Filière :	Développement des Systèmes d'Information - DSI	Durée :	4 heures
Épreuve :	Développement des Applications Informatiques - DAI	Coefficient :	45

ÉTUDE DE CAS : ASSOCIATION DES PERSONNES PORTEUSES DE TRISOMIE 21

DOSSIER I : RÉCUPÉRATION DES INFORMATIONS D'UNE FORMATION

❖ Sous Dossier 1-1 : STRUCTURE DE DONNÉES *(14 pts)*

1. Implémentation de la classe « **Formation** » :

```
public class Formation implements Serializable{
private String codef, libelle;
 private Date dateDebut, dateFin;
a)                      (1 pt)
public Formation(String cf, String lib, Date dd, Date df)
{
this.codef= cf;  this.libelle=lib; this.dateDebut = dd; this.dateFin= df;
}
b)                        (1 pt)
   @Override
public boolean equals(Object ob)
{
if(this == ob ) return true;
if(ob == null) return false;
if(ob instanceof Formation == false) return false;
Formation f=(Formation) ob;
return this.codef.equalsIgnoreCase(f.codef);
}
c)                      (1 pt)
   @Override
   public String toString()
   {
 SimpleDateFormat f= new SimpleDateFormat("dd/MM/yyyy");
 return "Code :"+this.codef+", Libelle :"+this.libelle+" effectuée entre
:"+f.format(this.dateDebut)+" et "+f.format(this.dateFin);
       }
}
```

2. Implémentation de la classe « **FormationTheorique** » :

```
public class FormationTheorique extends Formation {
   private String nomFormateur;
   private String salle;
   (1,5 pts)
public FormationTheorique(String cf, String lib, Date dd, Date df,String nf, String s)
{
   super(cf,lib,dd,df);
   this.nomFormateur=nf;
   this.salle=s;
 }
   (1,5 pts)
  @Override
public String toString()
{
 return super.toString() +"\n" + "Formateur :"+this.nomFormateur+ ", salle:"+this.salle;
 }
}
```

3. Implémentation de la classe « **Beneficiaire** » :

```
public class Beneficiaire{
   private String codeB;
   private String nom, prenom;
   private String ville;
   private Date dateNais;
   private ArrayList<Formation> listeFormations;
```

a) (1 pt)

```
public Beneficiaire(String cb, String n, String p, String v,Date d)
{
 this.codeB ; this.nom=n ; this.prenom=p ; this.ville=v ; this.dateNais=d ;
this.listeFormations=new ArrayList() ;
 }
```

b) (1 pt)

```
public boolean addFormation(Formation F)
{
  return this.listeFormations.add(F);
}
   //-------------------------------------------------------
```

c) (1 pt)

```
public Formation delFormation(int index)
{
try{
        return this.listeFormations.remove(index);
}catch(Exception ex){ return null;}
 }
```

d) (1 pt)

```
public boolean delFormation(Formation F)
{
 return this.listeFormations.remove(F);
}
```

e) (1 pt)

```
public Formation setFormation(int index, Formation F)
{
try{
        return this.listeFormations.set(index,F);
}catch(Exception ex){ return null;}

 }
```

f) (1 pt)

```
@Override
public String toString()
{
String m= ”Code Ben :”+this.codeB+”Nom complet : ”+ this.nom+” ”+this.prenom+” ayant
bénéficié des formations:\n”;
int i=1;
for(formation f : this.listeFormations)
{
    m+= i+” - ”+f.toString()+”\n”;
}
return m;
 }
```

