

Cet examen contient 6 pages.

Les Documents, calculatrices, téléphone portable sont interdits. Veuillez rendre une copie propre et claire. La qualité de l'écriture et de la présentation sera prise en compte dans la note finale.

Questions de réflexion : (3 points)

1. Nous souhaitons avoir un conteneur intermédiaire qui construit une interface divisée en 2 parties. Quel est le conteneur qui permet de le faire ?
2. A quel package appartient la classe DriverManager ?
3. l'utilisation de la méthode getConnection() du DriverManager est susceptible de lancer une exception mais de quel type ? cette exception est elle obligatoire à attraper ou pas (checked ou unchecked) ?



-
1. Le conteneur qui permet d'avoir une interface divisé en 2 parties est le JSplitPane **(1pt)**
 2. java.sql **(1pt)**
 3. SQLException **(0.5pt)** Checked (obligatoire) **(0.5pt)**
-



► **Exercice 1. Exception (4 points)**

Que produit le programme suivant lorsqu'on lui fournit en donnée :

1. la valeur 2,
2. aucune valeur et on clique sur ok

```
class Except extends Exception
{ public Except (int x) { n = x ; }
public Except() {System.out.println("attention");}
public static int n=5 ;
}
public class Chemin
{ public static void main (String args[])
{ int n=1; ;
String retour=JOptionPane.showInputDialog(null, "entrez une valeur");
try
{ n=Integer.parseInt(retour);
System.out.println ("debut premier bloc try") ;
if (n!=0) throw new Except (n) ;
System.out.println ("fin premier bloc try") ;
}
catch (NumberFormatException e) { System.out.println("valeur non numérique"); }
catch (Except e){ System.out.println ("catch 1 - n = " + Except.n);}
System.out.println ("suite du programme") ;
try
{
System.out.println ("debut second bloc try") ;
if (n==1) throw new Except () ;
System.out.println ("fin second bloc try") ;
}
catch (Except e){System.out.println ("catch 2 - n = " + Except.n) ;}
finally {System.out.println("bye");}
System.out.println ("fin programme") ;
}}
```

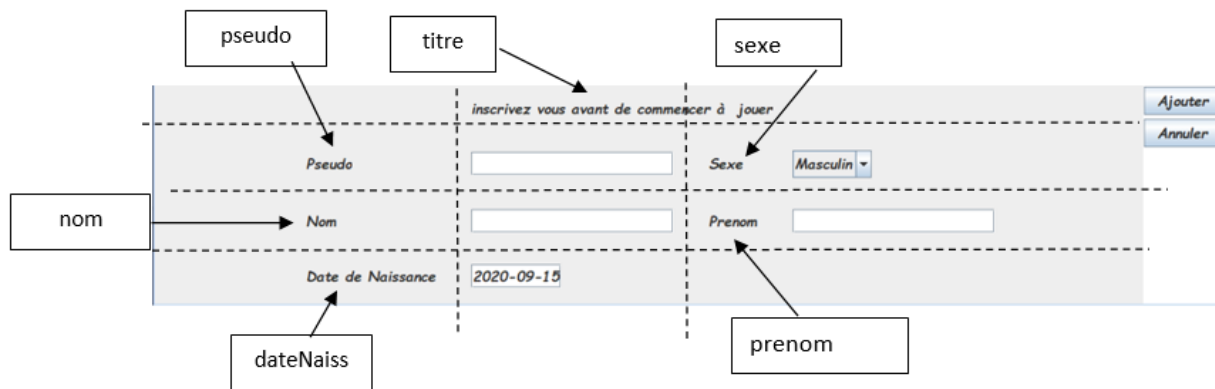


1. debut premier bloc try(0.25pt)
catch 1 - n = 2(0.25pt)
suite du programme(0.25pt)
debut second bloc try(0.25pt)
fin second bloc try(0.25pt)
bye(0.5pt)
fin programme(0.25pt)
2. valeur non numérique(0.25pt)
suite du programme(0.25pt)
debut second bloc try(0.25pt)
attention(0.5pt)
catch 2 - n = 5(0.25pt)
bye(0.25pt)
fin programme(0.25pt)



► Exercice 2. Swing (13 Points)

Nous revenons à notre application du projet jeu de devinette. La zone centre et est de l'onglet **Inscription** et **mise à jour** a l'aspect suivant :



Nous avons défini une classe Vue comme suit.

