

TP JAVA n°7: Interface Graphique

Exercice 1 : Première Interface

Implémenter dans un nouveau projet **TestGraph**, les fichiers **PremierPanel** et **TestPremiereFenetre** vus en TD et tester.



Ne pas oublier de programmer l'événement de la souris : lorsque l'utilisateur clique sur le panel, le message : « Saisir votre identité et valider » est affiché comme label et la zone de saisie est vidée (on revient ainsi dans l'état initial)

Exercice 2 : Mise en place de l'application graphique CabinetMedical

Dans un premier temps, dans le paquetage **com.iut.cabinet.presentation**, créer une classe héritant de **JFrame** que vous appellerez **CabMedMainFrame.java**. N'écrivez pas de code pour l'instant dans cette classe...

La classe de lancement de l'application graphique se trouvera comme d'habitude dans le paquetage **com.iut.cabinet.essai**. Dans ce paquetage implémenter donc la classe **CabMed_appliGraphique.java** avec le code suivant . (Vous pouvez récupérer ce code sur la zone libre ...)

```
package com.iut.cabinet.essai;
import javax.swing.JFrame;
import com.iut.cabinet.presentation.CabMedMainFrame;

public class CabMed_appliGraphique {

    public static void main(String[] args) {
        CabMedMainFrame fenetre = new CabMedMainFrame();
        fenetre.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        fenetre.setTitle("Cabinet Medical PIC'OUZ");
        fenetre.setSize(500,500);
        fenetre.setVisible(true);
    }
}
```

Lancer cette application, vous devriez obtenir la fenêtre ci-contre .

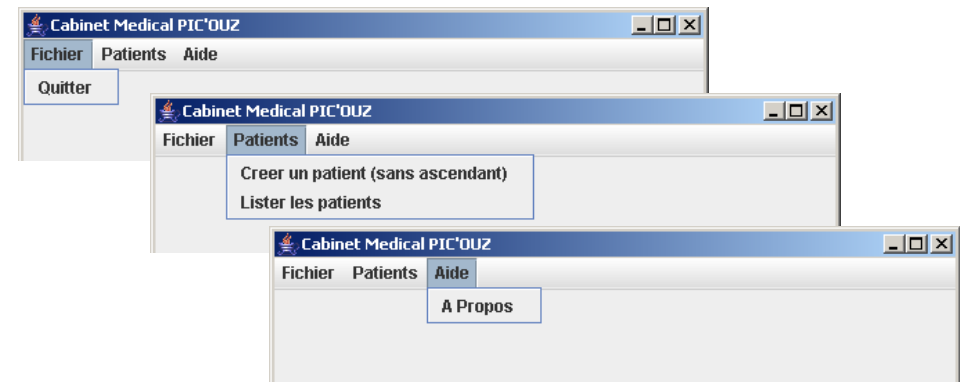
Pour commencer, nous souhaitons **associer un menu à cette fenêtre**.

Nous souhaitons créer directement le menu dans la classe **CabMedMainFrame** sans passer par une sous-classe de **JMenu** (comme exemple **JMenuEditeur** du cours).

→ En vous aidant du cours (transparent n°11) mais en utilisant des composants Swing !!!, écrire dans une méthode privée **initialiserMenus**, le code permettant d'obtenir le menu proposé dans la copie d'écran ci-dessous.

Rappel : En swing, la barre de menu est un objet de type **JMenuBar**. La barre de menu s'ajoute ensuite à la fenêtre en utilisant la méthode **setJMenuBar**.

Remarque : Attention dans le cours, nous avons créé une classe **MenuEditeur** qui héritait de **JMenu**. Ici, nous souhaitons adopter une autre démarche : nous ne créerons pas de sous-classe de **JMenu**, mais nous rajouterons directement les **JMenu**, **JMenuItem** à un objet **JMenuBar** que vous rattacherez à la fenêtre en cours (objet en cours).



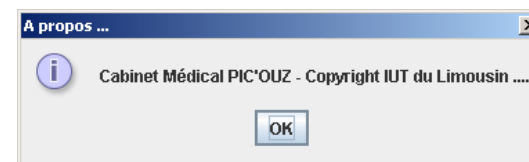
→ Rajouter un constructeur par défaut dans votre classe **CabMedMainFrame** qui appelle la méthode **initialiserMenus**.

→ Tester l'affichage de votre menu en lançant votre application graphique.

