

TP4 : JEE

☰ Tags	
▼ status	sent

MVC

L'objectif de ce TP est de se familiariser avec le model MVC, le découpage de l'application en 3 couches :

Web, Métier et Donnée, et l'implémentation du model MVC dans la couche Web.

Exercice 1 : Création d'une application du calcul de l'IMC (Calcul_IMC)

Couche Metier :

1. Création de l'interface :

```
IImcMetier.java ×  
1 package metier;  
2  
3 public interface IImcMetier {  
4     public double calculerImc(int poids, double taille);  
5 }
```

2. Implémentation de l'interface :

```
1 package metier;
2
3 public class ImcMetierImpl implements IImcMetier {
4     @Override
5     public double calculerImc(int poids, double taille) {
6         double imc = poids/(taille*taille);
7         // TODO Auto-generated method stub
8         return imc;
9     }
10 }
```

3. Test et validation du traitement métier

```
1 package metier;
2
3 public class TestMetier {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         ImcMetierImpl metier=new ImcMetierImpl();
7         double taille=1.2;
8         int poids=65;
9         double imc=metier.calculerImc(poids, taille);
10        System.out.println(imc);
11    }
12 }
```

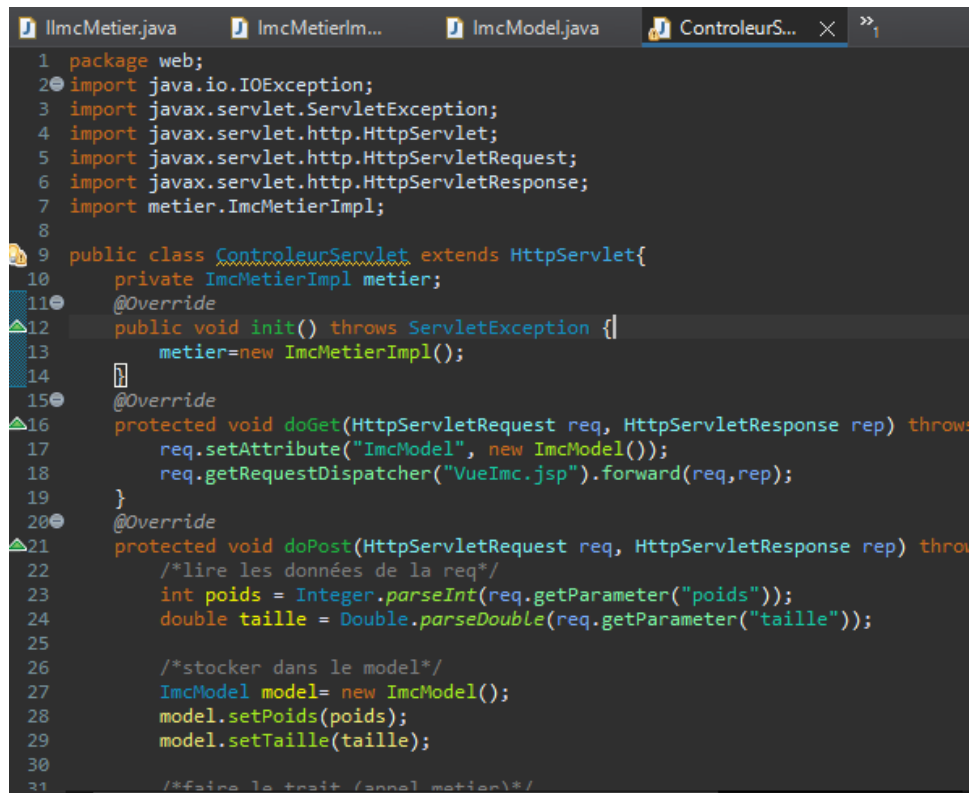
```
Problems Servers Terminal Data Source Explorer Properties Console
<terminated> TestMetier [Java Application] D:\Users\ibtissam\Documents\eclipse-jee-2023-09-R-win32-x86_64
45.13888888888889
```

Couche Web :

