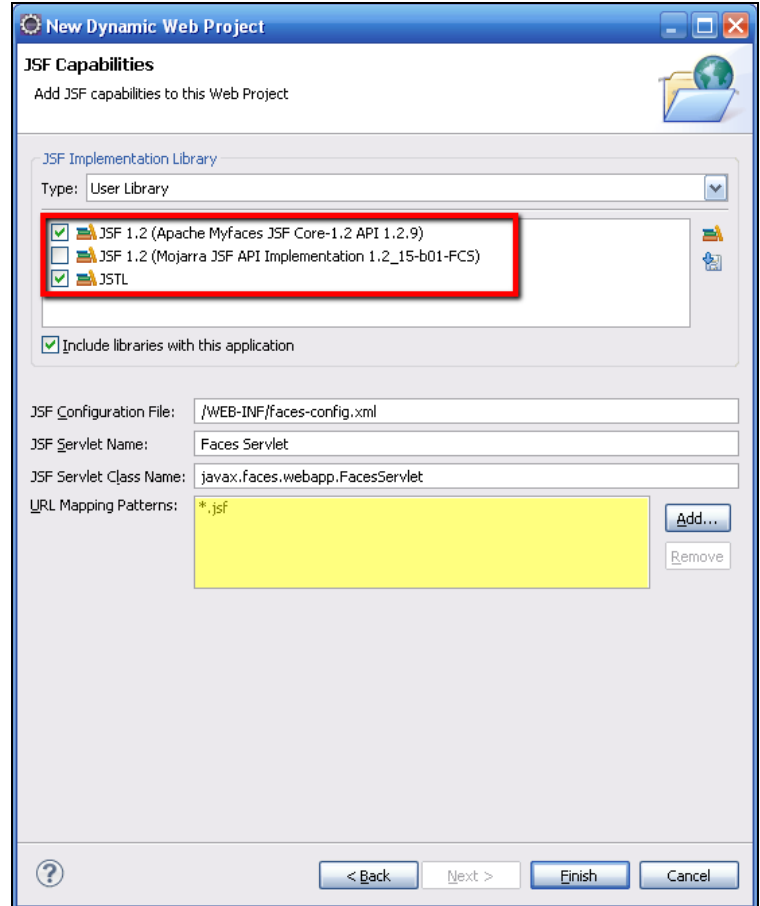
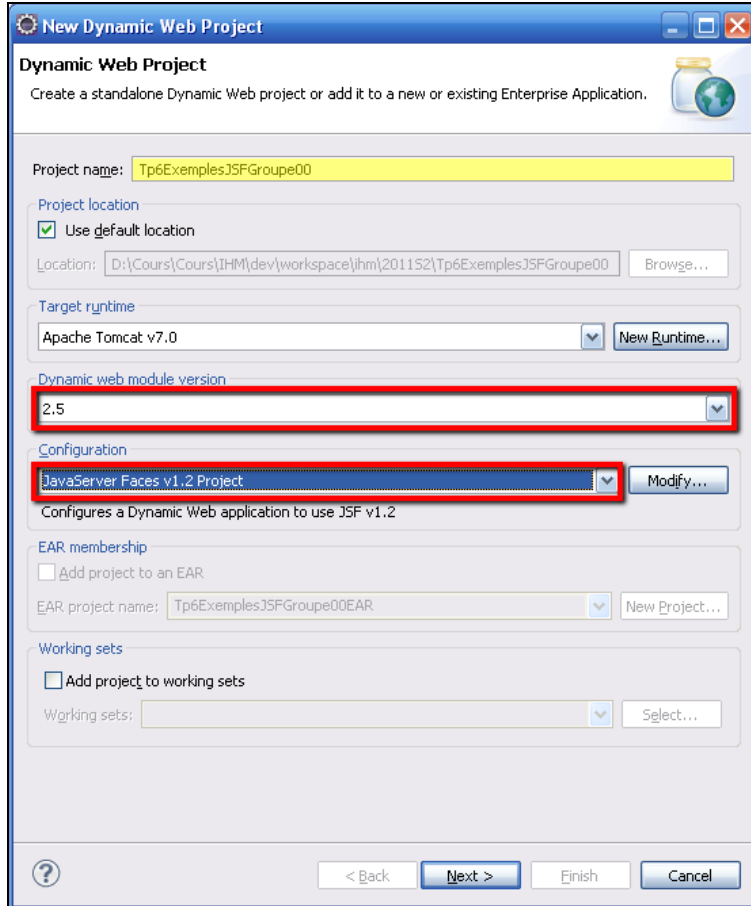


TP IHM 4^{ème} Année I.S.I : JSF

I. Création d'un projet Web Dynamique :

1) Le nom du projet web dynamique est 'Tp6ExemplesJSFGroupeXX'.