Gestion des événements du Menu :

Le menu doit réagir de la manière suivante :

- Lorsque le menu **Quitter** est sélectionné : on quitte le programme **proprement** (c-a-d en utilisant l'instruction suivante : `System.exit(0);`)
- Pour l'instant, on ne gèrera aucun événement sur le menu **Patients**...
- Lorsque le menu **A propos** est sélectionné, une fenêtre s'ouvre en pop-up et propose un message d'information du genre :

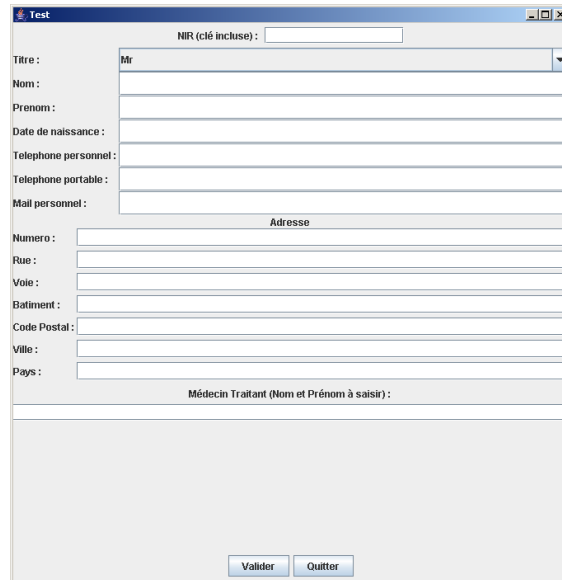


Pour réaliser cette opération, utiliser la classe **JOptionPane** (voir javadoc...)

🔗 Mise en place de PanelCreerPatient :

→ Ecrire ensuite dans le package `com.iut.cabinet.presentation` la classe `PanelCreerPatient` qui permet d'obtenir l'affichage ci-contre (pour l'instant aucune gestion d'événement dans le Panel n'est demandé, on doit juste obtenir cet affichage...).

Remarque : le champs `idPersonne` n'est pas présent dans le formulaire, c'est volontaire



→ Pour tester votre panel, vous ajouterez la méthode `main` suivante à votre classe `PanelCreerPatient`. Vous pourrez ainsi exécuter directement la classe `PanelCreerPatient` pour la phase de développement.

```
public static void main(String args[])
{
    JFrame fenetreTest = new JFrame();
    JPanel mainPanel = new PanelCreerPatient();

    fenetreTest.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    fenetreTest.setTitle("Test");
    fenetreTest.getContentPane().add(mainPanel);
    fenetreTest.setSize(500,500);
    fenetreTest.setVisible(true);
}
```

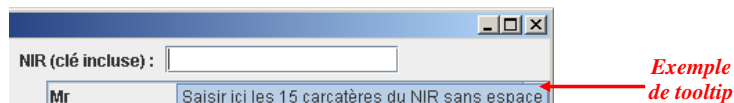
🔗 Ajout de Tooltips (ou bulles d'aide) :

Sur tous les composants Swing, il est possible d'ajouter un *tooltip*.

Un *tooltip* est une bulle d'information qui apparaît lorsque l'utilisateur laisse sans bouger le curseur de la souris au-dessus d'un composant.

Pour ajouter un tooltip, il vous suffit d'utiliser la méthode `setToolTipText` de la classe `JComponent` (aller voir la javadoc).

→ Commencer par rajouter un tooltip sur le NIR comme le montre la copie d'écran ci-dessous...



→ Afin de faciliter la saisie des données du Patient, rajouter autant de tooltips que cela vous semble nécessaire... Notamment pour la date de naissance, indiquer le format à saisir !!!

... Attention : pour la prochaine séance, vous devez avoir terminé ce TP

Correction TP JAVA n°7: Interface Graphique

Exercice 2 : Mise en place de l'application graphique CabinetMedical

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class CabMedMainFrame extends JFrame{

    JMenuItem menuAPropos;
    JMenuItem menuQuitter;
    JMenuItem menuCreerPatient;
    JMenuItem menuListerPatients;

    public CabMedMainFrame()
    {
        InitialiserMenus();

        // Création du Panel principal
        JPanel panelCreerPatient = new PanelCreerPatient();
        getContentPane().add(panelCreerPatient);
    }

    private void InitialiserMenus()
    {
        //Création du Menu
        JMenuBar monMenuCabMed = new JMenuBar();

        JMenu menuFichier = new JMenu("Fichier");
        menuQuitter = new JMenuItem("Quitter");
        menuFichier.add(menuQuitter);
        menuQuitter.addActionListener(new MonMenuActionListener());

