



| | |
|-----------|--|
| Filière : | Développement des Systèmes d'Information – DSI – |
| Épreuve : | Développement des Applications Informatiques – DAI – |

| | |
|---------------|----------|
| Durée : | 4 heures |
| Coefficient : | 45 |

ÉTUDE DE CAS : Plateforme d'enseignement à distance

DOSSIER I : SERVICE PÉDAGOGIQUE

❖ SOUS-DOSSIER 1-1 : STRUCTURE DE DONNÉES

(14 pts)

1. Implémentation de la classe « Membre » :

```
public class Membre implements Serializable{
    private String code;
    private String nom,prenom;
    private Date dateNais;
// a) (1 pt) -----
public Membre(String code, String nom, String prenom, Date dateNais) {
    this.code = code;
    this.nom = nom;
    this.prenom = prenom;
    this.dateNais = dateNais;
}
b) (1 pt) -----
public String getCode() {
    return code;
}
public String getNom() {
    return nom;
}
public String getPrenom() {
    return prenom;
}
c) (1 pt) -----
@Override
public boolean equals(Object obj) {
    if (obj == null) return false;
    if(this==obj) return true;
    if (obj instanceof Membre == false) return false;
    Membre ob = (Membre) obj;
    return this.code.equalsIgnoreCase(ob.code);
}
d) (1 pt) -----
@Override
public String toString() {
    SimpleDateFormat f=new SimpleDateFormat("dd/MM/yyyy");
    return "code : " + code + ", Nom Complet: " + nom + " "+ prenom + ", Date
naissance : " + f.format(dateNais);
}
}
```

2. Implémentation de la classe « Enseignant » :

```
public class Enseignant extends Membre{
    private String email, lieuTravail, fonction;

    a) (1 pt) -----
    public Enseignant( String code, String nom, String prenom, Date dateNais,String
    email, String lieuTravail, String fonction) {
        super(code, nom, prenom, dateNais);
        this.email = email;
        this.lieuTravail = lieuTravail;
        this.fonction = fonction;
    }

    b) (1 pt) -----
    @Override
    public String toString() {
        return super.toString()+"\nEnseignant [Email : " + email + ", lieu de
    travail : " + lieuTravail + ", fonction : " + fonction + "];
    }
}
```

3. Implémentation de la classe « Equipe » :

```
public class Equipe {
    private int numEq;
    private String nomEq;
    private ArrayList<Membre> liste;

    a) (1 pt) -----
    public Equipe(int numEq, String nomEq) {
        this.numEq = numEq;
        this.nomEq = nomEq;
        liste=new ArrayList(); // ou liste=new ArrayList<Membre>
    }

    b) (1 pt) -----
    public boolean addMembre(Membre m)
    {
        return liste.add(m);
    }

    c) (1 pt) -----
    public Membre delMembre(int index)
    {
        try {
            return liste.remove(index);
        } catch (Exception e) {
            return null;
        }
    }
    /* ou -----
    public Membre delMembre(int index)
    {
        if(index<0 || index>= liste.size()) return null;
        return liste.remove(index);
    }
    */

    d) (1 pt) -----
    public boolean delMembre(Membre m)
    {
        return liste.remove(m);
    }
}
```

e) (1 pt) -----

```
public Membre updateMembre(int index,Membre m)
{
    try {
        return liste.set(index,m);
    } catch (Exception e) {
        return null;
    }
}
/* ou -----
public Membre updateMembre(int index, Membre m)
{
    if(index<0 || index>= liste.size()) return null;
    return liste.set(index,m);
}
*/
```

f) (1,5 pt) -----

```
@Override
public String toString() {
    String msg="Equipe[numéro : " + numEq + ", Nom Equipe : " + nomEq + "]\n
contient les membres suivants:\n";
    for(int i=0;i<liste.size();i++)
    {
        msg += (i+1) + liste.get(i).toString() + "\n";
    }
    return msg;
}
/* ou -----
public String toString() {
    String msg="Equipe[numéro : " + numEq + ", Nom Equipe : " + nomEq + "]\n
contient les membres suivants:\n";
    int i=1;
    for(Membre m:liste)
    {
        msg += i + m.toString() + "\n";
    }
    return msg;
}
*/
```

g) (1,5 pt) -----

```
public boolean saveEnseignants(String f)
{
    try {
        ObjectOutputStream out=new ObjectOutputStream(new FileOutputStream(f));
        for(Membre m:liste)
        {
            if(m instanceof Enseignant)
            {
                out.writeObject(m);
                out.flush();
            }
        }
        out.close();
        return true;
    } catch (Exception e) {
        return false;
    }
}
}
```

❖ SOUS-DOSSIER 1-2 : ARCHITECTURE CLIENT/SERVEUR

(8 pts)