1. Création de model:

```
lmcMetier.java  lmcMetierImpl.java  TestMetier.java  lmcModel.java X
1 package web;|
2
3 public class lmcModel {
4     private int poids;
5     private double taille;
6     private double imc;
7     public int getPoids() {
8         return poids;
9     }
10    public void setPoids(int poids) {
11        this.poids = poids;
12    }
13    public double getTaille() {
14        return taille;
15    }
16    public void setTaille(double taille) {
17        this.taille = taille;
18    }
19    public double getImc() {
20        return imc;
21    }
22    public void setImc(double imc) {
23        this.imc = imc;
24    }
25    public lmcModel() {
26        super();
27        // TODO Auto-generated constructor stub
28    }
29    public lmcModel(int poids, double taille, double imc) {
30        super();
31        this.poids = poids;
```

2. Création du contrôleur

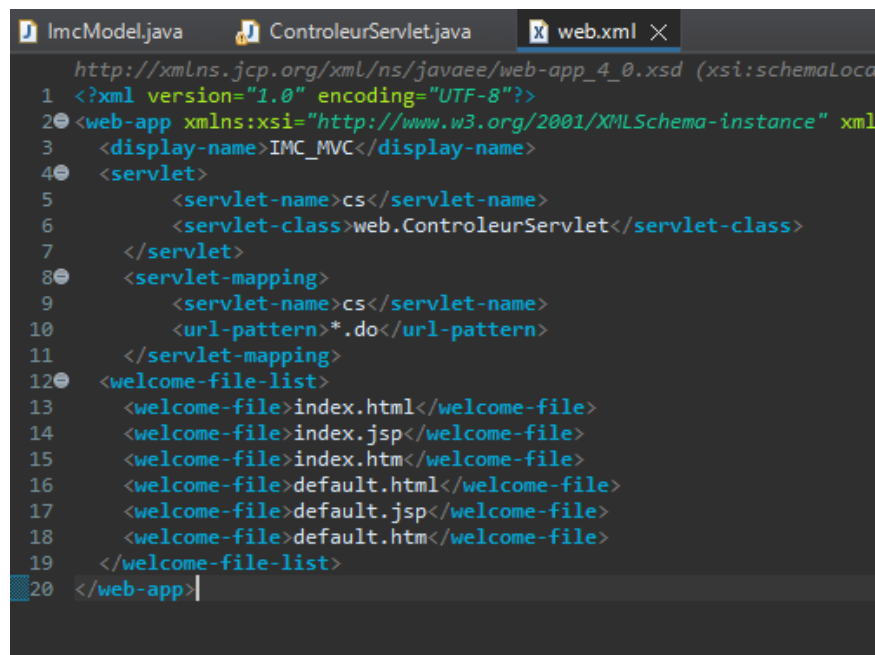


```

1 package web;
2 import java.io.IOException;
3 import javax.servlet.ServletException;
4 import javax.servlet.http.HttpServlet;
5 import javax.servlet.http.HttpServletRequest;
6 import javax.servlet.http.HttpServletResponse;
7 import metier.ImcMetierImpl;
8
9 public class ControleurServlet extends HttpServlet{
10     private ImcMetierImpl metier;
11     @Override
12     public void init() throws ServletException {
13         metier=new ImcMetierImpl();
14     }
15     @Override
16     protected void doGet(HttpServletRequest req, HttpServletResponse rep) throws
17         req.setAttribute("ImcModel", new ImcModel());
18         req.getRequestDispatcher("VueImc.jsp").forward(req,rep);
19     }
20     @Override
21     protected void doPost(HttpServletRequest req, HttpServletResponse rep) thro
22         /*lire les données de la req*/
23         int poids = Integer.parseInt(req.getParameter("poids"));
24         double taille = Double.parseDouble(req.getParameter("taille"));
25
26         /*stocker dans le model*/
27         ImcModel model= new ImcModel();
28         model.setPoids(poids);
29         model.setTaille(taille);
30
31         /*faire le trait (appel metier)*/

