

TP IHM 4^{ème} Année I.S.I : JSF – Part II

I. Création d'un projet Web Dynamique :

1) Le nom du projet web dynamique est 'Tp7ExemplesJSFGroupeXX'.

II. Exemples pratiques :

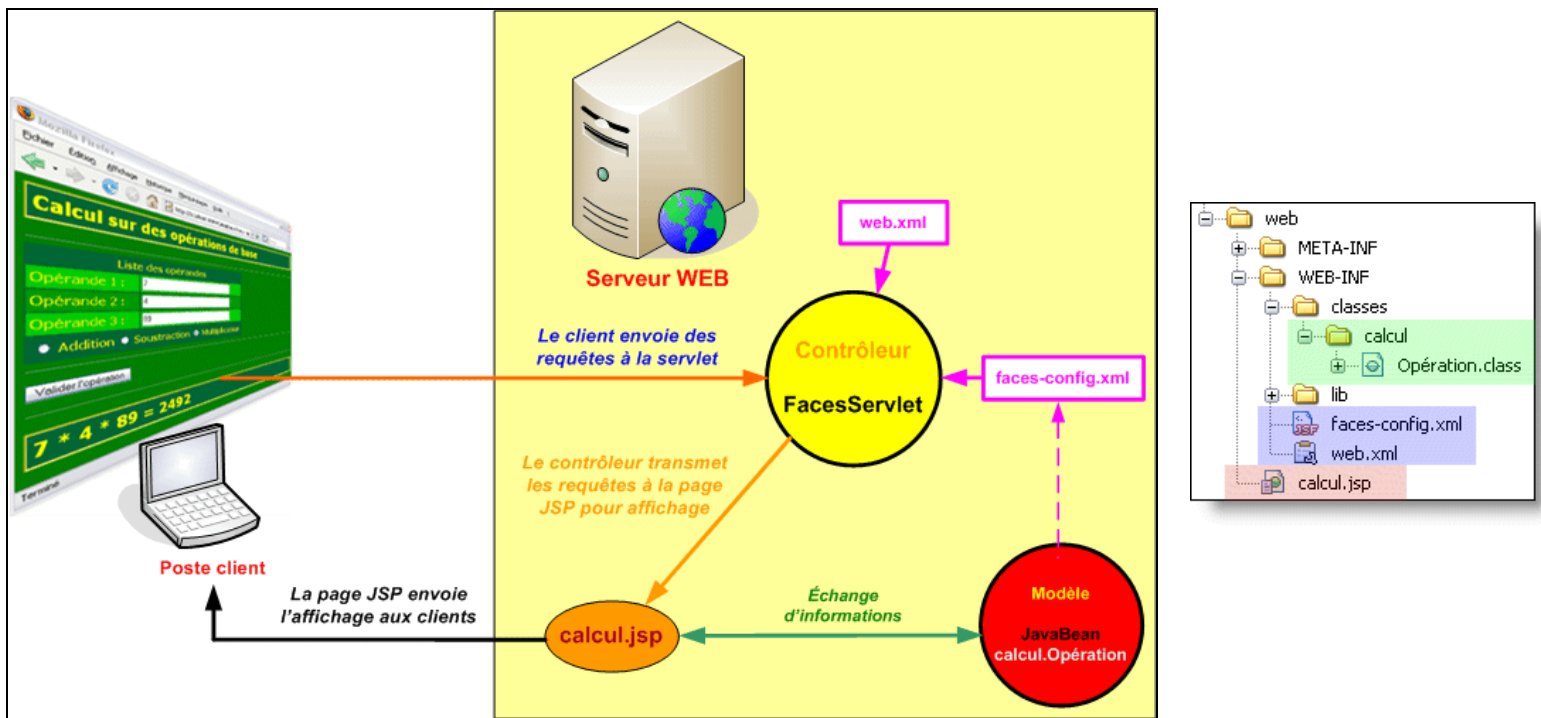
1) Gestionnaire de disposition, les radio-boutons, les styles :

a. Description :

L'objectif de cette partie est de reprendre l'application Web sur le calcul des opérations de base et faire une présentation plus agréable en mettant en œuvre un panneau qui représente toutes les informations sous la forme d'une grille pour voir comment mettre en œuvre les radio-boutons et donc comment mettre en place une liste prédéfinie de valeurs.

b. Modèle MVC :

- ✓ Tout l'affichage s'exécute sur la même page **calcul.jsp** ;
- ✓ Le bean **calcul.Operation** devra gérer l'affichage du résultat et récupérer le type d'opération afin de pouvoir calculer le résultat correctement.



