

#### d. Eléments de corrections (DAI) :

## GESTION DES PROJETS

### A) Réalisation des classes :

1- Réaliser la classe Tache qui est définie comme suit :

```
public classe Tache {
private String CodeTache ;
private String nomTache ;
private Date DateDebut ;
private Date DateFin ;
a-
public Tache(String CodeTache, String nomTache, Date DateDebut, Date
DateF)
{
this.CodeTache=CodeTache;
this.nomTache=nomTache;
this.DateDebut=DateDebut;
this.DateFin=DateFin;
}
b-
public void setDateDebut(Date d)
{ this.DateDebut=d;}

public void setDateFin(Date d)
{ this.DateFin=d;}
c-
@Override
public String toString( )
{return "Code : "+codeTache+", la tache intitulée :"+ nomTache+", Date
de début : "+dateDebut+", Date de fin : +dateFin ;
}
d-
@Override
public boolean equals(Object Obj )
{Tache p=(Tache)Obj ;
return this.codeTache.equals(p.codeTache) ;}
}
```

## 2- Réaliser les classes dérivées de la classe Tache :

```
public class TacheOrdinaire extends Tache {
private String Description;

public TacheOrdinaire(String CodeTache, String nomTache, Date DateDebut, Date
DateF,String Description)
{
Super(CodeTache, nomTache, DateDebut, DateF);
this.Description=Description;
}
@Override
public String toString( )
{return super.toString()+" , Description :"+this.Description ;
}
}
```

### c- Créer la classe TacheCritique qui doit contenir

- Un constructeur avec arguments
- La méthode toString( ) qui doit retourner une chaîne décrite comme suit :

**Code : xxx , la tache intitulée : xxx , Date de début : xxx, Date de fin : xxx, DelaiMax :xxx**

```
public classe TacheCritiqueextends Tache {
private intDelaiMax;

public TacheOrdinaire(String CodeTache, String nomTache, Date
DateDebut, Date DateF,intDelaiMax)
{
Super(CodeTache, nomTache, DateDebut, DateF);
this.DelaiMax=DelaiMax;
}
@Override
public String toString( )
{return super.toString()+" , DelaiMax :"+this.DelaiMax;}}
```

## 3-

- ### a- Cette classe contient une collection pour traduire le lien logique : pour chaque projet on a plusieurs taches.

```
public classe Projet {
private String Code ;
private String nomProjet ;
privateVector<Tache>Liste=new Vector<Tache>( ) ;

b-
public Projet(String Code, String nomProjet)
{this.Code=Code ; this.nomProjet=nomProjet ;}

c-
```

```

public voidAjouter_Tache(Tache d)
{Liste.add(d);}

    d-
public void Supprimer_Tache(intindex)
{
if((index>=0)&&(index<Liste.size()))
Liste.remove(index);}

    e-
@Override
public String toString( )
{String a = "Code :"+ Code+" , Nom projet :"+nomProjet+" , La liste
des taches de ce projet:\n" ;
for(int i=0 ; i<Liste.size() ;i++)
a += Liste.get(i).toString()+"\n";
return a;
}

    f-
@Override
public boolean equals(Object Obj )
{Projet p=(Projet)Obj ;
return this.code.equals(p.code) ;}
}

```

#### 4- Test des classes :

```

Projet P=new Projet("P1","Gestion d'absences") ;
Tache T[ ] = new Tache[2] ;
T[0] = new TacheOrdinaire("t1","conception",new Date("14/02/2014"),
new Date("24/02/2014"), "Analyse") ;
T[1] = new TacheCritique("t3","test unitaire",new Date("12/03/2014"),
new Date("15/03/2014"), 7) ;
For (int i=0 ; i < T.length ; i++) System.out.println(T[i]);
For (int i=0 ; i < T.length ; i++) P.Ajouter_Tache(T[i]);
System.out.println(P);

```

#### B) Persistance des données

L'IHM est illustrée par la figure suivante :

codeTache	nomTache	dateDebut	dateFin	code
T1	analyse	02/12/2013	16/12/2013	P1
T2	Création de la base	17/12/2013	23/12/2013	P1
T3	analyse	16/12/2013	23/12/2013	P2
T4	analyse	16/12/2013	23/12/2013	P2