1. Complétez le constructeur de la classe Vue suivante :

```
public class Vue extends JPanel{
    protected JPanel pC, pE;
    protected JButton ajouter, annuler;
    protected JTextField txtPs,txtNom,txtPrenom;
    protected JComboBox<String> txtSexe;
    protected JFormattedTextField txtdateNaiss;
    protected JLabel pseudo, nom, prenom, sexe,dateNaiss, titre;
    Date dd = new Date();
    SimpleDateFormat formater = new SimpleDateFormat("yyyy-MM-dd");
    protected Connection conn;
    protected Statement st;
    protected ResultSet rs;

    Vue(){
        setLayout(new BorderLayout());
        pC=new JPanel();
        pE=new JPanel();
    }
}
```

```

pE.setBackground( Color . white );
add(pC);          add(pE, " East" );
ajouter=new JButton( " Ajouter" );
annuler=new JButton( " Annuler" );
pE.add(ajouter); pE.add(annuler);
charger();

String [] donnees={" Masculin" , " Feminin" };
titre=new JLabel( " Inscrivez vous avant de commencer à jouer" );
pseudo=new JLabel( " Pseudo" );
txtPs = new JTextField( "" ,18);
nom=new JLabel( " Nom" );
txtNom = new JTextField( "" ,18);
prenom=new JLabel( " Prenom" );
txtPrenom = new JTextField( "" ,18);
sexe=new JLabel( " Sexe" );
txtSexe= new JComboBox<String>(donnees);
dateNaiss=new JLabel( " Date de Naissance" );
txtdateNaiss = new JFormattedTextField( formater . format( dd ));

pC.setLayout( new GridBagLayout () );
GridBagConstraints gbc= new GridBagConstraints ();
gbc.insets = new Insets(20, 25, 20, 20);
//Ajoutez les composants au panel centre :pC
//...
}
}

```



7pts

```

//7points
gbc.gridx = 1;//0.25pt
gbc.gridy = 0;//0.25pt
gbc.gridwidth=GridBagConstraints.REMAINDER;//0.25pt
pC.add(titre , gbc);//0.25pt

gbc.gridwidth=1;//0.25pt
gbc.gridx = 0;//0.25pt
gbc.gridy = 1;//0.25pt
pC.add(pseudo , gbc);//0.25pt

gbc.gridx = 1;//0.25pt
pC.add(txtPs ,gbc); //0.25pt

gbc.gridx = 2;//0.25pt
pC.add(sexe , gbc);//0.25pt

gbc.gridx = 3;//0.25pt
pC.add(txtSexe , gbc);//0.25pt

gbc.gridx = 0;//0.25pt
gbc.gridy = 2;//0.25pt
pC.add(nom ,gbc);//0.25pt

gbc.gridx = 1;//0.25pt
pC.add(txtNom , gbc);//0.25pt

gbc.gridx = 2;//0.25pt

```

```
pC.add(prenom, gbc); //0.25 pt

gbc.gridx = 3; //0.25 pt
pC.add(txtPrenom, gbc); //0.25 pt

gbc.gridx = 0; //0.25 pt
gbc.gridy = 3; //0.25 pt
pC.add(dateNaiss, gbc); //0.25 pt

gbc.gridx = 1; //0.25 pt
pC.add(txtdateNaiss, gbc); //0.25 pt
```

2. Nous devons juste traiter les événements de type Action déclenchés par le bouton ajouter de la zone est. Rappelons que votre base de données (nommée db_jeu) contient une table joueur :

Colonne	Type
pseudo (primary key)	varchar(10)
nom	varchar(20)
prenom	varchar(20)
datnais	date
sexe	varchar(10)
nbressai	int

Comme vous l'avez remarqué la classe **Vue** possède un attribut de type Connection (**conn**) un attribut de type ResultSet (**rs** qui contient tous les joueurs) et une méthode charger() appelée au niveau du constructeur et définie comme suit :

```
public void charger(){
    try {
        conn = DriverManager.getConnection("jdbc:mysql://localhost/db_jeu","root","");
        Statement st = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
            ResultSet.CONCUR_UPDATABLE);
        String query = "SELECT * FROM joueur";
        rs = st.executeQuery(query);
        JOptionPane.showMessageDialog(null,"connexions effectuée!");
    } catch (SQLException e){
        JOptionPane.showMessageDialog(null,"problème de connexion"); }
}
```

Nous allons définir juste la méthode du bouton de clique sur ajouter.