        JMenu menuGererPatients = new JMenu("Patients");
        menuCreerPatient = new JMenuItem("Creer un patient (sans ascendant)");
        menuListerPatients = new JMenuItem("Lister les patients");
        menuGererPatients.add(menuCreerPatient);
        menuGererPatients.add(menuListerPatients);
        menuCreerPatient.addActionListener(new MonMenuActionListener());
        menuListerPatients.addActionListener(new MonMenuActionListener());

        JMenu menuAide = new JMenu("Aide");
        menuAPropos = new JMenuItem("A Propos");
        menuAide.add(menuAPropos);
        menuAPropos.addActionListener(new MonMenuActionListener());

        monMenuCabMed.add(menuFichier);
        monMenuCabMed.add(menuGererPatients);
        monMenuCabMed.add(menuAide);
        setJMenuBar(monMenuCabMed); // ajout du menu dans la fenêtre
    }

    //////////////////////////////////////////////////
    // Gestion des événements du menu
    // Classe Interne
    //////////////////////////////////////////////////
```

```
private class MonMenuActionListener implements ActionListener {

    public void actionPerformed(ActionEvent event) {
        if(event.getSource() == menuAPropos)
        {
            JOptionPane.showMessageDialog(null, "Cabinet Médical PIC'OUZ - Copyright IUT du Limousin ...", "A propos ...",JOptionPane.INFORMATION_MESSAGE);
        }

        if(event.getSource() == menuQuitter)
        {
            System.exit(0); // 0 car on quitte normalement
            // par convention status code !=0, si on a quitté "forcé" à cause d'un problème ...
        }
    }
}

package com.iut.cabinet.presentation;

import java.awt.BorderLayout;
import java.awt.GridLayout;

import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class PanelCreerPatient extends JPanel{

    //Tous les composants nécessaires

    // Attributs du patient
    private static final String [] TITRES={"Mr","Mme","Melle"};
    private JComboBox jcb_saisieTitre = new JComboBox (TITRES);
    private JTextField jtx_saisieNom = new JTextField (30);
    private JTextField jtx_saisiePrenom = new JTextField (30);
    private JTextField jtx_saisieTelPerso = new JTextField (10);
    private JTextField jtx_saisieTelPortable = new JTextField (10);
    private JTextField jtx_saisieMailPerso = new JTextField (120);

    private JTextField jtx_saisieNIR = new JTextField (15);
    private JTextField jtx_saisieMedecin = new JTextField (120);

    // Attributs de l'adresse

    private JTextField jtx_saisieAdresseNumero= new JTextField (6);
    private JTextField jtx_saisieAdresseRue = new JTextField (120);
    private JTextField jtx_saisieAdresseVoie = new JTextField (120);
    private JTextField jtx_saisieAdresseBatiment = new JTextField (120);
    private JTextField jtx_saisieAdresseCodePostal = new JTextField (60);
    private JTextField jtx_saisieAdresseVille = new JTextField (120);
    private JTextField jtx_saisieAdressePays = new JTextField (120);

    // Panels intermédiaires nécessaires à la mise en page

    // Boutons de validation
    JButton jb_Valider = new JButton("Valider");
    JButton jb_Quitter = new JButton("Quitter");

    //////////////////////////////////////////////////
    // Constructeur
```

```

////////////////////////////////////
public PanelCreerPatient(){

    ///////////////////////////////////
    // Panel NIR
    ///////////////////////////////////
    JPanel jp_panelNIR = new JPanel(); //flowlayout par défaut ...
    jp_panelNIR.add(new JLabel ("NIR (clé incluse) : "));
    jp_panelNIR.add(this.jtx_saisieNIR);
    jtx_saisieNIR.setToolTipText("Saisir ici les 15 caractères du NIR sans
espace");

    ///////////////////////////////////
    // Panel Caractéristique Personne (hors adresse et descendant)
    ///////////////////////////////////
    JPanel jp_panelPersonne= new JPanel();
    // sera composé de 2 panneaux :
    // => panneau des labels et panneaux des saisies...

