

TD2 – JSP

Exercice 1 :

Ecrire une page JSP qui lit deux valeurs entières positives à partir de deux zones textes d'un formulaire et qui retourne comme **résultat une page web affichant un tableau** dont le nombre de lignes est égal à la première valeur et le nombre de colonnes est égal à la deuxième valeur. Le contenu d'une cellule de ce tableau est la paire (i,j) ou i (j) est la ligne(colonne) dans la quelle elle se trouve .

Dans le cas où l'une des valeurs lue n'est pas un entier positif il y aura affichage du message « **format non valide** ».

Exercice 2 :

Soit une page JSP qui contient une zone de texte et un bouton *submit*.

Ecrire une page JSP génèrent un fichier HTML qui contient des listes non ordonnée () contenant des nombres entier aléatoires.

(Le nombre de liste est celui récupéré de la zone de texte)

Exercice 3 :

Soit un formulaire avec les champs suivants :

- Deux zones de textes pour saisir le nom et le prénom de l'utilisateur
- Des cases à cocher pour la saisie des Langages de programmations connus : Java, C, C++, c#, PHP, ...
- Une liste déroulante des sections : GCR, GCV, GCP, GEA
- Deux boutons *submit* et *reset*

Ecrire le code de la page JSP qui affiche les valeurs reçues par le formulaire.

Exercice 4 :

Dans cet exercice, nous allons faire une authentification avec un login et un mot de passe en utilisant un Java Bean représenté par la classe User. Cette authentification va être faite avec les actions JSP.

1. Créer la classe User (login, password).
2. Créer le formulaire d'authentification (Voir Figure 1).
3. Au clic sur le bouton "Envoyer" les données seront transmises vers la page "load.jsp", dans cette page créer un objet user et enregistrer le dans une session.
4. Créer une page web "affiche.jsp" permettant d'afficher le login et le mot de passe que l'utilisateur a saisis (Voir Figure 2).

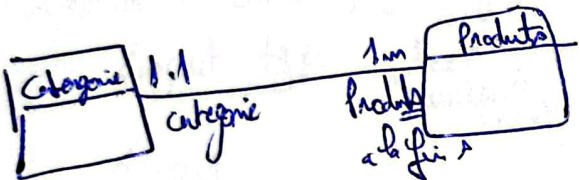
1) Transformer une classe à des entités
 @ entity

```
Public class Catégorie {
    private long id;
    private String nom cat;
    private String description cat;
```

@Id
 // unique
 // id de l'entité

@GeneratedValue

// getters, setters
 // les constructeurs



sig one to many

// crée liste Produits après catégorie

// ~~Association~~ association entre Catégorie et Produits
 @ OneToMany (mapped by = "catégorie")

Private list <Produit> Produits

sig une catégorie
 Possède plusieurs
 Produits

class Produit

@ entity

Public class Produit {

@Id

@GeneratedValue

Private long id;

Private String nom Prod;

Private double Prix Prod;

Private Date date Creation;

// ajout attribut catégorie

@ManyToOne
 Private catégorie cat;

}

sig plusieurs

Produit Possède une seule
 catégorie (catégorie)

* Ajout d'un attribut à une
 entité (Produit)

@ ManyToOne

Private catégorie cat

↑
 ajout attribut dans
 la class

lecture model

@Entity

```
public class Student {
```

@Id

```
private String name;
```

```
public Student();
```

```
+ // getters / setters
```

another class

@Entity

```
public class
```

@Id

```
private int id;
```

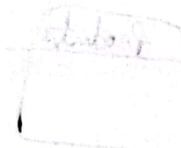
```
public class
```

@Id

```
private String date;
```

```
private Date;
```

```
}
```



Student < Student >

Collaboration de servlet

```
@webServlet("name="seu1", urlPatterns = {"/index"})
public class seu1 extends HttpServlet {
```

```
<head>
<body>
<form name="F01" method="post" action="seu1exp">
<p>
    valeur 1: <input type="text" name="valeur1">
    <input type="submit" value="s'envoyer">
</p>
</form>
</body>
```

```
<form action="page2.jsp" method="Post">
<input type="text" min="1" name="val1">
<input type="text" name="val2">
<input type="submit" value="envoyer">
</form>
```

```
page2.jsp: try {
<% int v1 = Integer.parseInt(request.getParameter("val1"));
    int v2 = Integer.parseInt(request.getParameter("val2"));
    catch (Exception e) { %> fNonValide %> }
}
```

objet implémente sans l'opération

recupérer les variables puisque nous sommes dans le servlet

```
<table border="1">
<% for (i=1; i<=v1; i++) { %>
```

```
<tr>
<% for (j=1; j<=v2; j++) { %>
```

```
<td>
<% = i + " " + j %> </td>
```

```
<% } %> <% } %> </table>
```

<table>
entete = AB

00	01	02
----	----	----

request.getAttribute("obj")
obj = obj donne ma request

```

<form action = "load.jsp" methode = "post" >
  <input type = "text" name = "login" >
  <input type = "text" name = "password" >
  <input type = "submit" value = connexion >
</form>