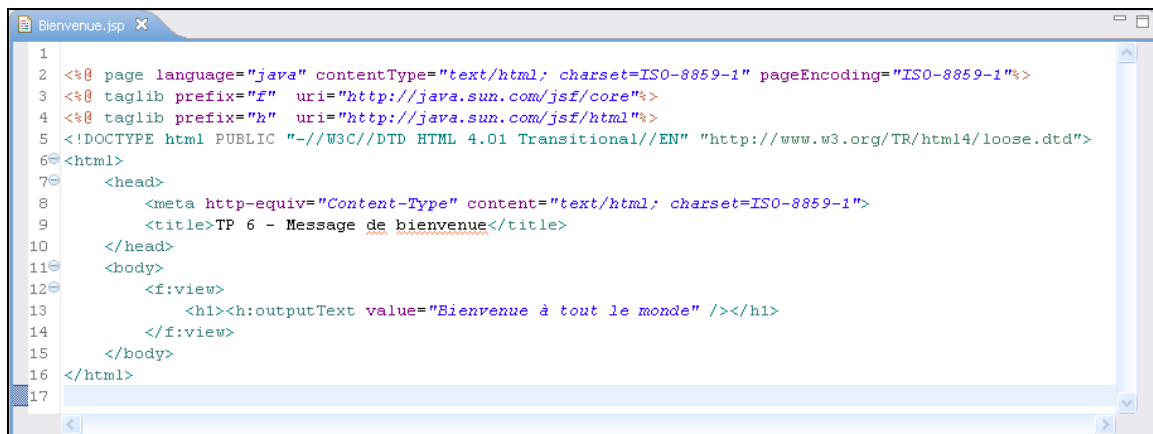
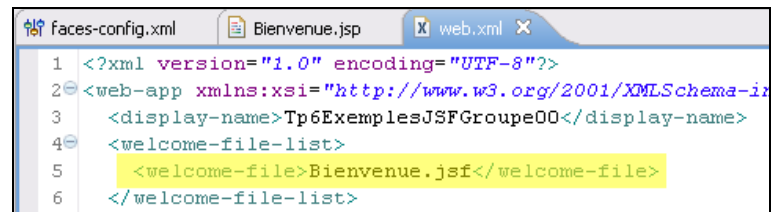
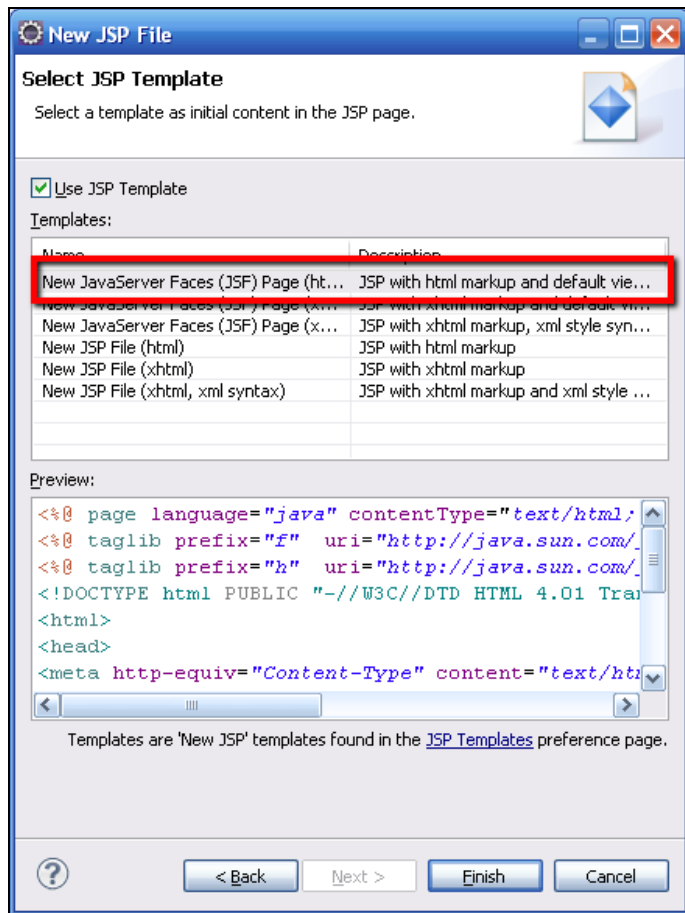


II. Exemples pratiques :

1) Message de bienvenue :

- Créer la page 'Bienvenue.jsp'.
- La page 'Bienvenue.jsp' doit être définie comme page d'accueil du projet.
- Le code de la page doit être le suivant.