    // Creation du panneau des labels : 6 lignes, 1 colonne
    // avec un écart de 5 pixels entre les composants
    // car les infos sont alignées !!!
    JPanel jp_panelLabels = new JPanel(new GridLayout(6,1));
    jp_panelLabels.add(new JLabel ("Titre : "));
    jp_panelLabels.add(new JLabel ("Nom : "));
    jp_panelLabels.add(new JLabel ("Prenom : "));
    jp_panelLabels.add(new JLabel ("Telephone personnel : "));
    jp_panelLabels.add(new JLabel ("Telephone portable : "));
    jp_panelLabels.add(new JLabel ("Mail personnel : "));
    //jp_panelLabels.add(new JLabel ("Adresse : "));

    // Creation du panneau de saisie : 6 lignes, 1 colonne
    // avec un écart de 5 pixels entre les composants
    // Saisie des caractéristiques d'une personne
    JPanel jp_panelSaisiePers = new JPanel(new GridLayout(6,1));
    jp_panelSaisiePers.add(this.jcb_saisieTitre);
    jp_panelSaisiePers.add(this.jtx_saisieNom);
    jp_panelSaisiePers.add(this.jtx_saisiePrenom);
    jp_panelSaisiePers.add(this.jtx_saisieTelPerso);
    jp_panelSaisiePers.add(this.jtx_saisieTelPortable);
    jp_panelSaisiePers.add(this.jtx_saisieMailPerso);

    ///////////////////////////////////
    // Panel Caractéristiques des Adresses
    ///////////////////////////////////
    JPanel jp_panelAdresse= new JPanel();

    // labels des adresse
    JPanel jp_panelAdresse_Labels = new JPanel(new GridLayout(7,1,5,5));
    jp_panelAdresse_Labels.add(new JLabel ("Numero : "));
    jp_panelAdresse_Labels.add(new JLabel ("Rue : "));
    jp_panelAdresse_Labels.add(new JLabel ("Voie : "));
    jp_panelAdresse_Labels.add(new JLabel ("Batiment : "));
    jp_panelAdresse_Labels.add(new JLabel ("Code Postal : "));
    jp_panelAdresse_Labels.add(new JLabel ("Ville : "));
    jp_panelAdresse_Labels.add(new JLabel ("Pays : "));
    // champs de saisie des adresses
    JPanel jp_panelAdresse_SaisiePers = new JPanel(new GridLayout(7,1,5,5));
    jp_panelAdresse_SaisiePers.add(this.jtx_saisieAdresseNumero);
    jp_panelAdresse_SaisiePers.add(this.jtx_saisieAdresseRue);
    jp_panelAdresse_SaisiePers.add(this.jtx_saisieAdresseVoie);
    jp_panelAdresse_SaisiePers.add(this.jtx_saisieAdresseBatiment);
    jp_panelAdresse_SaisiePers.add(this.jtx_saisieAdresseCodePostal);
    jp_panelAdresse_SaisiePers.add(this.jtx_saisieAdresseVille);
    jp_panelAdresse_SaisiePers.add(this.jtx_saisieAdressePays);

    jp_panelAdresse.setLayout(new BorderLayout());

```

```

        jp_panelAdresse.add(new JLabel ("Adresse
",JLabel.CENTER),BorderLayout.NORTH);
        jp_panelAdresse.add(jp_panelAdresse_Labels,BorderLayout.WEST);
        jp_panelAdresse.add(jp_panelAdresse_SaisiePers,BorderLayout.CENTER);
        ///////////////////////////////////
        //jp_panelSaisiePers.add(jp_panelAdresse);

        // Ajout des sous-panneaux ...
        // utilisation d'un gestionnaire BorderLayout
        // avec écart de 5 pixels entre les composants
        jp_panelPersonne.setLayout(new BorderLayout());
        jp_panelPersonne.add(jp_panelLabels,BorderLayout.WEST);
        jp_panelPersonne.add(jp_panelSaisiePers,BorderLayout.CENTER);

        ///////////////////////////////////
        // Panel Medecin
        ///////////////////////////////////
        JPanel jp_panelMedecin= new JPanel();//flowlayout par défaut ...
        jp_panelMedecin.add(new JLabel ("Médecin Traitant (Nom et Prénom à saisir)
: "));

        jp_panelMedecin.add(this.jtx_saisieMedecin);

        ///////////////////////////////////
        // Panel Patient => regroupant les panels Personne et Medecin HORS NIR
        ///////////////////////////////////
        JPanel jp_panelPatient= new JPanel();
        jp_panelPatient.setLayout(new GridLayout(3,1));
        jp_panelPatient.add(jp_panelPersonne);
        jp_panelPatient.add(jp_panelAdresse);
        jp_panelPatient.add(jp_panelMedecin);