```

3. Edition Descripteur de déploiement de la servlet : web.xml



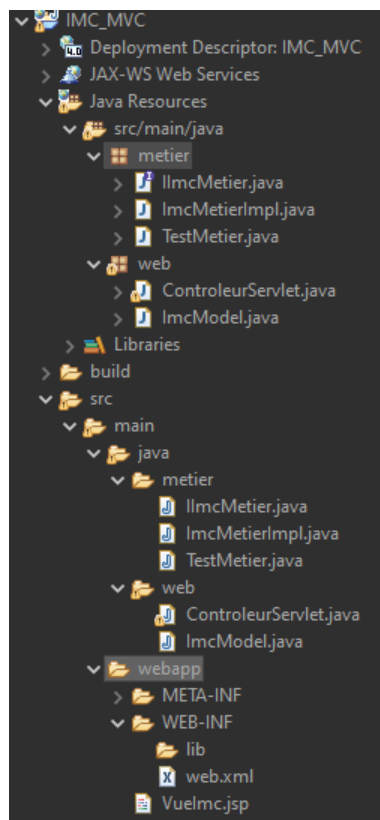
```

http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd (xsi:schemaLoca
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml
3     <display-name>IMC_MVC</display-name>
4     <servlet>
5         <servlet-name>cs</servlet-name>
6         <servlet-class>web.ControleurServlet</servlet-class>
7     </servlet>
8     <servlet-mapping>
9         <servlet-name>cs</servlet-name>
10        <url-pattern>*.do</url-pattern>
11    </servlet-mapping>
12    <welcome-file-list>
13        <welcome-file>index.html</welcome-file>
14        <welcome-file>index.jsp</welcome-file>
15        <welcome-file>index.htm</welcome-file>
16        <welcome-file>default.html</welcome-file>
17        <welcome-file>default.jsp</welcome-file>
18        <welcome-file>default.htm</welcome-file>
19    </welcome-file-list>
20 </web-app>

```

4. Création de la vue :

```
ImcModel.java  ControleurServlet.java  web.xml  Vuelmvc.jsp ×
1 <%@page import="web.ImcModel"%>
2 <%@page language="java" contentType="text/html; charset=ISO-8859-1"
3 pageEncoding="ISO-8859-1"%>
4 <%
5 ImcModel model = (ImcModel) request.getAttribute("ImcModel");
6 %>
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <meta charset="ISO-8859-1">
11 <title>IMC</title>
12 </head>
13 <body>
14 <div>
15 <form action="calculerIMC.do" method="post">
16 Poids :<input type="text" name="poids" value="<%=model.getPoids()%>"> en
17 Taille :<input type="text" name="taille" value="<%=model.getTaille()%>">
18 <input type="submit" value="Calculer">
19 </form>
20 </div>
21 <p></p>
22 <div>
23 IMC :<%=model.getImc()%>
24 </div>
25 </body>
26 </html>
```



Poids : 2 en kg
Taille : 2.0 en m
Calculer
IMC :0.5

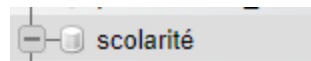
Exercice 2 : Création d'une application de gestion des notes (Gestion_Note)

Nous souhaitons créer une application web J2EE qui permet de gérer les notes des étudiants en respectant le model MVC. L'application doit permettre de :

- Saisir le numéro d'inscription d'un étudiant,
- En validant le formulaire, afficher les notes de cet étudiant en affichant la moyenne de ses notes.







A. Couche Donnée :

1. Créer la base de données Scolarité



2. Créer la table Notes :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	Id_Note	int			No	None			Change Drop More
<input type="checkbox"/> 2	Num_Ins	varchar(20)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/> 3	Matiere	varchar(20)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/> 4	Note	float			No	None			Change Drop More

<div><div><div><div></div><div></div><div></div></div><div></div><div></div></div></div>						Id_Note	Num_Ins	Matiere	Note
<input type="checkbox"/>		Edit		Copy		Delete	12 123	JEE	18
<input type="checkbox"/>		Edit		Copy		Delete	13 123455	English	16

