

IFT1176 - Session Été, Final

Mohamed Lokbani

IFT1176 - Final

Nom: _____ | Prénom(s): _____ |

Signature: _____ | Code perm: _____ |

Section: _____ |

Date : 24 juillet 2002

Durée: 3 heures (de 18h30 à 21h30) Local: 1140 du Pavillon André-Aisenstadt (P.A-A).

Ce document contient 16 pages

Directives:

- Toute documentation permise.
- Calculatrice **non** permise.

- Répondre directement sur le questionnaire.

1. _____/20 Q1.1, Q1.2, Q1.3, Q1.4 (GUI)
2. _____/20 Q2.1, Q2.2, Q2.3 : a, b, c (Threads)
3. _____/20 Q3.1, Q3.2, Q3.3 (SQL + JDBC)
4. _____/20 Q4.1, Q4.2 (Beans)
5. _____/20 Q5.1, Q5.2 (Servlets)

Total: _____/100

Question 1 (20 points)

Soit le programme suivant, qui compile et s'exécute correctement :

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4  class PositionnementDesComposants {
5      PositionnementDesComposants(int choix) {
6          JFrame laFenêtre = new JFrame("Placement de composants");
7          JPanel leConteneurIntermédiaire = new JPanel();
8          laFenêtre.setContentPane(leConteneurIntermédiaire);
9          JButton un = new JButton("un"), deux = new JButton("deux"), trois = new JButton("trois"),
              quatre = new JButton("quatre"), cinq = new JButton("cinq");
10         laFenêtre.addWindowListener(new WindowAdapter () {
11             public void windowClosing (WindowEvent e) {System.exit(0);});
12         switch (choix) {
13             case 0:
14                 leConteneurIntermédiaire.setLayout(new FlowLayout());
15                 leConteneurIntermédiaire.add(un);
16                 leConteneurIntermédiaire.add(deux);
17                 leConteneurIntermédiaire.add(trois);
18                 leConteneurIntermédiaire.add(quatre);
19                 leConteneurIntermédiaire.add(cinq);
20                 break;
21             case 1:
22                 leConteneurIntermédiaire.setLayout(new BorderLayout());
23                 leConteneurIntermédiaire.add(un,BorderLayout.NORTH);
24                 leConteneurIntermédiaire.add(deux,BorderLayout.WEST);
25                 leConteneurIntermédiaire.add(trois,BorderLayout.CENTER);
26                 leConteneurIntermédiaire.add(quatre,BorderLayout.EAST);
27                 leConteneurIntermédiaire.add(cinq,BorderLayout.SOUTH);
28                 break;
29             case 2:
30                 leConteneurIntermédiaire.
31                     setLayout(new BoxLayout(leConteneurIntermédiaire,BoxLayout.Y_AXIS));
32                 un.setAlignmentX(Component.CENTER_ALIGNMENT);
33                 deux.setAlignmentX(Component.LEFT_ALIGNMENT);
34                 trois.setAlignmentX(Component.CENTER_ALIGNMENT);
35                 quatre.setAlignmentX(Component.RIGHT_ALIGNMENT);
36                 cinq.setAlignmentX(Component.CENTER_ALIGNMENT);
37                 leConteneurIntermédiaire.add(un);
38                 leConteneurIntermédiaire.add(deux);
39                 leConteneurIntermédiaire.add(trois);
40                 leConteneurIntermédiaire.add(quatre);
41                 leConteneurIntermédiaire.add(cinq);
42                 break;
43             }
44         laFenêtre.pack();
45         laFenêtre.setVisible(true);
46     }
47     public static void main (String[] arguments){
48         new PositionnementDesComposants(Integer.parseInt(arguments[0]));
49     }
49 }
```

Tout en justifiant votre réponse, que vont afficher (dessiner les figures correspondantes) l'exécution des commandes suivantes (à partir du précédent programme).

-Q1.1- « java PositionnementDesComposants 0 »

Réponse

-Q1.2- « java PositionnementDesComposants 1 »

Réponse

-Q1.3- « java PositionnementDesComposants 2 »

Réponse

-Q1.4- Compléter le programme précédent, afin d'obtenir la figure suivante, lorsque cette commande est exécutée : « java PositionnementDesComposants 3 »



Vous devez utiliser pour cela un gestionnaire de mise en forme permettant de disposer les différents composants suivants une grille régulière, chaque composant occupant une cellule.

Réponse

Insérer le code qui suit entre les lignes : _____ et _____.

Question 2 (20 points)

Soit la classe Compteur :