        // Panel Boutons
        ///////////////////////////////////
        JPanel jp_panelBoutons= new JPanel();//flowlayout par défaut ...
        jp_panelBoutons.add(jb_Valider);
        jp_panelBoutons.add(jb_Quitter);

        ///////////////////////////////////
        // AU FINAL
        this.setLayout(new BorderLayout());
        this.add(jp_panelNIR,BorderLayout.NORTH);
        this.add(jp_panelPatient,BorderLayout.CENTER);
        this.add(jp_panelBoutons,BorderLayout.SOUTH);

    } // fin constructeur

    public static void main(String args[])
    {
        JFrame fenetreTest = new JFrame();
        JPanel mainPanel = new PanelCreerPatient();

        fenetreTest.setTitle("Test");
        fenetreTest.getContentPane().add(mainPanel);
        fenetreTest.setSize(500,500);
        fenetreTest.setVisible(true);

        new PanelCreerPatient();
    }
}

```

Documentation JOptionPane :

☞ Show an error dialog that displays the message, 'alert':
JOptionPane.showMessageDialog(null, "alert",
"alert", JOptionPane.ERROR_MESSAGE);

☞ Show an internal information dialog with the message, 'information':
JOptionPane.showInternalMessageDialog(mainPanel,
"information", "information", JOptionPane.INFORMATION_MESSAGE);

Idem mais associé au panel ⇒ à l'intérieur du panel

☞ Show an information panel with the options yes/no and message
'choose one':
JOptionPane.showConfirmDialog(null, "choose
one", "choose one", JOptionPane.YES_NO_OPTION);

☞ Show an internal information dialog with the options yes/no/cancel
and message 'please choose one' and title information:
JOptionPane.showInternalConfirmDialog(frame, "please choose one", "information",
JOptionPane.YES_NO_CANCEL_OPTION, JOptionPane.INFORMATION_MESSAGE);

Idem mais associé au panel ⇒ à l'intérieur du panel

☞ Show a warning dialog with the options OK, CANCEL, title 'Warning', and message 'Click OK to continue':
Object[] options = { "OK", "CANCEL" };
JOptionPane.showOptionDialog(null, "Click OK to continue", "Warning",
JOptionPane.DEFAULT_OPTION,
JOptionPane.WARNING_MESSAGE,
null, options, options[0]);

☞ Show a dialog asking the user to type in a String:
String inputValue =
JOptionPane.showInputDialog("Please input a
value");

☞ Show a dialog asking the user to select a String:
Object[] possibleValues = { "First", "Second",
"Third" };
Object selectedValue =
JOptionPane.showInputDialog(null,
"Choose one", "Input",
JOptionPane.INFORMATION_MESSAGE, null,
possibleValues, possibleValues[0])

showMessageDialog

```
public static void showMessageDialog(Component parentComponent,  
                                   Object message,  
                                   String title,  
                                   int messageType)  
    throws HeadlessException
```

Brings up a dialog that displays a message using a default icon determined by the messageType parameter.

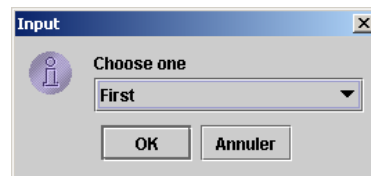
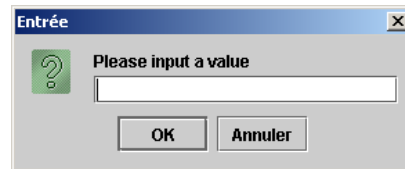
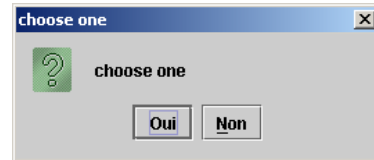
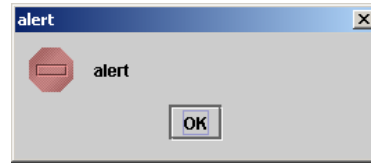
Parameters:

parentComponent - determines the Frame in which the dialog is displayed, if null, or if the parentComponent has no Frame, a default Frame is used
message - the Object to display
title - the title string for the dialog
messageType - the type of message to be displayed: ERROR_MESSAGE, INFORMATION_MESSAGE, WARNING_MESSAGE, QUESTION_MESSAGE, or PLAIN_MESSAGE

Throws: HeadlessException - if GraphicsEnvironment.isHeadless returns true

See Also: GraphicsEnvironment.isHeadless()

en popup



Remarque : Pour le menu Apropos, c'est la méthode `showMessageDialog` qui nous intéresse :