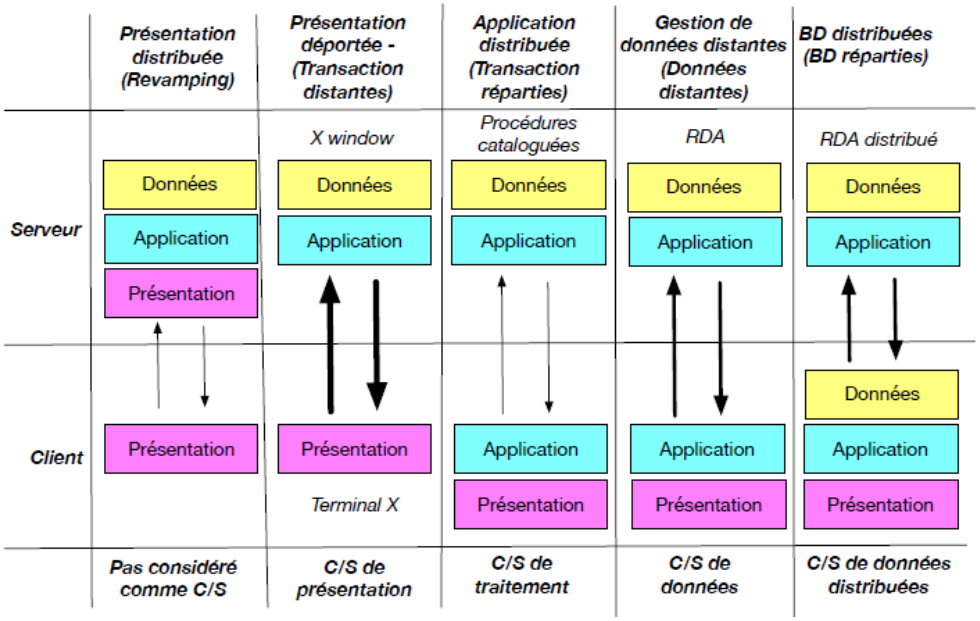
g) (2 pts)

```
public void saveFormations(String file)
{
   try{
         ObjectOutputStream out=new ObjectOutputStream(FileOutputStream(file));
         for(Formation f : this.listeFormations)
          {
             if(f instanceof FormationTheorique)
             {
               out.writeObject(f);
               out.flush();
              }
         }
          Out.close();
       }catch(Exception ex){}
 }
}
```

❖ Sous Dossier 1-2 : ARCHITECTURE CLIENT/SERVEUR *(8 pts)*

1. C'est une architecture 2 tiers puisque c'est une liaison directe entre un client et un serveur qui dispose d'une base de données. *(0,5 pt)*

2. Donner le modèle de Gartner-group correspondant à cette architecture. *(1 pt)*



3. Classe : Client TCP *(3 pts)*

Implémenter la classe client qui contient :

- ✓ **3** attributs ;
- ✓ Un constructeur avec **2** arguments qui crée un socket et initialise les objets d'entrée/sortie **in** et **out** ;
- ✓ Une méthode **demandeFormation** qui envoie au serveur **TCP** le code d'une formation et qui reçoit et retourne l'objet formation correspondant à ce code.

```
public class Client {
private Socket sc=null;
private ObjectInputStream in = null;
private PrintWriter out=null;
public Client(String IP, int Port)
{
 try{
 this.sc=new Socket(IP, Port) ;
 this.out=new PrintWriter(this.sc.getOutputStream(),true) ;
 this.in=new ObjectInputStream(this.sc.getInputStream()) ;
}catch(Exception ex){}
}
public Formation demandeFormation(String cf)
{
try{
 this.out.println(cf) ;
 return (Formation)this.in.readObject() ;
}catch(Exception ex){return null ;}
}
}
```

4. Classe : Serveur TCP *(3,5 pts)*

```
public class Serveur {
private ServerSocket ss;
private Socket sc=null;
private BufferedReader in = null;
private ObjectOutputStream out= null;
public void startServer()
{ String c ;
  try{
    //[1]
   this.ss=new ServerSocket(2000) ;
   this.sc=ss.accept() ;
this.in=new BufferedReader(new InputStreamReader(ss.getInputStream())) ;
this.out=new ObjectOutputStream(ss.getOutputStream()) ;
    while(true)
     {
         //... [2]
       c= in.ReadLine() ;
       out.writeObject(this.getFormation(c)) ;
       out.flush() ;
     }
    }catch(Exception ex) {}
}
public Formation getFormation(String cf)
{ /*  code non demandé   */}
}
```

DOSSIER II : PLANIFICATION DES FORMATIONS. *(10pts)*

1) Les objets de connexion et la procédure « **Connexion**» ***(1,5 pts)***

Mode connecté / non connecté

```
Module Module1
    Public Con As SqlConnection
    Sub Connexion()
        Try
 Con = New SqlConnection("initial catalog=db_formation;data
source=192.168.1.200;integrated security=true;")
   Con.Open()
       Catch ex As Exception
            MessageBox.Show("Echec de connexion")
        End Try
    End Sub
End Module
```