Figure 5 : IHM de gestion des taches des projets

Le code de cette classe Form est le suivant :

```
Imports System.Data.SqlClient
PublicClass Form1
1-
Private cn As SqlConnection
Private dst AsNew DataSet
Private adptProjets As SqlDataAdapter
Private adptTaches As SqlDataAdapter
2-
Sub connecter()
Try
cn = New SqlConnection("Data Source=SERVER\SQLEXPRESS;Integrated
Security=true;Initial Catalog=Base")
cn.Open()
Catch ex As Exception
MessageBox.Show("Erreur de connexion"+ex.Message)
End
EndTry
EndSub
3-
Sub chargerDataSet()
Dim cmdp, cmdt As SqlCommand
cmdp = New SqlCommand("select * from projet", cn)
cmdt = New SqlCommand("select * from tache order by code,codetache",
cn)

adptProjets = New SqlDataAdapter(cmdp)
adptTaches = New SqlDataAdapter(cmdt)
adptProjets.Fill(dst, "projet")
adptTaches.Fill(dst, "tache")
EndSub
```

4-

```
Sub listerProjets()  
Me.cbProjets.DisplayMember = "nomprojet"  
Me.cbProjets.ValueMember = "code"  
Me.cbProjets.DataSource = dst.Tables("projet")  
EndSub
```

5-

```
Sub listerTaches()  
Me.DGV.DataSource = dst.Tables("tache")  
EndSub
```

```
PrivateSub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load  
    connecter()  
    chargerDataSet()  
    listerProjets()  
    listerTaches()  
EndSub
```

6-

```
Function CodeTache_existe(Byval code As String) As Boolean  
Dim L() As DataRow  
    L = dst.Tables("tache").Select("codetache='" + code + "'")  
If L.Length <> 0 Then  
    Return True  
EndIf  
Return False  
End Function
```

7-

```
PrivateSub Ajouter_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Ajouter.Click  
If cbProjets.Text.Trim.Equals("") OrMe.codetache.Text.Equals("") Then  
    MessageBox.Show("Formulaire incomplet")  
Exit Sub  
EndIf
```

```
If Me.debut.value.compareto(Me.fin.value) < 0 then  
MessageBox.Show("Dates invalides")  
ExitSub  
EndIf
```

```
If CodeTache_existe(Me.codeTache.text) then  
MessageBox.Show("Code éxistant")  
Exit Sub  
EndIf
```

```
Dim A As DataRow  
    A = dst.Tables("tache").NewRow  
    A(0) = Me.codetache.Text  
    A(1) = Me.NomTache.Text  
    A(2) = Me.debut.Value  
    A(3) = Me.fin.Value  
    A(4) = Me.cbProjets.SelectedValue
```

```

dst.Tables ("tache").Rows.Add (A)

Dim bd As New SqlCommandBuilder (adptTaches)
adptTaches.Update (dst, "tache")
dst.Clear ()
chargerDataSet ()
listerProjets ()
listerTaches ()

EndSub
8-
PrivateSub Supprimer_Click (ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Supprimer.Click
Dim pos As Integer = DGV.CurrentRow.Index
If pos = -1 Then Exit Sub
dst.Tables ("tache").Rows (pos).Delete ()
Dim bd AsNew SqlCommandBuilder (adptTaches)
adptTaches.Update (dst, "tache")
dst.Clear ()
chargerDataSet ()
listerProjets ()
listerTaches ()

EndSub
EndClass

```

### C) Consultation en ligne : (8 pts)

## Gestion des projets

Projet

Gestion de stock ▼  
 Gestion de stock  
 Gestion de notes  
 Gestion d'emploi de temps

Afficher details

### Liste des taches du projet : Gestion de stock

CodeTache	nomTache	Date Début	Date Fin	Supprimer
T1	Analyse	2014/02/01	2014/03/01	<a href="#" style="color: purple; text-decoration: underline;">sup</a>
T2	Base de données	2014/03/02	2014/04/01	<a href="#" style="color: purple; text-decoration: underline;">sup</a>

Figure 4 : Portail de consultation en ligne

1-

```

<html>
<head>
</head>
<body>
  <h1 align="center"> Gestion des projets</h1>

```

```

        <br>
        <table align="center" width="60%" border="2">
        <form>
        <tr><td width="50%">Projet</td><td><select
name="ListeProjets">

<?php
    $c=mysql_connect("localhost","root","") or die("Erreur de
connexion au serveur");
    mysql_select_db("Base",$c) or die("Erreur de connexion au
serveur");
    $rs=mysql_query("select * from projet");
    while($row=mysql_fetch_row($rs))
    echo "<option value='$row[0] '$row[1]</option>";

?>

</select>
</td></tr>
<tr><td colspan="2"><input type="submit" value ="  Afficher
details  " name="Afficher"></td></tr>
</form>
</table>