2) Nombre aléatoire :

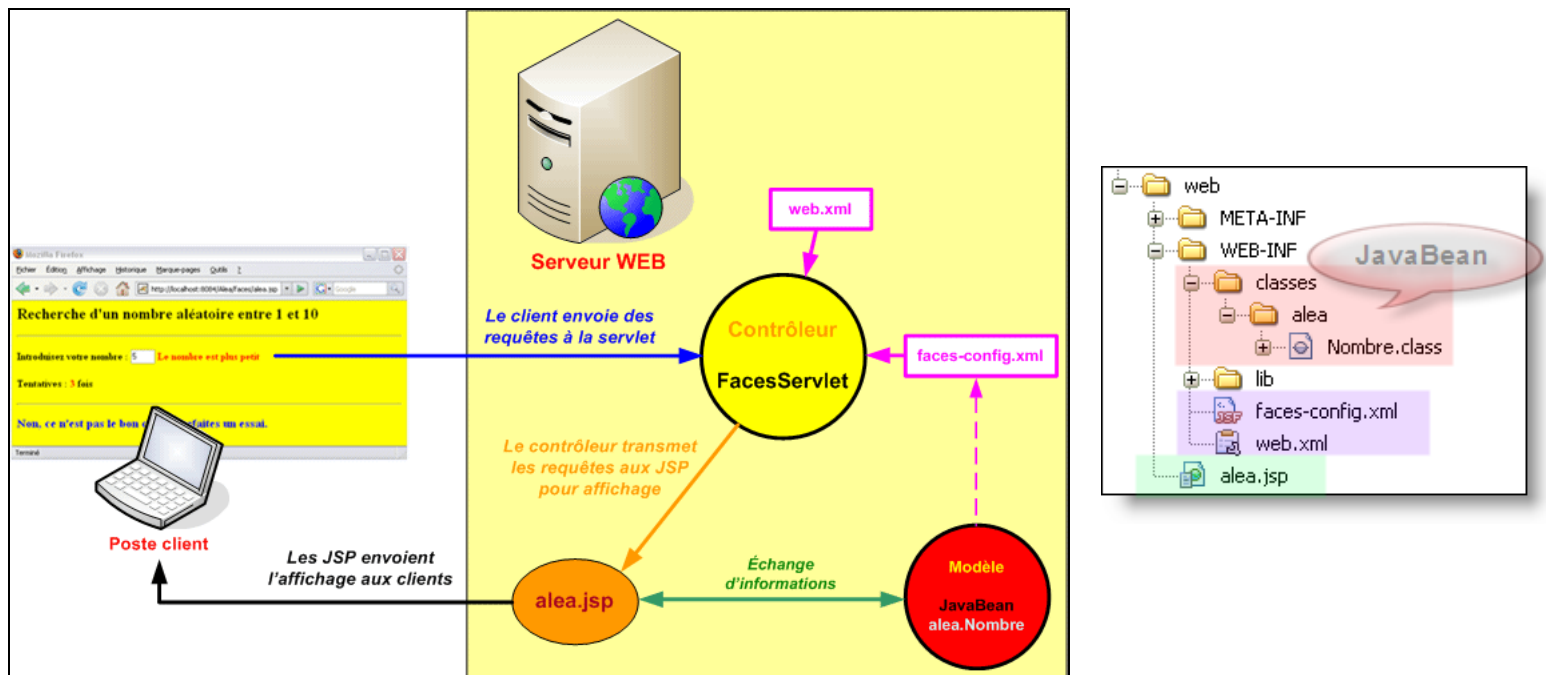
a. Description :

L'application devra permettre d'exécuter un tout petit jeu qui tire automatiquement un nombre aléatoire à la connexion de la session et dont nous devons retrouver la valeur en proposant des saisies successives.

b. Modèle MVC :

✓ On a besoin de fabriquer un nombre aléatoire de 1 à 10.

