

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {
    .....
}
```

- 5- La méthode événementielle du bouton « Go » : Permet d'exécuter la méthode « **vehDisponible** » sur le serveur. La valeur retournée (**ArrayList**) doit être affichée dans la liste « **Liste** ». (0,5 pt)

```
private void GoActionPerformed(java.awt.event.ActionEvent evt) {
    .....
}
```

IV. Éléments de correction de DAI :

DOSSIER I : ETUDE DE L'APPLICATION DESKTOP JAVA

(13 PTS)

- 5- Classe « **Vehicule** » :

(3 pts)

```
public class Vehicule{
    private int idVeh ;
    private String Matricule ;
    private Date DateAchat;

    public Vehicule(int idVeh, String Matricule, Date DateAchat){ (0,5 pt)
        this.idVeh= idVeh; this.Matricule= Matricule;
        this.DateAchat= DateAchat;
    }

    public Vehicule(int idVeh, String Matricule){ (0,5 pt)
        this.idVeh= idVeh; this.Matricule= Matricule;
        this.DateAchat= new Date();
    }

    @Override
    public String toString(){ (1 pt)
        return "ID de véhicule: "+ this.idVeh +", Matricule: "
            + this.Matricule + ", Acheté le : " + this.DateAchat;
    }

    @Override
    public boolean equals(Object ob){ (1 pt)
        Vehicule c=( Vehicule)ob;
        return this.idVeh==c.idVeh;
    }
}
```

- 6- Classe « **camion** » : (2 pts)

```
public class Camion extends Vehicule{
    private double Charge ;
    private double Tare ;

    public Camion(int idVeh, String Matricule, Date DateAchat, double Charge, double Tare) {
        super(idVeh, Matricule, DateAchat) ;
    }
}
```

```

        this.Charge= Charge ;
        this.Tare= Tare;
    }

    public Camion(int idVeh, String Matricule, double Charge, double Tare) {
        super(idVeh,Matricule) ;
        this.Charge= Charge ;
        this.Tare= Tare;
    }

    @Override
    public String toString(){
        return super.toString()+ "\n Tare :"+ this.Tare+
            " et Charge :"+ this.Charge ;
    }
}

```

7- Classe « Garage » :

(5,5 pts)

```

public class Garage {
    private int id ;
    private String Libelle ;
    private long Longitude, Lattitude;
    private ArrayList<Vehicule> Liste;

    public Garage (int id, String Libelle, long Longitude, long Lattitude) {
        this.id id; this.Libelle= Libelle; this.Longitude= Longitude;
        this.Lattitude= Lattitude;
        Liste= new ArrayList<Vehicule>();
    }

    public boolean addVehicule(Vehicule V) {
        if(this.searchVehicule(V)==-1) return false;
        return Liste.add(V);
    }

    public Vehicule delVehicule(int index){
        If((index>=0) && (index<Liste.size()))
        return Liste.remove(index);
        return null;
    }

    public Vehicule setVehicule(Vehicule nouveau, int index){
        If((index>=0) && (index<Liste.size()))
        return Liste.set(index,nouveau);
        return null;
    }

    public int searchVehicule(Vehicule V){

```

```

        return Liste.IndexOf(V) ;
    }

    @Override
    public String toString(){
        String r="";
        for(Vehicule a:Liste) r += a.toString()+"\n";
        return r;
    }
}

```

8- Test des classes :

(2,5 pts)

Créer un programme de test permettant d'effectuer les étapes suivantes :

```

Camion C1=new Camion(1, "33552 A 1",new Date(),5000,100000);
Camion C2 =new Camion(2, "56782B 1",new Date(),2000,90000);

Garage G = new Garage(10,"Garage1",100,10) ;
If(!G.addVehicule(C1) || !G.addVehicule(C2)){
System.err.println("Un échec d'ajout est détecté") ;
}
System.out.println(G) ;
System.out.println("Camion "+(G.searchVehicule(c1)!=-1)?"Existe":
"inexistant");
G.delVehicule(2) ;
System.out.println(G) ;

```

DOSSIER II : ETUDE DE L'APPLICATION DESKTOP SOUS VB.NET

(11 PTS)

1- Les objets de gestion de cette base de données

(1,5 pt)

```

Public cn As SqlConnection
Public dst AsNew DataSet
Public adptclient As SqlDataAdapter
Public adptgarage As SqlDataAdapter
Public adptveh As SqlDataAdapter
Public adptvehs As SqlDataAdapter