Le clique sur ajouter permet d'ajouter **via le ResultSet** rs un nouveau joueur en récupérant ses informations des différents JTextField et ComboBox et un nbressai initialisé à 8. Si le nom ou prenom ou pseudo du joueur ne sont pas saisis il faudra afficher un message dans un JOptionPane indiquant que les champs nom, prenom et pseudo sont obligatoires. Si au moment de l'insertion, la date de naissance est mal saisie afficher un message dans un JOptionPane indiquant que l'insertion n'a pas pu être effectuée car la date est erronée. Une fois l'insertion effectuée mettre tous les champs de saisie du nom, prenom et pseudo à vide, remettre le ComboBox sur la première sélection et la date de naissance avec la date du jour.

Terminez la méthode actionPerformed du bouton ajouter suivante :

```
ajouter.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        if(!txtPs.getText().isEmpty()&&!txtNom.getText().isEmpty()&&
            !txtPrenom.getText().isEmpty())
        {
            try {
```

```

.....//finir avec l'insertion du tuple via le ResultSet
JOptionPane.showMessageDialog(null, "Insertion effectuée");
} catch (SQLException e) {
    ....//compléter avec le problème de date
}
....//compléter avec le reset des champs
}
else
{....//compléter si un des champs obligatoires est vide
}
}});

```



```

//4 Points
rs.moveToInsertRow(); //0.25 pt
rs.updateString(1, txtPs.getText()); //0.25 pt
rs.updateString(2, txtPrenom.getText()); //0.25 pt
rs.updateString(3, txtNom.getText()); //0.25 pt
rs.updateString(4, txtdateNaiss.getText()); //0.25 pt
rs.updateString(5, (String) txtSexe.getSelectedItem()); //0.25 pt
rs.updateString(6, "8"); //0.25 pt
rs.insertRow(); //0.5 pt
JOptionPane.showMessageDialog(null, "Insertion effectuée");
} catch (SQLException e) {
    JOptionPane.showMessageDialog(null, "problème de date"); //0.25 pt
}
txtPs.setText(""); //0.25 pt
txtNom.setText(""); //0.25 pt
txtPrenom.setText(""); //0.25 pt
txtdateNaiss.setText(formater.format(dd)); //0.25 pt
txtSexe.setSelectedIndex(0); //0.25 pt
}
else
{JOptionPane.showMessageDialog(null, "les champs pseudo,
nom et prenom sont obligatoires!"); //0.25 pt
}

```



3. Soit la table joueur de la base de données db_jeu suivante :

pseudo	nom	prenom	datnais	sexe	nbressai
Deda	Deda	deda	2019-10-16	Feminin	4
Yos	Hannachi	yosra	2018-08-30	Feminin	5
Dragon	Beji	Ali	2018-08-30	Masculin	2
Roi	Ben Saleh	mohamed	2018-08-30	Masculin	4

Que permet de faire la suite d'instructions suivante :

```

PreparedStatement req= bd.prepareStatement
("UPDATE joueur SET sexe = ? WHERE nbressai <=?");
req.setString(1, "Feminin");
req.setString(2, "2");
req.executeUpdate();

```



.....
mettre à jour le sexe des joueurs ayant un nbressai ≤ 2 à féminin. (0,5pt)
.....



4. Finir les instructions suivantes permettant de mettre à jour le nbressai de tous les joueurs en enlevant 1 à toutes les joueuses (Feminin) et ajoutant 1 à tous les joueurs (Masculin).

```
try {  
  
Statement st = conn.createStatement(ResultSet.TYPE_FORWARD_ONLY,  
                                     ResultSet.CONCUR_UPDATABLE);  
ResultSet res = st.executeQuery("select pseudo,sexe, nbressai from joueur");  
//  
    ...  
  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}
```



```
while(res.next()) {  
    if(res.getString(2).equals("Feminin")){  
        res.updateInt(3, res.getInt(3)-1); //0.5 pt  
        res.updateRow(); //0.5 pt  
    }  
    else  
    {  
        res.updateInt(3, res.getInt(3)+1); //0.5 pt  
        res.updateRow(); //0.5 pt  
    }  
}
```



Bon travail