2) Dans la classe de l'IHM, Écrire le code de la procédure « **charger_CodeForm** » qui charge le comboBox « **cmbCodeF »** par les codes de toutes les formations. ***(1,5 pts)***

Mode connecté

```
Private Sub charger_CodeForm()
      Dim cmd As New SqlCommand("select codef from formation", Con)
        Dim res As SqlDataReader = cmd.ExecuteReader
        Dim t As New DataTable
        t.Load(res)
        res.Close()
        cmbCodeF.DisplayMember = "codef"
        cmbCodeF.ValueMember = "codef"
        cmbCodeF.DataSource = t
End Sub
```

Mode non connecté

```
Private Sub charger_CodeForm()
        Dim ds As New DataSet
       Dim adp As New SqlDataAdapter("select codef from formation", Con)
        adp.Fill(ds, "form") ;
        cmbCodeF.DisplayMember = "codef"
        cmbCodeF.ValueMember = "codef"
        cmbCodeF.DataSource = ds.Tables("form")
End Sub
```

3) Écrire le code de la fonction « **getLibelleForm** » qui reçoit, comme argument, le code d'une formation et qui retourne son libellé. ***(1,5 pts)***

Mode connecté

```
Private Function getLibelleForm(ByVal CodeF As String ) As String
   Dim cmd As New SqlCommand("select libelle from formation where codef='" &
    code & "'", Con)
   Dim r As String = cmd.ExecuteScalar
   Return r
End Sub
```

Mode non connecté

```
Private Function getLibelleForm(ByVal CodeF As String ) As String
    Dim ds As New DataSet
    Dim adp As New SqlDataAdapter("select libelle from formation where codef='" &
    code & "'", Con)
    adp.Fill(ds, "fo")
   Dim r As String = ds.Tables("fo").Rows(0)(0)
   Return r
End Sub
```

4) Écrire le code de la procédure «**affiche_listeForm** » qui liste toutes les formations correspondantes à un bénéficiaire dans l'objet DataGridView nommé « **DGVListe ».** La procédure prend en argument le code d'un bénéficiaire. Voici un aperçu de l'objet DataGridView. ***(2 pts)***

Mode connecté

```
Private Sub affiche_listeForm(ByVal codeBen As String)
   Dim cmd As New SqlCommand("select f.codef as [Code Formation],libelle,datedeb as
[Date  Début],datefin as [Date Fin] from suivre s inner join formation f on
s.codef=f.codef inner join beneficiaire b on s.codeb = b.codeb where b.codeb = '" &
codeb & "'", Con)
   Dim res As SqlDataReader = cmd.ExecuteReader
   Dim t As New DataTable
   t.Load(res)
   res.Close()
   DGVListe.DataSource = t
End Sub
```

Mode non connecté

```
Private Sub affiche_listeForm(ByVal codeBen As String)
   Dim ds As New DataSet
  Dim adp As New SqlDataAdapter("select f.codef as [Code Formation],libelle,datedeb
as [Date  Début],datefin as [Date Fin] from suivre s inner join formation f on
s.codef=f.codef inner join beneficiaire b on s.codeb = b.codeb where b.codeb = '" &
codeb & "'", Con)

   adp.Fill(ds, "liste")
   DGVListe.DataSource = ds.Tables("liste")
End Sub
```

5) Écrire le code de la fonction « **keyExist** » qui vérifie l'existante de la clé primaire de la table « **SUIVRE ».** Le code du bénéficiaire et celui de la formation sont donnés en arguments. ***(1,5 pts)***

Mode connecté

```
Private Function keyExist(ByVal CodeF As String,ByVal codeB As String) As Boolean
     Dim cmd As New SqlCommand("select count(*) from suivre where codef='" &
codef & "' and codeb='" & codeb & "'", Con)

    Dim n As Integer = cmd.ExecuteScalar
    Return n <> 0
End Function
```

Mode non connecté

```
Private Function keyExist(ByVal CodeF As String,ByVal codeB As String) As Boolean
 Dim ds As New DataSet
 Dim adp As New SqlDataAdapter("select * from suivre", Con)
 adp.Fill(ds, "suivre")
  Dim t() As DataRow = ds.Tables("suivre").select(codef='" &    codef & "'
and codeb='" & codeb & "'")
    Return t.Length <> 0
End Function
```

6) On suppose que l'objet comboBox « **CmbBen** » est rempli par les noms complets de bénéficiaires et que les dates de début et de fin de la formation sont valides. ***(2 pts)***