```
public class Compteur {  
    private int compte = 0;  
    public int incr() {  
        int n = compte;  
        compte = n + 1;  
        return n;  
    }  
}
```

La classe Compteur est utile pour compter par exemple le nombre de fois où un bouton a été appuyé ou bien le nombre de fois qu'une page Web a été visitée etc. Nous allons examiner dans cet exercice le comportement de la classe Compteur dans un environnement multithread.

Supposez que vous avez un programme Java dans lequel deux threads ont été déclarés, et ces deux threads vont faire appel à la même instruction (affectant donc le même objet Compteur)

```
Compteur counter = new Compteur() ;  
int cnt = counter.incr();
```

-Q2.1- Expliquer le comportement des threads pour le scénario suivant :

Thread 1	Thread 2	compte
cnt = counter.incr();	---	0
n = compte; // 0	---	0
compte = n + 1; // 1	---	1
return n; // 0	---	1
---	cnt = counter.incr();	1
---	n = compte; // 1	1
---	Compte = n + 1; // 2	2
---	return n; // 1	2

Réponse

-Q2.2- Expliquer le comportement des threads pour le scénario suivant :

Thread 1	Thread 2	Compte
cnt = counter.incr();	---	0
N = compte; // 0	---	0
---	cnt = counter.incr();	0
---	n = compte; // 0	0
---	Compte = n + 1; // 1	1
---	return n; // 0	1
Compte = n + 1; // 1	---	1
return n; // 0	---	1

Réponse

-Q2.3- À partir des conclusions que vous venez de tirer des questions Q2.1- et Q2.2- :

Q2.3-a Pensez-vous qu'il faudra apporter des modifications à la classe Compteur ?

OUI ☐ NON ☐

Pourquoi? (Courte explication)

Réponse

Q2.3-b Si oui lesquelles (une ou plusieurs) ?

Réponse

Q2.3-c Décrire sous la forme d'un tableau (comme dans les questions Q2.1- et Q2.2-) le comportement des deux threads en tenant compte de vos modifications si vous avez répondu oui à la question Q2.3-a.

Réponse

Thread 1	Thread 2	Compte

Explications si nécessaires

Question 3 (20 points)

Le tableau suivant récapitule la consommation en café des programmeurs d'un projet. Un rapport hebdomadaire a été effectué par le chef de projet qui inclut le nombre total de tasses de café consommées et le nombre de tasses de café consommées par programmeur et par jour.

Programmeur	Jour	# Tasses
Gilbert	Lun	1
Wally	Lun	2
Edgar	Mar	8
Wally	Mar	2
Eugene	Mar	3
Josephine	Mer	2
Eugene	Jeu	3
Gilbert	Jeu	1
Clarence	Ven	9
Edgar	Ven	3
Josephine	Ven	4

Nom de la table : *CafeCouleursData*

La requête SQL utilisée pour créer la table CafeCouleursData est comme suit :

```
create table CafeCouleursData (  
    programmeur varchar (32),  
    jour char (3),  
    tasses integer  
)
```

Questions SQL : Exprimer en SQL les requêtes suivantes :

Q3.1- Affiche le nom de la personne qui a consommé le plus de tasses de café en une journée ainsi que sa consommation ce jour là.

Réponse

Q3.2- Renvoie le nombre total de tasses consommées en une semaine (par l'ensemble des programmeurs).

Réponse

Q3.3- Ecrire en Java la méthode suivante :

```
public static int nbreTotalTassPgm(Statement stmt,String programmeur) throws ...;
```

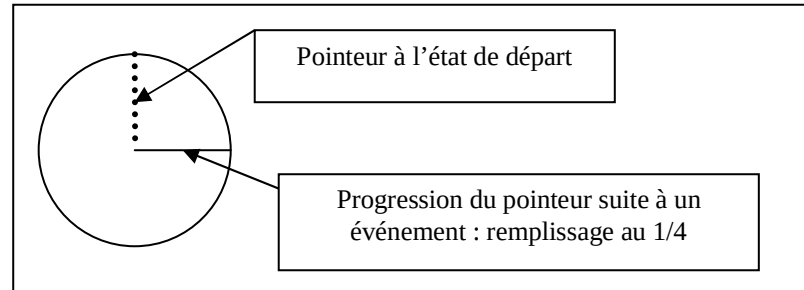
Où :

```
/*  
    Renvoie le nombre total de tasses consommées par un programmeur donné et  
    affiche le détail journalier des consommations et le nombre total de  
    tasses consommées.  
    @param stmt l'objet Statement qui permet de faire la requête.  
    @param programmeur le nom du programmeur.  
    @return le nombre total de tasses consommées.  
*/
```

Réponse

Question 4 (20 points)

Soit une jauge qui va afficher une progression donnée de son pointeur, suite à un événement donné. La figure suivante donne une représentation graphique de cette jauge.



Le code de la Jauge est comme suit:

```
1 package majauge;
2 import java.awt.Graphics;
3 import java.awt.*;
4
5 public class Jauge extends Component {
6     int nombre=0,pas=0,angle=270;
7     public Jauge(int angle, int pas, int nombre) {
8         super();
9         this.angle = angle;
10        this.pas = pas;
11        this.nombre = nombre;
12    }
13    public void progression(){
14        if(nombre <= pas && pas!=0){
15            angle = 270 + (360/pas) * nombre;
16            repaint();
17            nombre++;
18        }
19    }
20    public void modifiePas(int pas){
21        this.pas = pas;
22        nombre = 0;
23        progression();
24    }
25    public int affichePas(){
26        return pas;
27    }
28    public Dimension afficheTailleMax(){
29        return new Dimension(100, 100);
30    }
31    public Dimension afficheTailleMin(){
32        return new Dimension(100, 100);
33    }
34    public void paint(Graphics g){
35        g.drawOval(0, 0, 100, 100);
36        g.drawLine(50, 50, (int) (50 + 50 *
37            Math.cos(angle * Math.PI/180)),
38            (int) (50 + 50 * Math.sin(angle * Math.PI/180)));
39    }
40 }
```

Q4.1- Apporter les modifications nécessaires au programme précédent pour que la Jauge puisse être considérée comme un bean. Par ailleurs :

- Vous devez permettre l'accès aux attributs : « pas » (mode lecture et écriture), « TailleMax » (lecture uniquement) et « TailleMin » (lecture uniquement).
- Il vous est permis de changer les noms des méthodes et de retirer des lignes du code précédent ; mais il ne vous est pas permis d'en ajouter.

Réponse

Une modification à la fois, sous la forme: Ligne No : __ ; Modification apportée : __ ; Explication : __

Q4.2- On voudrait agir sur la jauge par l'intermédiaire d'un clic sur la souris. Comme il ne vous a pas été permis d'ajouter des lignes de code dans le programme « Jauge.java », vous devez écrire cette méthode dans une classe à part, qui se trouve dans le fichier « ClicJauge.java ».

Cette classe doit avoir la forme d'un bean, portera donc le nom de « ClicJauge », aura au moins une référence « cible » du type « Jauge ». Vous devez compléter cette classe avec la méthode permettant d'agir sur la cible suite à un clic de souris qui va provoquer ainsi un appel à la méthode progression pour modifier la position du pointeur. Vous devez permettre l'accès à l'attribut « cible » en mode écriture uniquement.

Question 5 (20 points)

Q5.1- Dessiner la page html associée au code html suivant :

```
<HTML>
<head>
<title>Exercice #4</title>
</head>

<body>

<FORM ACTION="../../servlet/Commandes" METHOD="POST">

    <div align="center">
        <h1>Bienvenu au centre d'achat "<u>Il y a tout!</u>"</h1>
    </div>
    <h2>Que voulez-vous acheter aujourd'hui ?</h2>
    <P>
    <table border = 1>
    <tr>
        <td><input type=checkbox name=produit1> Veste</td>
        <td>Quantité <select name=quantite1>
            <option value=1> 1 <option value=2> 2 <option value=3> 3
        </select> </td>
    </tr>
    <tr>
        <td> <input type=checkbox name=produit2> T-shirt </td>
        <td> Quantité <select name=quantite2>
            <option value=1> 1 <option value=2> 2 <option value=3> 3
        </select> </td>
    </tr>
    </table>
    <P>
    <br><h2>Méthode de livraison</h2>
    <input type=radio name=livraison value="Xpresspost" checked>
    ExpressPosteCanada
    <BR> <input type=radio name=livraison value="fedex"> Federal
    Express
    <P>
    Nom: <input type=text name="nom">
    <P>
    Adresse Email: <input type=text name="email">
    <P>
    <input type = submit value="Commander maintenant!">
</form>
</body>
</HTML>
```

Réponse

Q5.2- Ecrire la servlet qui suite à la soumission d'une requête donnée via la page html décrite dans la question Q4.1-, retourne une réponse sous la forme suivante :

Exercice #4! - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost:8080/examples/servlet/Commandes> Go

Nom: test

Email: test@place.com

Votre commande!

Produit	Quantite	Prix	Total
Veste	3	100.0	300.0
T-shirt	2	20.0	40.0
ExpressPosteCanada			10.0
			350.0

Vous devez tenir compte des hypothèses suivantes :

Produit	Prix (\$)
Veste	100
T-shirt	20
ExpressPosteCanada	10
Federal Express	20

Réponse

Bonnes Vacances !