c. La page d'accueil

http://localhost/Tp7ExemplesJSFGroupe00/calcul.jsf

Calcul sur des opérations de base

Liste des opérands

Opérande 1 : 5

Opérande 2 : 6

Opérande 3 : 7

☒ Addition ☐ Soustraction ☐ Multiplication

Valider l'opération

5 + 6 + 7 = 18

d. Méthode 1 :

```
1 package calcul;
2
3 public class Operation {
4
5     private int premier;
6     private int deuxieme;
7     private int troisieme;
8     private char type = '+';
9     private int resultat;
10
11     public void setType(char type) {}
12
13
14     public char getType() {}
15
16
17
18     public int getPremier() {}
19
20
21
22     public void setPremier(int premier) {}
23
24
25
26     public int getDeuxieme() {}
27
28
29
30     public void setDeuxieme(int deuxieme) {}
31
32
33
34     public int getTroisieme() {}
35
36
37
38     public void setTroisieme(int troisieme) {}
39
40
41
42     private void calcul() {}
43
44
45
46     public String getResultat() {
47         calcul();
48         if (premier==0) return "Aucun résultat";
49         return premier+" "+type+" "+deuxieme+" "+type+" "+troisieme+" = "+resultat;
50     }
51
52
53
54 }
55
56
57
58
```

```

calcul.jsp x
5 <!-- Entet JSF -->
6 <style type="text/css">
7     .un {background-color: #00CC00; }
8     .deux {background-color: #009900; }
9     .general {background-color: #006633; font-weight: bold; }
10    h2 { background-color: #006633; padding: 5px; border : groove; }
11    body {font-family: verdana, Arial; color: yellow; background-color: green; }
12 </style>
13
14 <html>
15     <body>
16         <f:view>
17             <h2><h:outputText value="Calcul sur des opérations de base" /></h2>
18             <hr />
19             <h:form>
20                 <h:panelGrid columns="2" cellpadding="3" styleClass="general" rowClasses="un, deux">
21                     <f:facet name="header"><h:outputText value="Liste des opérantes" /></f:facet>
22                     <h:outputText value="Opérante 1 : " />
23                     <h:inputText value="#{operation.premier}" size="30" maxlength="10" />
24                     <h:outputText value="Opérante 2 : " />
25                     <h:inputText value="#{operation.deuxieme}" size="30" maxlength="10" />
26                     <h:outputText value="Opérante 3 : " />
27                     <h:inputText value="#{operation.troisieme}" size="30" maxlength="10" />
28                     <f:facet name="Footer">
29                         <h:selectOneRadio value="#{operation.type}" style="font-weight: bold" >
30                             <f:selectItem itemLabel="Addition" itemValue="+" />
31                             <f:selectItem itemLabel="Soustraction" itemValue="-" />
32                             <f:selectItem itemLabel="Multiplication" itemValue="*" />
33                         </h:selectOneRadio>
34                     </f:facet>
35                 </h:panelGrid>
36                 <hr />
37                 <p><h:commandButton action="valider" value="Valider l'opération" /></p>
38             </h:form>
39             <hr />
40             <h2><h:outputText value="#{operation.resultat}" /></h2>
41         </f:view>
42     </body>
43 </html>
44

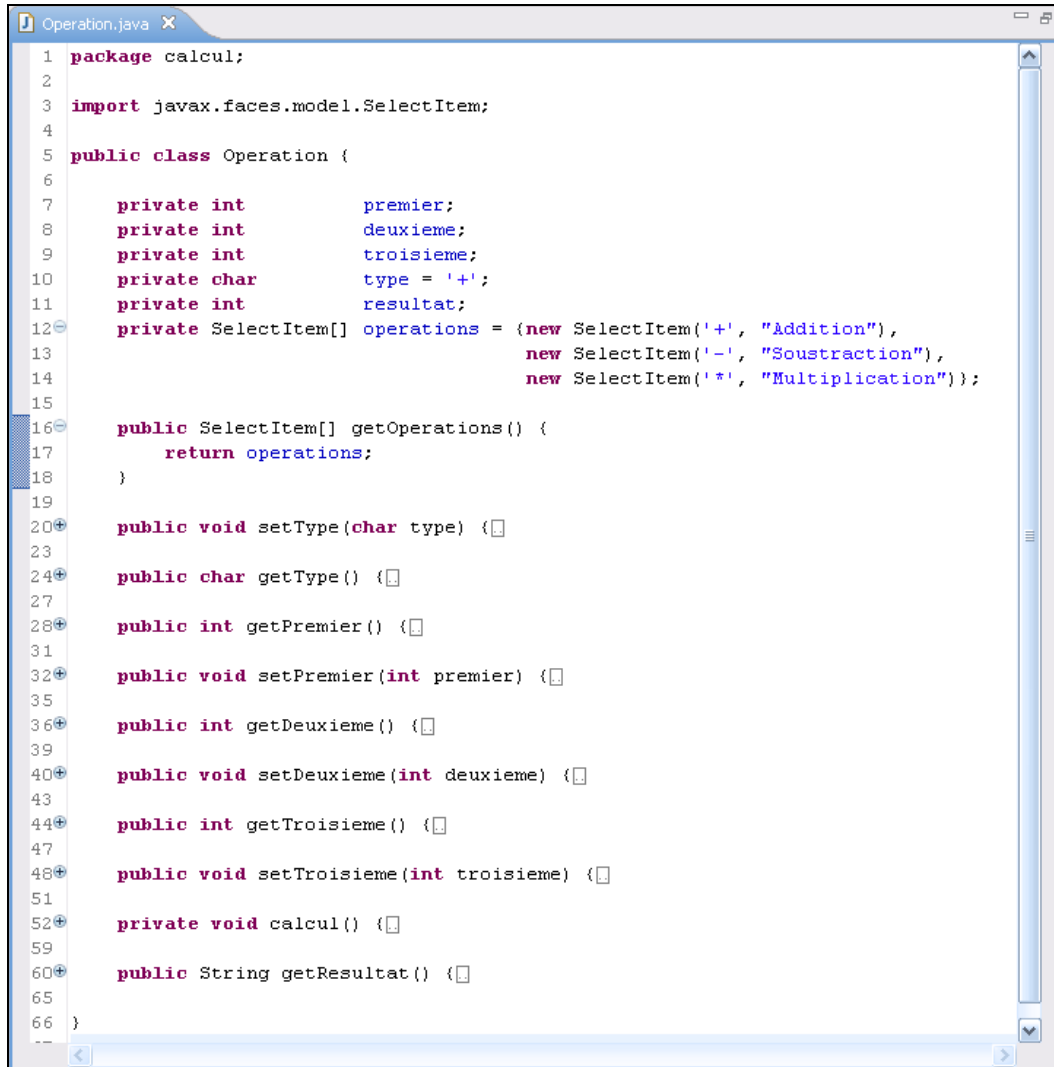
```