Mode connecté

```
Private Sub cmdAjouter_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdAjouter.Click
   Dim cf As String = cmbCodeF.SelectedValue.ToString
   Dim cb As String = cmbBen.SelectedValue.ToString

    If keyExist(cb, cf) Then
            MessageBox.Show("Formation deja attribuée à ce bénéficiaire")
            Exit Sub
    End If

    Dim req As String = "insert into suivre(codef,codeb,datedeb,datefin) values('"
& cf & "','" & cb & "','" & dtpDebut.Value & "','" & dtpFin.Value & "')"

    Dim cmd As New SqlCommand(req, Con)
    cmd.ExecuteNonQuery()
    afficher_listeform(cb)
End Sub
```

Mode non connecté

```
Private Sub cmdAjouter_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdAjouter.Click
   Dim cf As String = cmbCodeF.SelectedValue.ToString
   Dim cb As String = cmbBen.SelectedValue.ToString
   Dim ds As New DataSet
   Dim adp As New SqlDataAdapter("select * from suivre",con)
    If keyExist(cb, cf) Then
            MessageBox.Show("Formation deja attribuée à ce bénéficiaire")
            Exit Sub
    End If
    Dim R As DataRow = ds.Tables("suivre").NewRow
    R(0)=cb
    R(1)=cf
    R(2)=dtpDebut.value
    R(3)=dtpFin.value
    ds.Tables("suivre").Rows.Add(R)
    Dim cb As new SqlCommandBuilder(adp)
    adp.update(ds,"suivre")
    afficher_listeform(cb)
End Sub
```

DOSSIER III : GESTION DES FORMATIONS. *(8pts)*

1) Le script « **connexion.php** » permettant la connexion à la base de données « **db_formation** ».

```
<?php
   $con=mysqli_connect('Srv_ATRI21','association','association123','db_formation');
  if (mysqli_connect_errno()){
            die "erreur de connection".mysqli_connect_erreur();
   }
?>
```

2) La page d'accueil : **index.php** affiche la liste des formations : ***(3 pts)***

Le script « **index.php** » est le suivant :

```
<html>
  <head>
       <title> tableau de bord administrateur </title>
              <style type="text/css"> … </style>
  </head>
  <body>
       <?php  include('connexion.php');  ?>
        <fieldset>
       <h1> Liste des formations </h1>
       <table>
              <tr>
                <th> Code Formation </th>
                <th> Libellé </th>
                <th> </th>
               </tr>
             <?php
               // partie à compléter
          $req="select * from formation";
          if ($result=mysqli_query($con,$req)){
            while ($row=mysqli_fetch_row($result)){
              $codef=$row['0'];
              $libfor =$row['1'];
              echo "<tr>   <td>". $row[0]. "</td> <td>". $row[1]. "</td> <th>
  <a href='lstben.php?cf=$codef'> Liste b&eacute;n&eacute;ficiaires </a> </th>
  </tr>";}
       }
  ?>
       </table>
       </fieldset>
  </body>
</html>
```

3) La page « **lstben.php** » est la suivante : *(3 pts)*

Le script « **lstben.php** » est le suivant :

```
<html>
<body>
<?php include('connexion.php') ; ?>

$req="select libelle from formation where codef='".$_GET['cf']."'";
$result=mysqli_query($con,$req);
$f="";
if($row=mysqli_fetch_row($result)) $f=$row[0];
?>
<h1> Liste des b&eacute;n&eacute;ficiaires de la formation : <br><?=  $f; ?> </h1>
<table>
      <tr><th> Code Beneficiaire </th><th>Nom</th><th> Prenom</th>
      <th> Date Naissance</th> <th> Ville</th></tr>
<?php
   $req="select B.CodeB,Nom,Prenom,dateNais,ville from suivre S inner join
beneficiaire B On S.CodeB= B.codeB where codef='".$_GET['cf']."'";
   if ($result=mysqli_query($con,$req)){
         while ($row=mysqli_fetch_row($result)){
           echo "<tr><td>". $row[0]. "</td> <td>".$row[1]."</td> <td>" .
$row[2] . "</td> <td>". $row[3]."</td> <td>". $row[4]. "</td></tr>";   }
      }
 ?>
</table>
      <center><a href="index.php">Retour</a></center>
</body>
</html>
```

% Bon courage %