- ✓ Il faut récupérer la saisie proposée par l'utilisateur et la contrôler suivant le tirage réalisé ;
 - ✓ Il faut comptabiliser le nombre de coup effectués avant de trouver le bon chiffre ;
 - ✓ Il faut prévoir un certain nombre de messages adaptés à la situation afin d'orienter l'utilisateur pour son prochain choix.
- ✓ Le traitement est réalisé séparément, à l'aide du JavaBean *alea.Nombre* ;
 - ✓ La page d'accueil *alea.jsp* ;
 - ✓ Il n'y aura donc pas de navigation à mettre en œuvre ;
 - ✓ Cette page sera en relation directe avec le JavaBean *alea.Nombre* ;
 - ✓ Les échanges pourront s'effectuer simplement dans les deux sens.

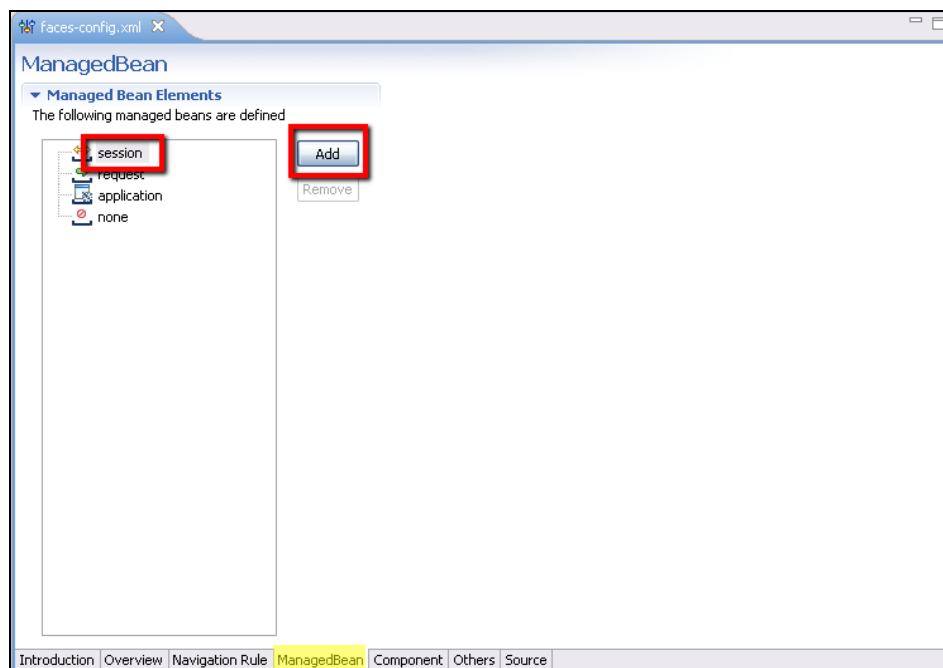


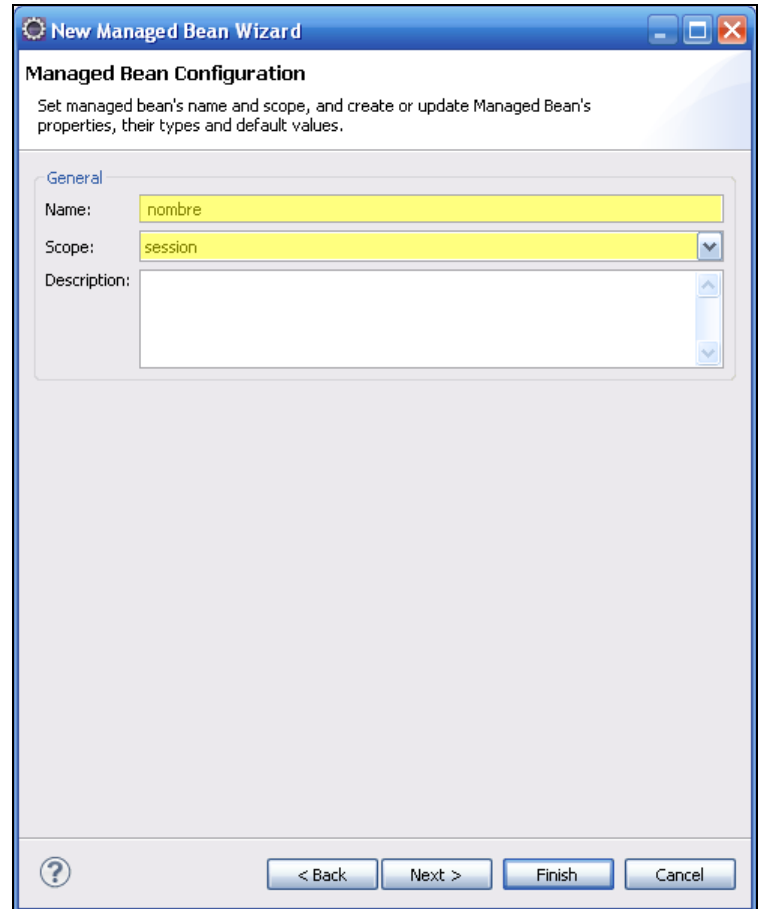
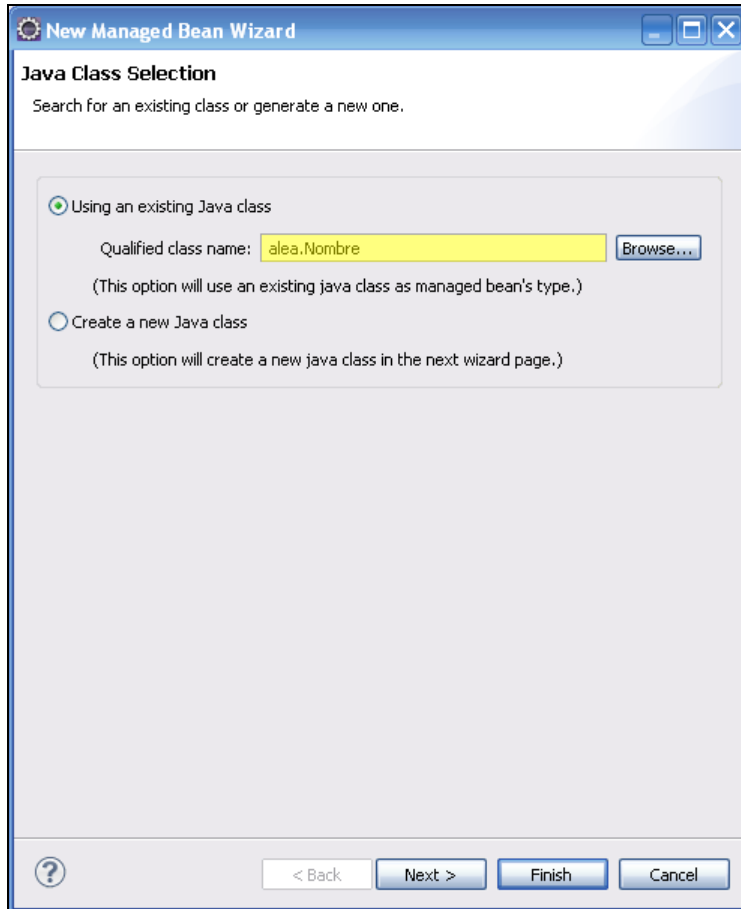
c. La page d'accueil



d. Créer et configurer le bean '*Nombre.java*'.

```
Nombre.java
8      private boolean test;
9
10     public Nombre() {
11         nombreARechercher = Math.round(Math.random()*10) + 1;
12     }
13
14     public void setValeur(long valeur) {
15         this.valeur = valeur;
16         tentative++;
17         test = nombreARechercher == valeur;
18     }
19
20     public long getValeur() {
21         return valeur;
22     }
23
24     public int getTentative() {
25         return tentative;
26     }
27
28     public String getResultat() {
29         String resultat;
30         if(tentative == 0)
31             resultat = "Tentez votre chance";
32         if(test)
33             resultat = "Bravo, vous avez trouvé !";
34         else
35             resultat = "Non, ce n'est pas le bon chiffre, refaites un essai.";
36         return resultat;
37     }
38
39     public String getProgression() {
40         String progression;
41         if((tentative == 0) || test)
42             progression = "";
43         else
44             progression = "Le nombre est plus " + (nombreARechercher > valeur ? "grand" : "petit");
45         return progression;
46     }
47
48 }
49
```





e. Créer la page d'accueil 'alea.jsp'.