Initialiser tous ces objets

Sub connecter()
Try
    cn = New SqlConnection("initial catalog=BD_Demandes; data source=PC_Demande;
Integrated Security=true;")
    cn.Open()
Catch ex As Exception
    MessageBox.Show(ex.Message)
EndTry
EndSub

Sub chargerdataset()
    adptclient = New SqlDataAdapter("select * from client", cn)
    adptgarage = New SqlDataAdapter("select * from garage", cn)
    adptveh = New SqlDataAdapter("select * from vehicule", cn)
    adptvehs = New SqlDataAdapter("select idveh,Matricule,Type,Libelle,NomPrenom
from vehicule v,client c,garage g where v.idclient=c.idclient and v.id=g.id", cn)

```

```

        adptclient.Fill(dst, "client")
        adptgarage.Fill(dst, "garage")
        adptveh.Fill(dst, "vehicule")
        adptvehs.Fill(dst, "vehicules")
EndSub

```

2- Les procédures :

(3 pts)

```

Sub chargerclients()
    cbClients.DisplayMember = "nomprenom"
    cbClients.ValueMember = "idclient"
    cbClients.DataSource = dst.Tables("client")
EndSub
Sub chargerGarages()
    cbGarages.DisplayMember = "libelle"
    cbGarages.ValueMember = "id"
    cbGarages.DataSource = dst.Tables("garage")
EndSub
Sub chargerTypes()
    cbTypes.Items.Clear()
    cbTypes.Items.Add("Camion")
    cbTypes.Items.Add("Voiture")
    cbTypes.Items.Add("Moto")
EndSub
Sub lister()
    table.DataSource = dst.Tables("vehicules")
    table.Columns(0).Width = table.Width / 10
EndSub

```

3- Le code des procédures évènementielles des 3 boutons :

(1 pt)

```

PrivateFunction formValide() AsBoolean
    IfMe.txtMat.Text.Trim.Equals("") OrMe.cbTypes.SelectedIndex = -1 Then
        ReturnFalse
    EndIf
    Try
        Integer.Parse(Me.txtID.Text)
    Catch ex As Exception
        ReturnFalse
    EndTry
    ReturnTrue
EndFunction

```

4-

(1 pt)

```

PrivateFunction cleExiste() AsBoolean
Dim t() As DataRow = dst.Tables("vehicule").Select("idveh='" + txtID.Text + "'")
    Return t.Length <> 0
EndFunction

```

5-

(3 pts)

```

PrivateSub cmdEnregistrer_Click(.....) Handles cmdEnregistrer.Click
    IfNotMe.formValide Then
        MessageBox.Show("Formulaire incomplet")
    ExitSub
    EndIf
    IfMe.cleExiste Then
        MessageBox.Show("clé déjà existante")
    ExitSub
    EndIf

    Dim t As DataRow = dst.Tables("vehicule").NewRow

```

```

t(0) = Integer.Parse(Me.txtID.Text)
t(1) = txtMat.Text
t(2) = Me.dateAchat.Value
t(3) = Me.cbTypes.Text
t(4) = Me.cbGarages.SelectedValue
t(5) = Me.cbClients.SelectedValue
dst.Tables("vehicule").Rows.Add(t)
Dim c As New SqlCommandBuilder(adptveh)
adptveh.Update(dst, "vehicule")
dst.Clear()
chargerdataset()
chargerclients()
chargerGarages()
chargerTypes()
lister()
EndSub

```

6-

(1,5 pt)

```

PrivateSub cmdSupprimer_Click(.....) Handles cmdSupprimer.Click
    Dim p As Integer = table.CurrentRow.Index
    If p = -1 Then ExitSub
    dst.Tables("vehicule").Rows(p).Delete()
    Dim c As New SqlCommandBuilder(adptveh)
    adptveh.Update(dst, "vehicule")
    dst.Clear()
    chargerdataset()
    chargerclients()
    chargerGarages()
    chargerTypes()
    lister()
EndSub

```

7-

(1,5 pt)