```

load.jsp

```

<% String l = request.getParameter("login"); String p = request.getParameter("password");
  User u = new User(l,p);
  session.setAttribute("u",u);
  response.sendRedirect("affiche.jsp"); %>

```

cote session ↳ redirection
 si objet requ = cote client

affiche.jsp

```

<% User u = session.getAttribute("u"); %>
<% if (u != null) { %>
  <p> login : <% = u.getLogin() %> </p>
  <p> pass : <% = u.getPassword() %> </p>
  <% } %>

```



```

<form action="index.jsp"
  method="post"
  name="vaP1" >
  vaP2 <input type="text" name="vaP2">
</form>

```

```
try {
```

```

  int v1 = Integer.parseInt(request.getParameter("vaP1")) ;
  int v2 = Integer.parseInt(request.getParameter("vaP2")) ;

```

```
<table>
```

```

for for (int i = 1; i <= v1; i++) {

```

```
  <tr>
```

```

    <td> for (int j = 1; j <= v2; j++) {

```

```

      <td> <%= i + " " + j %> </td>

```

```

    }
  }

```

```
</table>
```

```

<form action="page.jsp" method="post"

```

```

  vaP1 <input type="text" name="vaP1"

```

```

  vaP2 <input type="text" name="vaP2"

```

```
</form>
```

```
try {
```

```

  int v1 = Integer.parseInt(request.getParameter("vaP1")) ;

```

```

  int v2 = Integer.parseInt(request.getParameter("vaP2")) ;

```

```
<table>
```

```

  <tr> for (int i = 1; i <= v1; i++) {

```

```

    <td> for (int j = 1; j <= v2; j++) {

```

```

      <td> <%= i + " " + j %> </td>

```

```

    }
  }
</table>

```



```

Public class user {
    login
    pass
    // setter & getter
}
  
```

~~<jsp:useBean id="user"~~

scope = request >

```

<form action="load.jsp" method="get">
  login <input type="text" name="login">
  pass <input type="text" name="pass">
  <input type="submit" action="load.jsp" value="Envoyer">
</form>
  
```

load.jsp

```

<jsp:useBean id="user" class="model.user" scope="request">
  <jsp:setProperty property="login" name="user">
  <jsp:setProperty property="pass" name="user">
  <!-- session.setAttribute("user", user) -->
  <!-- response.sendRedirect("affiche.jsp"); -->
  
```

affiche.jsp

```

user u = new user()
  
```

```

String login = request.getParameter("login");
String pass = request.getParameter("pass");
  
```

```

// user u = session.getAttribute("u");
  
```

```

// if (u != null) {
  
```

```

    login : % = u.getLogin() % </P>
  
```

```

    pass : % = u.getPass() % </P>
  
```

```

  } %
  
```


response : côté client
 request : côté serveur
 redirection : make
 hypertext : objet

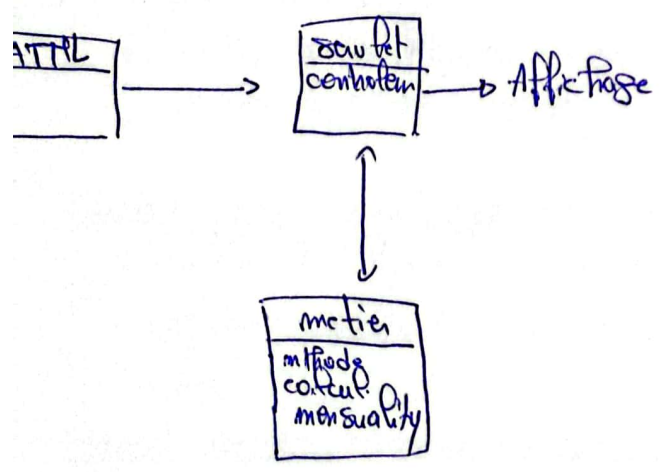
regeneration de page = getParametre.

@ (EPL : Entity) : annotation.

> src => node / controller / DAO

> lib => vue

p :



```

public class metier {
    public double montant;
    public int taux;
    public int durée;
  }
  
```

```

public metier () {
  }
  
```

```

public class implmetier implements metier {
    public double calcul(int m,
        double c, int t) {
        double A, B, k, n;
        k = c * t / 12;
        A = 1 - (1 + k / m);
        B = A * Math.pow(m, n);
        // m p m = k / B;
        return (m);
    }
  }
  
```

metier : contient les
 méthode de metier
 besoin

import java.io.*; Exception;

```

public class Servlet extends HttpServlet {
    public void doGet(request req, response rep) {
        double montant = Double.parseDouble(req.getParameter("montant"));
        int durée = Integer.parseInt(req.getParameter("duree"));
        int taux = Integer.parseInt(req.getParameter("taux"));
        credit c = new credit();
        double mens = c.calcul(montant, taux, durée);
        response.getWriter().println("Le montant est " + mens);
    }
  }
  
```

Sărbătorile: clasa : creșterea lădărilor - heite na HTR șerwa
 creșterea porțur JPM JVM
 comob-șter uue el ȘB ȘB



3) traitement de request service()

2) départ zero (lancement de l'App)
doGet()

best \rightarrow do best

lien App et texte \leadsto doc et

Other

I-bind bib: I know