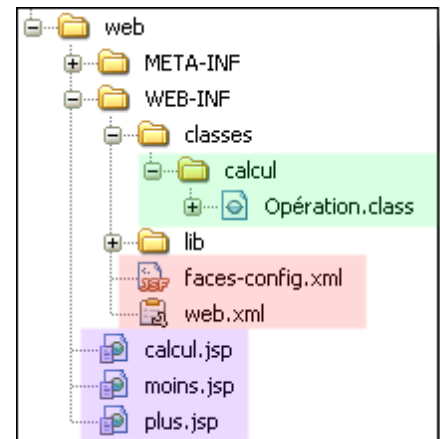
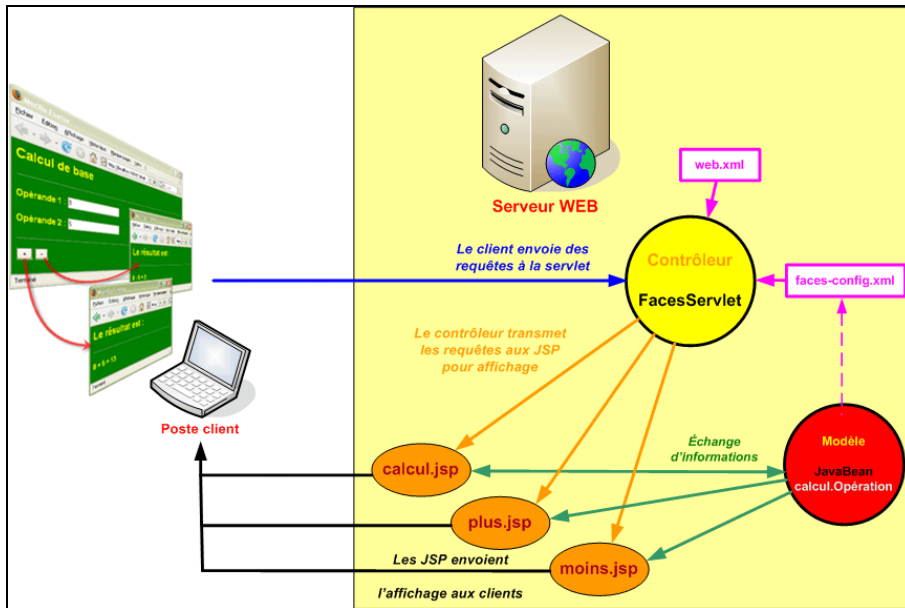
3) Calcul de base :

a. Description :

L'application réalisera des opérations de base, comme l'addition et la soustraction, sur deux opérands. Une fois que les opérands sont définis, il suffit de choisir l'opération à traiter à l'aide du bouton concerné, et une nouvelle page Web apparaît alors, avec le traitement correspondant à l'action souhaitée.

b. Modèle MVC :

- ✓ Le modèle doit prendre en compte la navigation entre plusieurs pages ;
- ✓ Le traitement et le stockage des informations sont toujours réalisés par un bean ***calcul.Operation***.