```

PrivateSub cmdModifier_Click(.....) Handles cmdModifier.Click
    IfNotMe.formValide Then
        MessageBox.Show("Formulaire incomplet")
        ExitSub
    EndIf
    IfNotMe.cleExiste Then
        MessageBox.Show("clé inexistante")
        ExitSub
    EndIf
    Dim a() As DataRow = dst.Tables("vehicule").Select("idveh='" + txtID.Text +
        "'")
    Dim t As DataRow = a(0) : t(1) = txtMat.Text
    t(2) = Me.dateAchat.Value : t(3) = Me.cbTypes.Text
    t(4) = Me.cbGarages.SelectedValue : t(5) = Me.cbClients.SelectedValue
    Dim c As New SqlCommandBuilder(adptveh)
    adptveh.Update(dst, "vehicule")
    dst.Clear()
    chargerdataset()
    chargerclients()
    chargerGarages()
    chargerTypes()
    lister()
EndSub

```

DOSSIER III : ETUDE DE L'INTERFACE CLIENT WEB

(8 PTS)

1- Le corps de la fonction « test »

(2pts)

```
<script type="text/javascript">
function test(){
x=document.f.id.value;
if(x.trim()==''){
alert('votre id est non spécifier');
return false; }
return true;
}
</script>
```

2- ,3 et 4 Script de connexion à la base de données.

(5 pts)

```
<?php
if(isset($_POST["id"]))
{
$a=$_POST["id"];
$c=mysqli_connect("serv_com","root",""," BD_Demandes ") or die("erreur");
$r=mysqli_query($c,"select Matricule,dateachat,type,libelle from vehicule
v,garage g where v.id=g.id and v.idclient=$a");
while($l=mysqli_fetch_row($r))
echo "<tr><td>$l[0]</td><td>$l[1]</td><td>$l[2]</td><td>$l[3]</td></tr>";
}
?>
```

DOSSIER IV : ETUDE DE L'APPLICATION DISTRIBUEE

(9 PTS)

1. PROGRAMME COTÉ SERVEUR :

b. :

```
publicinterface intObjetDistant extends Remote {
publicArrayList<Vehicule>vehDisponible() throws RemoteException;
publicArrayList<Document>invalidDocs() throws RemoteException;
}
```

c. :

```
publicclassObjetDistantextends UnicastRemoteObject implementsintObjetDistant {
private Connection conn;
private PreparedStatement ps;
private ResultSet rs;

//constructeur
publicObjetDistant() throws RemoteException {
super();

alDoc = new ArrayList<Document>();
try {
conn = DriverManager.getConnection
("jdbc:sqlserver://PC_Demande:1433; database = BD_Demandes ;
```

```

IntegratedSecurity=true;", "root", "" );
} catch (Exception ex) {
System.out.println("Erreur "+ex.toString());
}
}

//Implémentation des méthodes de l'objet distant
public ArrayList<Vehicule>vehDisponible() throws RemoteException{
ArrayList<Vehicule>alVh = new ArrayList<Vehicule>();
try {
ps = conn.prepareStatement("select * from Vehicule");
rs = ps.executeQuery();
while(rs.next()){
Vehicule v = new Vehicule(rs.getInt(1), rs.getString(2),
rs.getDate(3));
alVh.add(v);
}
} catch (SQLException ex) {
System.out.println("Erreur : "+ex.toString());
}
return alVh;
}
public ArrayList<Document>invalidDocs() throws RemoteException{
ArrayList<Document>alDoc = new ArrayList<Document>();
try {
ps = conn.prepareStatement("select * from Document");
rs = ps.executeQuery();
Date dl = new Date(Date().getTime()+7*24+3600*1000);
while(rs.next()){
if(dl.after(rs.getDate(4))){
Document doc = new Document(rs.getInt(1), rs.getString(2),
rs.getString(3), rs.getDate(4));
alDoc.add(v);
}
}
} catch (SQLException ex) {
System.out.println("Erreur : "+ex.toString());
}
return alDoc;
}
}

```

d. :

Ligne 3 : Lancer l'annuaire de noms rmiregistry sur le port standard 1099.
 Ligne 4 : Créer une instance de l'objet distant (le skeleton).
 Ligne 5 : Publier le nom de l'objet distant (le skeleton) sur l'annuaire de nom sous le nom NOBJ.

2. PROGRAMME COTÉ CLIENT :

a)

```

private void formWindowOpened(java.awt.event.WindowEvent evt) {
Obj = (intObjetDistant) Naming.lookup("rmi://localhost:1099/NOBJ");
}

```

b)

```
private void GoActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        ArrayList<VehDisponible> vhd = Obj.vehDisponible();  
        Liste.setListData(vhd.toArray());  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
}
```