q1.2-** Définissez ce qu'est l'instanciation d'un objet?

**-Q1.3-** Si un constructeur explicite est défini dans une classe, a-t-on accès au constructeur par défaut de cette classe? (encerclez la réponse correcte)

Oui, nous avons accès au constructeur par défaut.

Non, nous n'avons pas accès au constructeur par défaut.

Pourquoi?

**-Q1.4-** En plus d'être **public**, la méthode **main** doit-elle être aussi **static**? (encerclez la réponse correcte)

Oui, la méthode main doit être aussi **static**.

Non, la méthode main n'a pas à être **static**.

Pourquoi?

**-Q1.5-** Dans l'instruction:

```
Test tst = new Test();
```

À quoi réfère le **`Test`** à gauche du "**=**" ? Et celui à droite du "**=**"?

**-Q1.6-** Combien d'objets et de références à des objets sont crées dans le fragment de code suivant et pourquoi (expliquez votre réponse)?

```
Test premier, second;
premier = new Test();
Test exemple = new Test();
```

(une ou plusieurs réponses)

**a)** un objet a été crée
**b)** deux objets ont été crées
**c)** trois objets ont été crées
**d)** une référence a été créée
**e)** deux références ont été créées
**f)** trois références ont été créées

**-Q1.7-** Si vous avez le programme **`test.java`** et vous devez l'exécuter, laquelle des instructions suivantes permettra de le faire et pourquoi (expliquez aussi pourquoi les autres possibilités sont incorrectes).

**a)** `java Test`
**b)** `javac test`
**c)** `java test.java`
**d)** `java test.class`
**e)** `java test`
**f)** `java test.main()`

**-Q1.8-** Expliquez l'intérêt d'avoir la méthode **`toString`** dans une classe.

## Question 2 (9 points)

Que vont afficher en sortie les expressions suivantes et dans chacun des cas expliquez brièvement pourquoi?

**-Q2.1-** `System.out.println(6 + 7 - 1);`

**-Q2.2-** `System.out.println(-5 +7- -6);`

**-Q2.3-** `System.out.println(13 % 3);`

**-Q2.4-** `System.out.println(13.5 % 3.5);`

**-Q2.5-** `System.out.println(8 + 4 + "9");`

**-Q2.6-** `System.out.println("8" + 4 + 9);`

**-Q2.7-** `System.out.println(8 + 1.0f);`

**-Q2.8-** `System.out.println(55/7);`

**-Q2.9-** `System.out.println('c' + 2);`

## Question 3 (17 points)

**-Q3.1-** Expliquez le sens du paragraphe suivant :

« En java, le passage d'un objet dans une méthode se fait par référence. La méthode appelée ne peut pas modifier la valeur de la référence, associée à l'objet, de la méthode appelante, bien qu'elle puisse manipuler directement l'objet. »

**-Q3.2-** Que va afficher en sortie le programme suivant et pourquoi?

```
public class Test {
    public static void main(String args[]) {

      int i = 0; int j = 0;
      int [] tab = new int[1]; tab[0] = i;
      increment1(i); increment2(tab);
      System.out.println("i=" + i + " j=" + j + " tab[0]=" + tab[0]);
    }
    public static void increment1(int x) {x++;}
    public static void increment2(int[] x) {x[0]++;}
}
```

## Question 4 (20 points)

Le programme suivant contient plusieurs erreurs de syntaxe ; identifiez-les et corrigez-les pour qu'il puisse compiler et exécuter correctement. Les numéros de ligne qui figurent au début de chaque ligne ne font pas partie du code et servent à identifier clairement les erreurs.

```
1   PUBLIC CLASS Erreur {
2
3       PUBLIC void main(string args) {
4
5         double fahrenheit = 55.5;
6
7         double celsius = convertion(fahrenheit);
8
9         */ convertion /*
10
11
12        System.out.println(fahrenheit + 'F = ' + celsius + 'C');
13      }
14
15      double convertion(float fahrn) {
16        RETURN (fahr - 32) * 5 / 9;
17      }
18
19  }
```

Format de réponse : Ligne : xxx ; Erreur : ; Courte explication : ; Correction :

## Question 5 (30 points)

Écrivez un programme **`stats.java`** qui calcule la fréquence d'apparition de chaque mot dans une phrase passée en argument. Cette phrase peut contenir plusieurs mots alphanumériques écrits en majuscules et/ou minuscules. Ce programme devra afficher en sortie, les mots et leur fréquence respective, dans un ordre décroissant par rapport à la fréquence d'apparition. Dans le cas ou deux mots ont la même fréquence, affichez ces mots dans un ordre croissant alphabétique.
Nous considérons que deux mots sont identiques s'ils ne diffèrent que par le fait que certains de leur caractères sont en majuscule et minuscule. Par exemple : HOMME, Homme et homme sont considérés ici comme identiques, i.e. le même mot. Aussi, afin de faciliter l'exercice, la ponctuation est à ignorer ainsi que les caractères accentués.

Exemple, pour la phrase (citation de Albert Camus):

**« Certains hommes parlent pendant leur sommeil Les conferenciers parlent pendant le sommeil des autres »**

Séparateurs de mots : espace ou fin de phrase ou début de phrase. Le résultat doit être :

| **mot** | **fréquence** |
|---|---|
| parlent | 2 |
| pendant | 2 |
| sommeil | 2 |
| autres | 1 |
| certains | 1 |
| conferenciers | 1 |
| des | 1 |
| hommes | 1 |
| le | 1 |
| leur | 1 |
| les | 1 |