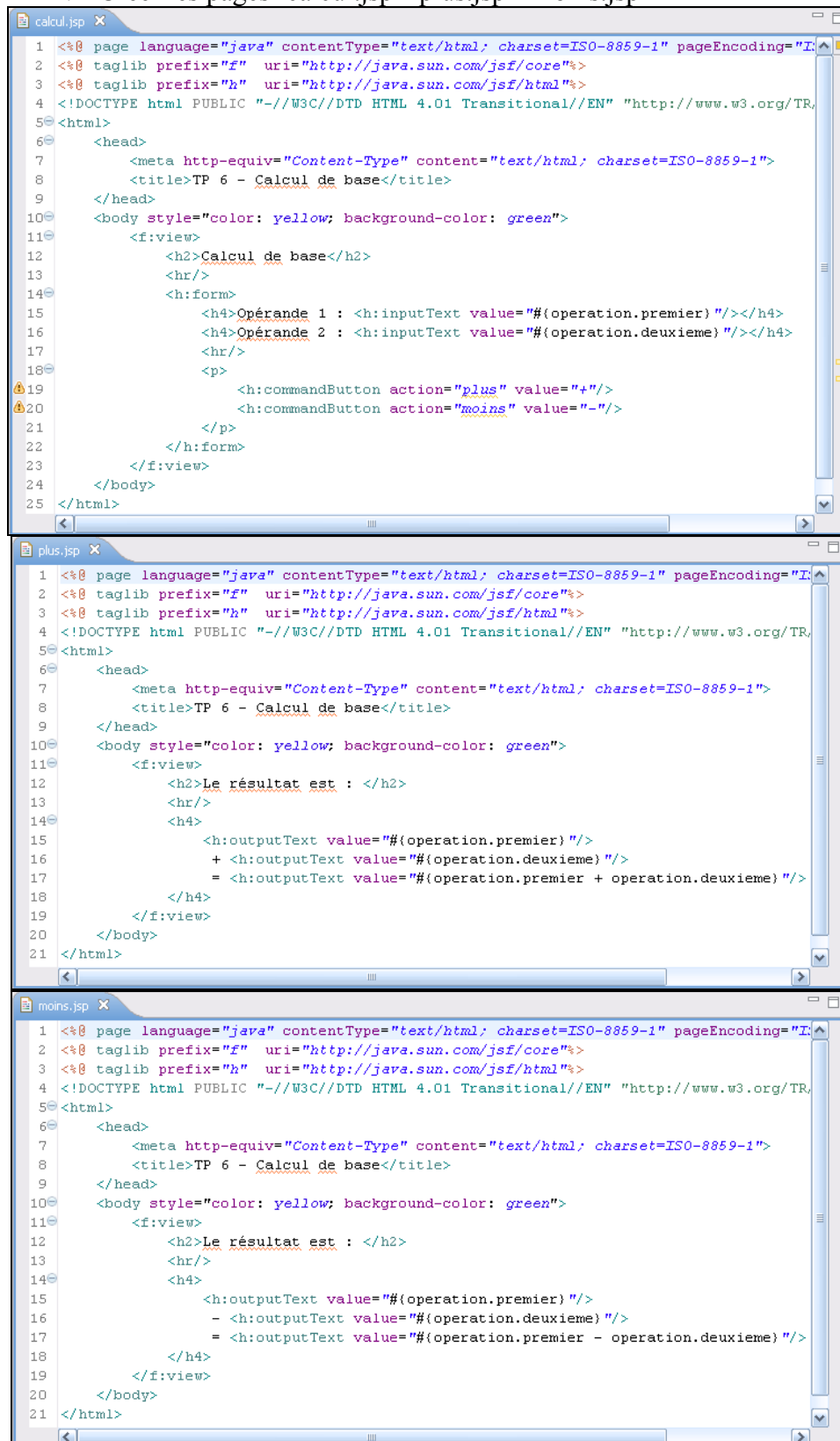


c. La page d'accueil : ***ressource II3.swf***

d. Créer le bean 'Operation.java'

```
1 package calcul;
2
3 public class Operation {
4
5     private int premier;
6     private int deuxieme;
7
8     public int getPremier() {
9         return premier;
10    }
11
12    public void setPremier(int premier) {
13        this.premier = premier;
14    }
15
16    public int getDeuxieme() {
17        return deuxieme;
18    }
19
20    public void setDeuxieme(int deuxieme) {
21        this.deuxieme = deuxieme;
22    }
23
24 }
25
```

- e. Configurer le bean : **ressource II3a.swf**
f. Créer les pages 'calcul.jsp' 'plus.jsp' 'moins.jsp'



```
calcul.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="UTF-8"%>
2 <%@ taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
3 <%@ taglib prefix="h" uri="http://java.sun.com/jsf/html"%>
4 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/strict.dtd">
5 <html>
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
8 <title>TP 6 - Calcul de base</title>
9 </head>
10 <body style="color: yellow; background-color: green">
11 <f:view>
12 <h2>Calcul de base</h2>
13 <hr/>
14 <h:form>
15 <h4>Opérande 1 : <h:inputText value="#{operation.premier}" /></h4>
16 <h4>Opérande 2 : <h:inputText value="#{operation.deuxieme}" /></h4>
17 <hr/>
18 <p>
19 <h:commandButton action="plus" value="+" />
20 <h:commandButton action="moins" value="-" />
21 </p>
22 </h:form>
23 </f:view>
24 </body>
25 </html>

plus.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="UTF-8"%>
2 <%@ taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
3 <%@ taglib prefix="h" uri="http://java.sun.com/jsf/html"%>
4 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/strict.dtd">
5 <html>
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
8 <title>TP 6 - Calcul de base</title>
9 </head>
10 <body style="color: yellow; background-color: green">
11 <f:view>
12 <h2>Le résultat est : </h2>
13 <hr/>
14 <h4>
15 <h:outputText value="#{operation.premier}" />
16 + <h:outputText value="#{operation.deuxieme}" />
17 = <h:outputText value="#{operation.premier + operation.deuxieme}" />
18 </h4>
19 </f:view>
20 </body>
21 </html>

moins.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="UTF-8"%>
2 <%@ taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
3 <%@ taglib prefix="h" uri="http://java.sun.com/jsf/html"%>
4 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/strict.dtd">
5 <html>
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
8 <title>TP 6 - Calcul de base</title>
9 </head>
10 <body style="color: yellow; background-color: green">
11 <f:view>
12 <h2>Le résultat est : </h2>
13 <hr/>
14 <h4>
15 <h:outputText value="#{operation.premier}" />
16 - <h:outputText value="#{operation.deuxieme}" />
17 = <h:outputText value="#{operation.premier - operation.deuxieme}" />
18 </h4>
19 </f:view>
20 </body>
21 </html>
```