1. Un serveur d'application s'occupe de toutes les opérations de base d'une application et offre aux clients légers des services distants (0,5 pt)
2. IP du serveur d'application : **192.168.1.20** est une adresse de classe C car l'octet du poids fort qui est **192** est compris entre 192 et 223. (0,5 pt)

3. MÉTHODE 1 : Socket TCP

a) Classe : Client TCP

```
public class Client {
    private Socket sc=null;
    private PrintWriter out=null;
    private ObjectInputStream in=null;
    (2 pts) -----
    public Client(String IP,int port)
    {
        try {
            sc=new Socket(IP,port);
            out=new PrintWriter(sc.getOutputStream(),true);
            in=new ObjectInputStream(sc.getInputStream());
        } catch (IOException e) {
            System.out.println("error");
        }
    }
}
```

```
(2 pts) -----
public ArrayList<Etudiant> getListeEtudiants(int numEq)
{
    try {
        out.println(numEq);
        return (ArrayList<Etudiant>)in.readObject();
    } catch (IOException | ClassNotFoundException e) {
        return null;
    }
} }
```

b) Classe : Serveur TCP

(3 pts)

```
public class Serveur {
    private ServerSocket ss;
    private Socket sc=null;
    private BufferedReader in = null;
    private ObjectOutputStream out= null;
    public void startServer()
    {
        int numEq;
        try{
            //[1] (0,5 pt) -----
            ss=new ServerSocket(3000);
            while(true) {
                //[2] (1,5 pt) -----
                sc=ss.accept();
                in=new BufferedReader(new InputStreamReader(sc.getInputStream()));
                out=new ObjectOutputStream(sc.getOutputStream());
                while(true) {
                    //[3] (1 pt) -----
                    numEq=Integer.parseInt(in.readLine());
                    if(numEq<0) break;
                    out.writeObject(getEtudiants(numEq));
                    out.flush();
                }
            }
        }catch(Exception ex) { System.out.println("Error"); }
    }
    public ArrayList<Etudiant> getEtudiants(int numEq)
    { // Le code source de la méthode getEtudiants est non demandé }
}
```

3. MÉTHODE 2 : Objets partagés RMI**a) Classe : Serveur RMI****(4 pts)**

```
// (1 pt) -----
public Interface Service extends Remote {
    public ArrayList<Etudiant> getListeEtudiants(int numEq) throws
    RemoteException;
}
// .....
public class Serveur extends UnicastRemoteObject implements Service {
    public Serveur() throws RemoteException {}
    // (1 pt) -----
    @Override
    public ArrayList<Etudiant> getListeEtudiants(int numEq) throws
    RemoteException
    { return getEtudiants(numEq); }
    //-----
    public ArrayList<Etudiant> getEtudiants(int numEq)
    { }
    // (2 pts) -----
    public static void main(String [] args) throws Exception {
        LocateRegistry.createRegistry(1099);
        Serveur obj=new Serveur();
        Naming.rebind("rmi://192.168.1.20:1099/OBJ", obj);
    } }
```

b) Classe : Client RMI**(3 pts)**

```
public class Client{
    private Service ob;
    // (1,5 pt) -----
    public Client() throws Exception
    {
        ob=(Service)Naming.lookup("rmi://192.168.1.20:1099/OBJ");
    }
    // (1,5 pt) -----
    public ArrayList<Etudiant> get(int numEq) throws RemoteException
    {
        return ob.getListeEtudiants(numEq);
    }
}
```

DOSSIER II : GESTION DES INSCRIPTIONS

(10 pts)

- 1) Dans un module, la déclaration des objets de connexion et la création de la fonction « **Connexion** » qui établit une connexion avec le serveur de base de données avec la gestion des exceptions. (2 pts)

```
Public con As SqlConnection = Nothing
Public Function Connexion() As Boolean
    Try
        con = New SqlConnection("initial catalog=db_Inscription;data
source=192.168.1.200 ;integrated security=true;")
        con.Open()
        Return True
    Catch ex As Exception
        Return False
    End Try
End Function
```

- 2) Dans la classe de l'IHM (Form), le code de la procédure « **charger_Equipes** » qui charge le comboBox « **cmbEquipe** » par les noms de toutes les équipes. Voici un aperçu de l'objet **cmbEquipe**. (2 pts)

```
'Mode connecté
Private Sub charger_Equipes()
    Dim req As String
    req = "select * from equipe"
    Dim cmd As New SqlCommand(req, con)
    Dim dr As SqlDataReader = cmd.ExecuteReader()

    Dim table As New DataTable
    If dr.HasRows() Then 'test optionnel
        table.Load(dr)
    End If

    dr.Close()
    cmbEquipe.ValueMember = "numeq"
    cmbEquipe.DisplayMember = "nomeq"
    cmbEquipe.DataSource = table
End Sub
'-----
'Mode non connecté
Private Sub charger_Equipes()
    Dim ds As New DataSet
    Dim req As String
    req = "select * from equipe"
    Dim adp As New SqlDataAdapter(req, con)
    adp.Fill(ds, "eq")
    cmbEquipe.DisplayMember = "nomeq"
    cmbEquipe.ValueMember = "numeq"
    cmbEquipe.DataSource = ds.Tables("eq")
End Sub
```