B. Couche Métier :

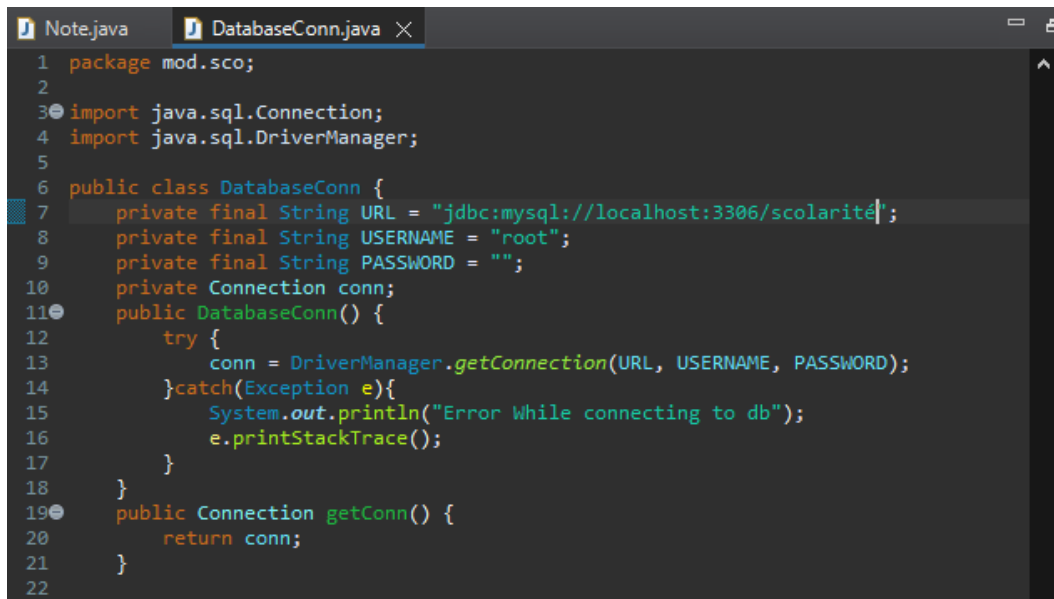
1. Créer le package mod.sco
2. Créer le model : Classe Note.java

```

Note.java X
1 package mod.sco;
2
3 public class Note {
4     private int id_note;
5     private String num_Ins;
6     private String matiere;
7     private float note;
8     public int getId_note() {
9         return id_note;
10    }
11    public void setId_note(int id_note) {
12        this.id_note = id_note;
13    }
14    public String getNum_Ins() {
15        return num_Ins;
16    }
17    public void setNum_Ins(String num_Ins) {
18        this.num_Ins = num_Ins;
19    }
20    public String getMatiere() {
21        return matiere;
22    }
23    public Note(int id_note, String num_Ins, String matiere, float note) {
24        super();
25        this.id_note = id_note;
26        this.num_Ins = num_Ins;
27        this.matiere = matiere;
28        this.note = note;
29    }
30    public void setMatiere(String matiere) {
31        this.matiere = matiere;
32    }
33    public float getNote() {
34        return note;
35    }
36    public void setNote(float note) {
37        this.note = note;
38    }
39 }
40

```

3. Créer la classe DatabaseConn



```
1 package mod.sco;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5
6 public class DatabaseConn {
7     private final String URL = "jdbc:mysql://localhost:3306/scolarité";
8     private final String USERNAME = "root";
9     private final String PASSWORD = "";
10    private Connection conn;
11    public DatabaseConn() {
12        try {
13            conn = DriverManager.getConnection(URL, USERNAME, PASSWORD);
14        } catch (Exception e) {
15            System.out.println("Error While connecting to db");
16            e.printStackTrace();
17        }
18    }
19    public Connection getConn() {
20        return conn;
21    }
22 }
```

4. Créer la classe `Scolarite.java`. Cette classe dispose d'une seule fonction `Vector getNotes(String num_Ins)` qui permet de retourner les notes d'un étudiant dont le num d'inscription est passé en paramètre.


```
Note.java DatabaseConn.java Sclarite.java X
1 package mod.sco;
2 import java.sql.Connection;
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5
6 import java.util.Vector;
7
8 public class Sclarite {
9     public Vector<Note> getNotes(String num_Ins){
10         Vector<Note> vectNotes = new Vector<Note>();
11         try {
12             DatabaseConn DBConn = new DatabaseConn();
13             Connection conn = DBConn.getConn();
14             String sql = "SELECT * FROM notes WHERE Num_Ins = ?";
15             PreparedStatement preparedStatement = conn.prepareStatement(sql);
16             preparedStatement.setString(1, num_Ins);
17             ResultSet resultSet = preparedStatement.executeQuery();
18             while (resultSet.next()) {
19                 int Id_note = resultSet.getInt("Id_Note");
20                 String Num_Ins = resultSet.getString("Num_Ins");
21                 String Matiere = resultSet.getString("Matiere");
22                 float note = resultSet.getFloat("Note");
23                 Note noteObj = new Note(Id_note, Num_Ins, Matiere, note);
24                 vectNotes.add(noteObj);
25             }
26         } catch (Exception e) {
27             System.out.println("error while getting data from database");
28             e.printStackTrace();
29         }
30     }
31     return vectNotes;
}
```