```

## 2-

```

<hr>
<h2 align="center">Liste des taches du projet : Gestion de
stock</h2>
<table align="center" width="60%" border="2">
<tr><th width="20%">CodeTache</th><th width="20%">nomTache</th><th
width="20%">Date Début</th><th width="20%">Date
Fin</th><th>Supprimer</th></tr>
<?php
    if(isset($_POST) and !empty($_POST))
    {
        $c=mysql_connect("localhost","root","") or die("Erreur
de connexion au serveur");
        mysql_select_db("Base",$c) or die("Erreur de connexion
au serveur");
        $a = $_POST["ListeProjets"];
        $rs=mysql_query("select * from Tache where code =
'$a'");
        while($row=mysql_fetch_row($rs))
            echo
            "<tr><td>$row[0]</td><td>$row[1]</td><td>$row[2]</td><td>$row
[3]</td><th><a href='#?n=\"$row[0]\"'>sup</a></th></option>";
    }

```

```

?>
</table>
3-
<?php
    if(isset($_GET['n']) and !empty($_GET['n']))
    {
        $c=mysql_connect("localhost","root","") or die("Erreur
de connexion au serveur");
        mysql_select_db("Base",$c) or die("Erreur de connexion
au serveur");
        $a=$_GET['n'];
        mysql_query("delete from Tache where codeTache='$a'");
        header("location:#");
    }
?>
</body>
</html>

```

#### D) Communication Intranet : (8 pts)

##### 1- Questions de cours :

- a- Modèle de Gartner Group
- b- Serveur lourd
- c- Statement : préparation de la requête  
ResultSet : Résultat de la requête.

##### 2- Programme Client :

```

public class Client {
private Socket sc=null;
private DataInputStream in = null;
private PrintStream out=null;

    a-
public boolean SocketConnexion (String IP, int Port )
{Try
{sc = new Socket (IP,Port);
in = new DataInputStream(new
BufferInputStream(sc.getInputStream()));
    out = new PrintStream(new
BufferedOutputStream(sc.getOutputStream()));
return true;}
catch (Exception e) {return False;}
}

    b-
public Object demandeTaches(String codeProjet)
{out.println(codeProjet) ;
return in.readLine ();}
}

```



```
}
```

### 3- Programme Serveur :

```
public class Serveur {
private ServerSocket socketserver
private Socket sc=null;
private DataInputStream in = null;
private PrintStream out= null;

    a-
public boolean acceptConnexion (int Port)
{
Try {
socketserver = new ServerSocket(Port) ;
sc = socketserver.accept() ;
in = new DataInputStream(new
BufferInputStream(sc.getInputStream()));
out = new PrintStream(new
BufferedOutputStream(sc.getOutputStream()));
return true;
} catch ( Exception e) { return false;}
}

    b-
public void EnvoiTaches()
{String code ;
If ((code = in.readLine()) != null)    { int pos = PFE.indexOf(new
Projet(code,null));
If (pos == -1) out.println("Projet introuvable");
Else out.println(PFE.get(pos));
}
```

### c- Programme Serveur Multi-client:

```
public class Serveur implements Runnable{
private Socket sc=null;
private DataInputStream in = null;
private PrintStream out= null;
Public Serveur (Socket S){
sc = S ;}

    public void run( ) {
if (acceptConnexion()) EnvoiTache( ) ;
else System.out.println(''Erreur de connexion'' ) ;
}

public boolean acceptConnexion ()
{
```

```

Try {
in = new DataInputStream(new
BufferInputStream(sc.getInputStream()));
out = new PrintStream(new
BufferedOutputStream(sc.getOutputStream()));
return true;
} catch ( Exception e) { return false;}

}

public void EnvoiTaches()
{String code ;
If ((code = in.readLine()) != null)    { int pos = PFE.indexOf(new
Projet(code,null));
If (pos == -1) out.println("Projet introuvable");
Else out.println(PFE.get(pos));
}
}

Public class Serveurecouteur{
ServerSocket SS = null;
Public Serveurecouteur(int port) {
Try { SS = new ServerSocket (port);
While (true)
{ Serveur A = new Serveur (SS.accept());
Thread t = new Thread(A);
t.start( );
}
} catch ( Exception e) {System.out.println('' Errur'');} }}

```