- 3) Le code de la procédure «**affiche_Etudiants**» qui liste tous les étudiants dans l'objet DataGridView nommé «**DGVListe**» dont l'aperçu :

(2 pts)

```
' Mode connecté
Private Sub affiche_Etudiants()

    Dim req As String
    req = "select Code,(nom+' '+ prenom) as [Nom Complet],dateNais as
[Date Naissance],numMassar as
            Massar,nomEq as Equipe from equipe e inner join etudiant
etd on e.numEq=etd.numEq"
    Dim cmd As New SqlCommand(req, con)
    Dim dr As SqlDataReader = cmd.ExecuteReader()

    Dim dt As New DataTable
    If dr.HasRows() Then      'test optionnel
        dt.Load(dr)
    End If
    DGVListe.DataSource = dt
End Sub

'-----
Mode non connecté
Private Sub affiche_Etudiants()
    Dim ds As New DataSet
    Dim req As String
    req = "select Code,(nom+' '+ prenom) as [Nom Complet],dateNais as [Date
Naissance],numMassar as Massar,nomEq as Equipe from equipe e inner join etudiant
etd on e.numEq=etd.numEq"
    Dim adp As New SqlDataAdapter(req, con)
    adp.Fill(ds, "et")
    DGVListe.DataSource = ds.Tables("et")
End Sub
```

- 4) Le code de la fonction «**genererCode**»

(2 pts)

```
'Mode connecté
Private Function genererCode() As Integer
    Dim req As String
    req = "select max(code) from etudiant"
    Dim cmd As New SqlCommand(req, con)
    If IsDBNull(cmd.ExecuteScalar()) Then Return 1
    Return CInt(cmd.ExecuteScalar()) + 1
End Function

'-----
'Mode non connecté
Private Function genererCode() As Integer
    Dim ds As New DataSet
    Dim req As String
    req = "select max(code) from etudiant"
    Dim adp As New SqlDataAdapter(req, con)
    adp.Fill(ds, "eq")
    If IsDBNull(ds.Tables("eq").Rows(0)(0)) Then Return 1
    Return CInt(ds.Tables("eq").Rows(0)(0)) + 1
End Function
```



```
Ou -----
Private Function genererCode() As Integer
    Dim ds As New DataSet
    Dim req As String
    req = "select * from etudiant"
    Dim adp As New SqlDataAdapter(req, con)
    adp.Fill(ds, "eq")
    Dim row() As DataRow = ds.Tables("eq").Select("code= max(code)")
    If row.Length = 0 Then
        Return 1
    End If
    Return CInt(row(0)(0)) + 1
End Function
```

5) Code de la procédure événementielle du bouton « cmdEnregistrer ».

(2 pts)

```
' Mode connecté
Private Sub cmdEnregistrer_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdEnregistrer.Click

    Dim req As String = "insert into etudiant values(" & CInt(txtCode.Text) & ",'" &
txtNom.Text & "','" & txtPrenom.Text & "','" & dtpDateNais.Value & "','" &
txtMassar.Text & "','" & CInt(cmbEquipe.SelectedValue) & ")"

    Dim cmd As New SqlCommand(req, con)
    cmd.ExecuteNonQuery()
    affiche_Etudiants()
End Sub

' -----
' Mode non connecté
Private Sub cmdEnregistrer_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdEnregistrer.Click
    Dim ds As New DataSet
    Dim req As String
    req = "select * from etudiant"
    Dim adpC As New SqlDataAdapter(req, con)
    adpC.Fill(ds, "etudiant")

    Dim lig As DataRow
    lig = ds.Tables("etudiant").NewRow
    lig(0) = CInt(txtCode.Text)
    lig(1) = txtNom.Text
    lig(2) = txtPrenom.Text
    lig(3) = dtpDateNais.Value
    lig(4) = txtMassar.Text
    lig(5) = CInt(cmbEquipe.SelectedValue)
    ds.Tables("etudiant").Rows.Add(lig)

    Dim cb As SqlCommandBuilder
    cb = New SqlCommandBuilder(adpC)
    adpC.Update(ds, "etudiant")
    affiche_Etudiants()
End Sub
```

DOSSIER III : GESTION DES PUBLICATIONS.