5. Créer une classe TestMetier. Java qui permet de tester la validité de couche métier

```
Note.java DatabaseConn.java Sclarite.java TestMetier.java X
1 package mod.sco;
2
3 import java.util.Vector;
4
5 public class TestMetier {
6 public static void main(String[] args) {
7     Sclarite sc = new Sclarite();
8     try {
9         Vector<Note> notes = sc.getNotes("12");
10        for(int i=0;i<notes.size();i++) {
11            Note note = notes.get(i);
12            System.out.println("matiere : "+note.getMetiere()+" note : "+note.getNote()+"\n");
13        }
14    }catch(Exception e) {
15        System.out.println("error");
16        e.printStackTrace();
17    }
18
19
20 }
21 }
22 }
```

```
Note.java DatabaseConn.java Sclarite.java TestMetier.java X
1 package mod.sco;
2
3 import java.util.Vector;
4
5 public class TestMetier {
6 public static void main(String[] args) {
7     Sclarite sc = new Sclarite();
8     try {
9         Vector<Note> notes = sc.getNotes("123");
10        for(int i=0;i<notes.size();i++) {
11            Note note = notes.get(i);
12            System.out.println("matiere : "+note.getMetiere()+" note : "+no
13        }
14    }catch(Exception e) {
15        System.out.println("error");
16        e.printStackTrace();
17    }
18
19
20 }
21 }
22 }
```

Problems Servers Terminal Data Source Explorer Properties Console X

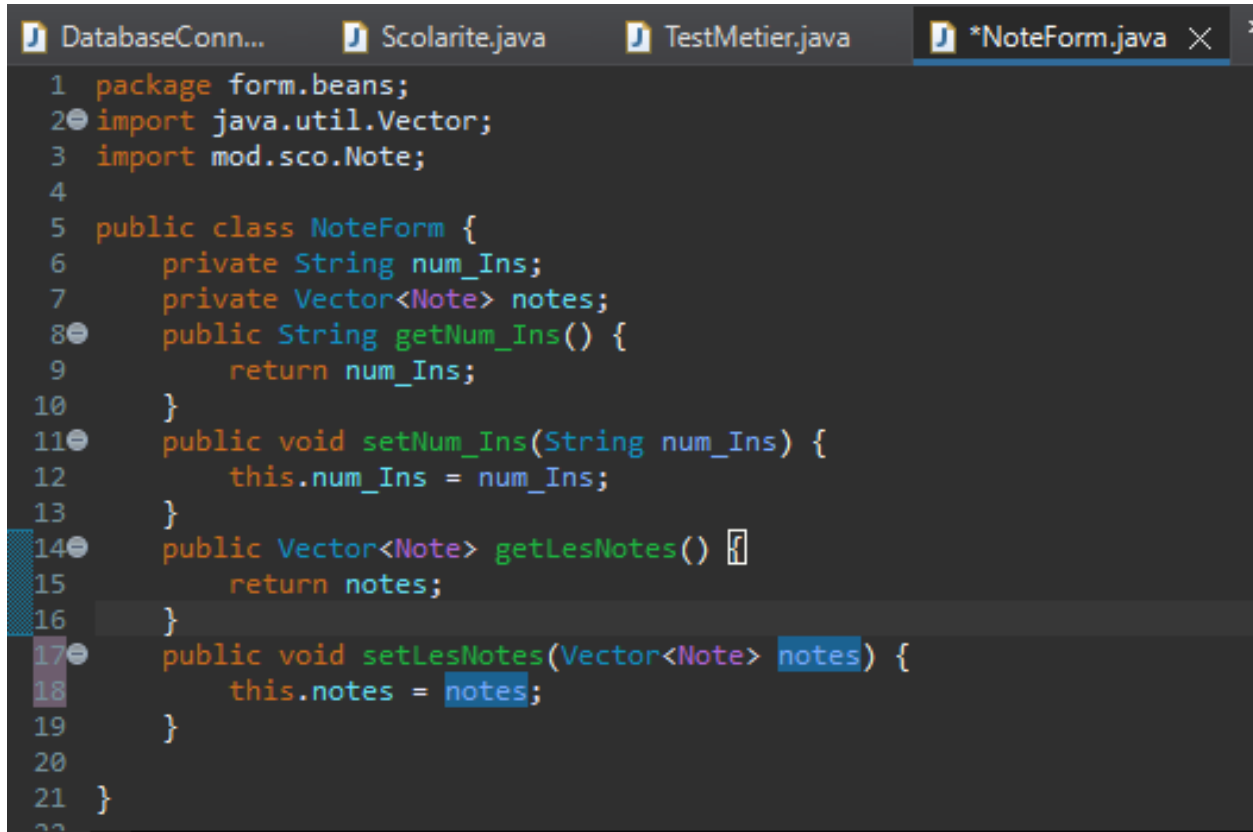
<terminated> TestMetier (1) [Java Application] D:\Users\ibtissam\Documents\eclipse-jee-2023-09-R-win32