e. Méthode 2 :

```

calcul.jsp x
5 <!-- Entet JSF -->
6 <style type="text/css">
13
14 <html>
15     <body>
16         <f:view>
17             <h2><h:outputText value="Calcul sur des opérations de base" /></h2>
18             <hr />
19             <h:form>
20                 <h:panelGrid columns="2" cellpadding="3" styleClass="general" rowClasses="un, deux" >
21                     <f:facet name="header"><h:outputText value="Liste des opérantes" /></f:facet>
22                     <h:outputText value="Opérante 1 : " />
23                     <h:inputText value="#{operation.premier}" size="30" maxlength="10" />
24                     <h:outputText value="Opérante 2 : " />
25                     <h:inputText value="#{operation.deuxieme}" size="30" maxlength="10" />
26                     <h:outputText value="Opérante 3 : " />
27                     <h:inputText value="#{operation.troisieme}" size="30" maxlength="10" />
28                     <f:facet name="footer">
29                         <h:selectOneRadio value="#{operation.type}" style="font-weight: bold" >
30                             <f:selectItems value="#{operation.operations}" />
31                         </h:selectOneRadio>
32                     </f:facet>
33                 </h:panelGrid>
34                 <hr />
35                 <p><h:commandButton action="valider" value="Valider l'opération" /></p>
36             </h:form>
37             <hr />
38             <h2><h:outputText value="#{operation.resultat}" /></h2>
39         </f:view>
40     </body>
41 </html>
42

```



```
1 package calcul;
2
3 import javax.faces.model.SelectItem;
4
5 public class Operation {
6
7     private int        premier;
8     private int        deuxieme;
9     private int        troisieme;
10    private char        type = '+';
11    private int        resultat;
12    private SelectItem[] operations = {new SelectItem('+', "Addition"),
13                                       new SelectItem('-', "Soustraction"),
14                                       new SelectItem('*', "Multiplication")};
15
16    public SelectItem[] getOperations() {
17        return operations;
18    }
19
20    public void setType(char type) {}
21
22    public char getType() {}
23
24    public int getPremier() {}
25
26    public void setPremier(int premier) {}
27
28    public int getDeuxieme() {}
29
30    public void setDeuxieme(int deuxieme) {}
31
32    public int getTroisieme() {}
33
34    public void setTroisieme(int troisieme) {}
35
36    private void calcul() {}
37
38    public String getResultat() {}
39
40 }
```

f. Configurer le bean et les règles de navigation pour les deux méthodes.

III. Exercices d'application : Technique avancée.

1) Application : *Tool Shop*

L'objectif de cette partie est de transformer la page d'accueil de l'application statique *Tool Shop* à une application client serveur dynamique à base de la technologie JSF.

IV. Déploiement :

- 1) Le déploiement sera effectué grâce à l'application 'Tomcat Manager'.
- 2) La génération du fichier 'war' sera effectuée par Eclipse.
- 3) Chaque groupe va déployer et tester le projet réalisé par un autre groupe.