(8pts)

- 1) Le code du script « **connecter.php** » permettant la connexion à la base de données « **db_Publication** ».

(2 pts)

Méthode 1 : Classe mysqli

```
<?php
    $servername="Server";
    $username="membre";
    $password="membre123";
    $bd="db_publication";
    $conn = new mysqli($servername, $username, $password,$bd);
    if ($conn->connect_error) {
        die("Connection failed: ".$conn->connect_error);
    }
?>
```

Méthode 2 : Classe PDO

```
<?php
    $servername="Server";
    $username="membre";
    $password="membre123";
    $bd="db_publication";

    try{
    $conn=new PDO("mysql:host=".$servername.";dbname=".$bd, $username,
    $password) ;
    $conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION) ;
    }
    catch(Exception $e)
    {
    die($e->getMessage()) ;
    }
?>
```

- 2) La page d'accueil **index.php** permet à un membre de s'authentifier. Le formulaire d'authentification est le suivant :

(3 pts)

- Le script du fichier <<**validerAuth.php**>>

Méthode 1 : Classe mysqli

```
<?php
    require 'connecter.php';
    $code=$_POST['code'];
    $mdp=$_POST['motpass'];
    $sqlm="select code,nom,prenom from membre where code=".$code." and
motpass='".$mdp."'";
    $resultm=$conn->query($sqlm);
    if ( $resultm->num_rows==0 )
    { header('location:index.php');}
    else
    {
        $rowm = $resultm->fetch_assoc();
        $prenom=$rowm["prenom"];
        $nom=$rowm["nom"];
        $_SESSION['code']=$code;
        $_SESSION['nom']=$nom;
        $_SESSION['prenom']=$prenom;
        header('location:lstpublications.php');
    }
?>
```

Méthode 2 : Classe PDO

```

<?php
    require 'connecter.php';
    $code=$_POST['code'];
    $mdp=$_POST['motpass'];
    $sqlm="select code,nom,prenom from membre where code=".$code." and
motpass=".$mdp."";
    $res=$conn->query($req)->fetchAll(PDO::FETCH_OBJECT) ;
    if ( count($res) == 0 )
    { header('location:index.php');}
    else
    {
        $prenom=$res[0]->prenom;
        $nom=$res[0]->nom;
        $_SESSION['code']=$code;
        $_SESSION['nom']=$nom;
        $_SESSION['prenom']=$prenom;
        header('location:lstpublications.php');
    }
?>

```

3) Le script « lstpublications.php » est le suivant : (3 pts)**Méthode 1 : Classe mysqli**

```

<html>
<head> </head>
<body>
<?php
session_start() ;
require 'connecter.php'; ?>
<h3> Publications de
<?php //[1]
echo $_SESSION['nom']. " ". $_SESSION['prenom']; ?>
<table>
    <thead>
<tr>
    <th>Numéro</th> <th>Message</th> <th>Nom du fichier</th> <th> </th>
</tr>
    </thead>
    <tbody>
        <?php //[2]
$code=$_SESSION['code'];
$sqlp="select * from publication where codem=".$code;
$resultp = $conn->query($sqlp);
if ($resultp->num_rows > 0) {
    while ($row = $resultp->fetch_assoc())
    {
        $num=$row["num"];
        $message=$row["message"];
        $nomfich=$row["nomfich"];

echo"<tr><td>". $num. "</td><td>".$message."</td><td>".$nomfich."</td>
<td><a href='modifpubl.php?nump=$num'>Modification</a></td></tr>";
    } }
?>
    </tbody></table></body></html>

```

Méthode 2 : Classe PDO

```
<html>
<head> </head>
<body>
  <?php
    session_start() ;
    require 'connecter.php'; ?>
  <h3> Publications de
  //[1]
  <?= $_SESSION['nom'] ; ?> &nbsp; ; <?= $_SESSION['prenom']; ?>
  <table>
    <thead>
      <tr>
        <th>Numéro</th> <th>Message</th> <th>Nom du fichier</th> <th> </th>
      </tr>
    </thead>
    <tbody>
      <?php //[2]
        $code=$_SESSION['code'];
        $sqlp="select * from publication where codem=".$code;
        $res=$conn->query($sqlp)->fetchAll(PDO::FETCH_OBJECT) ;
        if (count($res) > 0) {
          foreach($res as $p) {
            $num=$p->num;
            $message=$p->message;
            $nomfich=$p->nomfich;
            echo"<tr><td>". $num. "</td><td>".$message."</td><td>".$nomfich."</td>
            <td><a href='modifpubl.php?nump=$num'>Modification</a></td></tr>";
          } }
        ?>
    </tbody></table></body></html>
```