matiere : JEE note : 18.0

C. Couche Web :

1. Créer la package form.beans;

2. Créer la classe NoteForm.java qui permet de stocker le N° d'inscription saisie par l'utilisateur et les notes récupérer de la BD.

A screenshot of an IDE window showing the code for NoteForm.java. The window has tabs for DatabaseConn..., Sclarite.java, TestMetier.java, and *NoteForm.java. The code is as follows:

```
1 package form.beans;
2 import java.util.Vector;
3 import mod.sco.Note;
4
5 public class NoteForm {
6     private String num_Ins;
7     private Vector<Note> notes;
8     public String getNum_Ins() {
9         return num_Ins;
10    }
11    public void setNum_Ins(String num_Ins) {
12        this.num_Ins = num_Ins;
13    }
14    public Vector<Note> getLesNotes() {
15        return notes;
16    }
17    public void setLesNotes(Vector<Note> notes) {
18        this.notes = notes;
19    }
20
21 }
```

3. Créer le contrôleur ControleurServlet.java

4. Modifier la methode doPost()

```

1 package mod;
2
3 import java.io.IOException;
15
16 public class ControleurServlet extends HttpServlet{
17     /**
18      *
19      */
20     private static final long serialVersionUID = 1L;
21
22     @Override
23     public void doPost(HttpServletRequest req, HttpServletResponse rep)
24         throws ServletException, IOException{
25         Sclarite sco = new Sclarite();
26         NoteForm nf = new NoteForm();
27
28         String num_Ins = req.getParameter("num_Ins");
29         System.out.println(num_Ins);
30         nf.setNum_Ins(num_Ins);
31         Vector<Note> Notes = sco.getNotes(num_Ins);
32         nf.setLesNotes(Notes);
33
34         HttpSession session = req.getSession();
35         session.setAttribute("nf", nf);
36         rep.sendRedirect("Notes.jsp");
37     }
38 }
39
40

```

5. Editer le fichierweb.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://
3     <display-name>GestionNotes</display-name>
4     <servlet>
5         <servlet-name>cnt</servlet-name>
6         <servlet-class>mod.ControleurServlet</servlet-class>
7     </servlet>
8     <servlet-mapping>
9         <servlet-name>cnt</servlet-name>
10        <url-pattern>*.php</url-pattern>
11    </servlet-mapping>
12 </web-app>

```

6. Créer la vue Notes.jsp et ajouter les scriptlets nécessaires pour calculer et afficher la moyenne

```
Note.java  ControleurS...  web.xml  Notes.jsp  »4
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2    pageEncoding="ISO-8859-1"%>
3  <%@ page import="java.util.Vector" %>
4  <%@ page import="mod.sco.Note" %>
5  <%@ page import="form.beans.NoteForm" %>
6  <!DOCTYPE html>
7  <html>
8  <head>
9    <meta charset="ISO-8859-1">
10   <title>Insert title here</title>
11 </head>
12 <body>
13 <%
14   NoteForm nf = (NoteForm)session.getAttribute("nf");
15   Vector<Note> dataVect=null;
16   if(nf!=null){
17     dataVect = nf.getLesNotes() ;
18   }
19 %>
20 <form action="fetchNote.php" method = "post">
21   Num Inscription : <input type="text" name="num_Ins" >
22   <input type="submit" value="OK">
23 </form>
24 <table>
25   <tr>
26     <th>Matière</th>
27     <th>Note</th>
28   </tr>
29   <%
30     if(dataVect!=null){
31       for (Note note : dataVect){
32         %>
33         <tr>
34           <td>${note.matiere}</td>
35           <td>${note.note}</td>
36         </tr>
37       <}}}%>
38 </table>
39 </body>
40 </html>
```