g. Configurer les règles de navigation : *ressources II3b.swf*

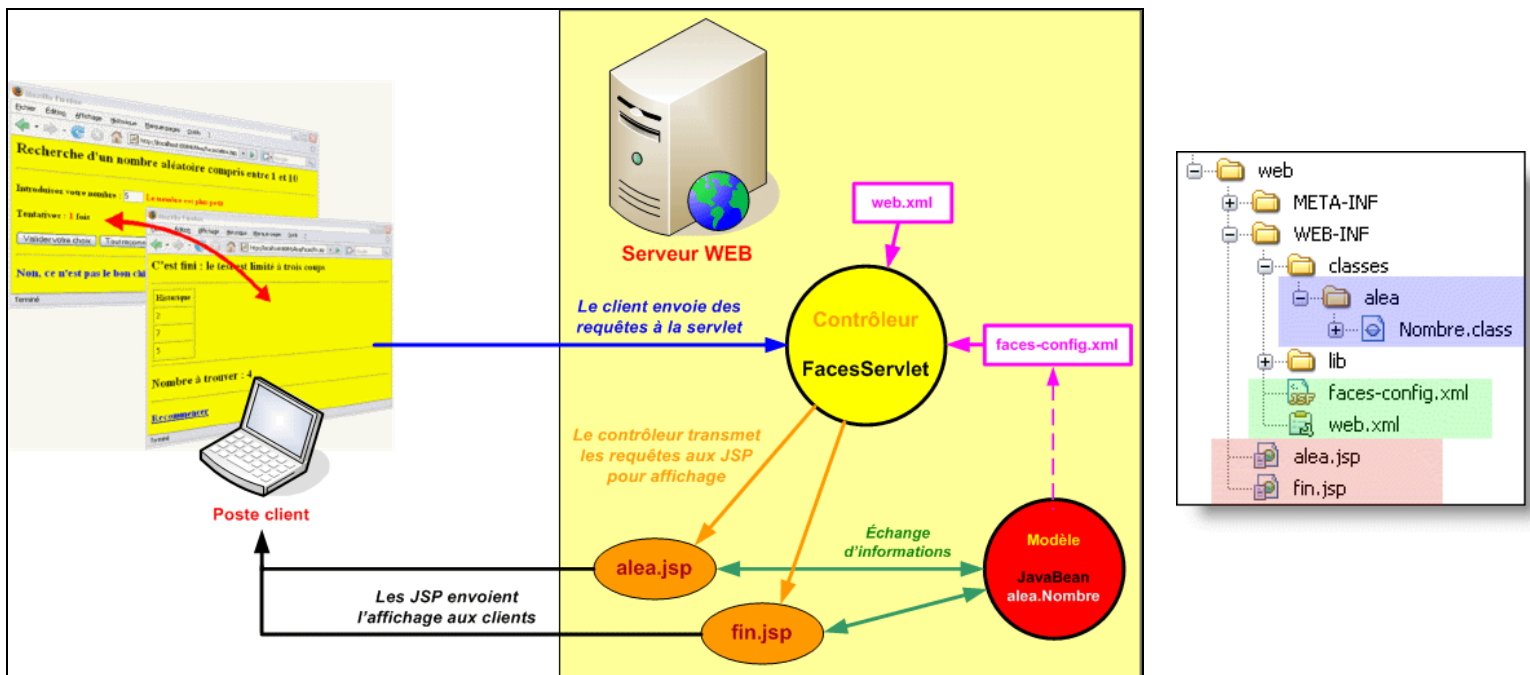
4) Nombre aléatoire 2 :

a. Description :

L'application permet de tirer automatiquement un nombre aléatoire à la connexion de la session, et dont on doit retrouver la valeur en proposant des saisies successives. Le nombre doit être retrouvé en trois coups au maximum. Dans le cas contraire une nouvelle page apparaît donnant la valeur qu'on doit trouver ainsi que l'historique des valeurs saisies.

b. Modèle MVC :

- ✓ Le modèle doit prendre en compte la navigation entre les deux pages ;
- ✓ Le traitement des informations est toujours réalisé par le bean *alea.Nombre*.



c. La page d'accueil :

III. 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.