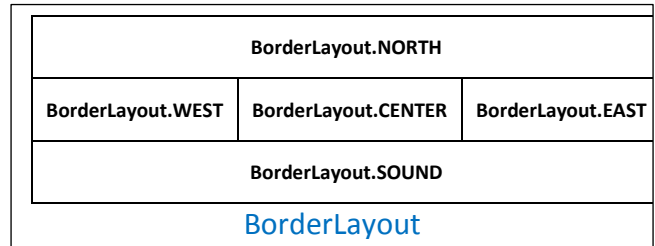
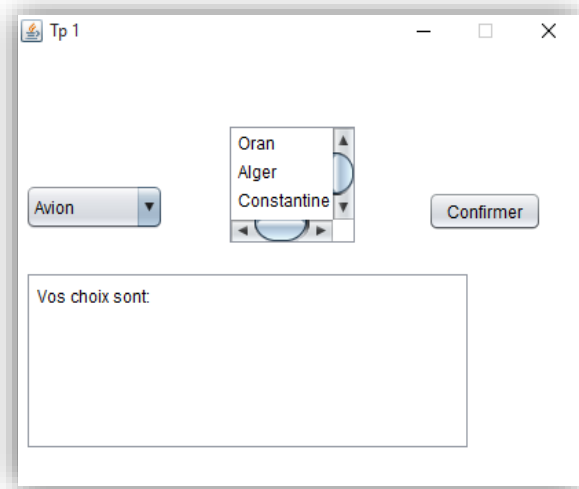


## TPIHM 01 2021/2022

### Exercice01 :



```
import java.awt.*;                                /* Bibliothèque */
import java.awt.event.ActionEvent;                /* Bibliothèque Action des événements */
import java.awt.event.ActionListener;
import java.awt.event.*;
import java.util.Vector;                          /* Bibliothèque vecteur de liste des villes */
import javax.swing.*;                             /* Bibliothèque graphique */
import javax.swing.border.EmptyBorder;

public class Exercice01 extends JFrame {
    protected Vector<String> list1;
    private static final Dimension DEFAULT_TEXT_FIELD_DIMENSION = new Dimension(0, 200);

    public Exercice01 () {
        super("Tp 1"); // l'intitulé de la fenêtre
        // pour la fermer
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // panel conteneur principal
        /* type de découpage de panel est BorderLayout */
        JPanel conteneur1 = (JPanel) this.getContentPane();
        conteneur1.setLayout(new BorderLayout(20,20));

        // question 1 :
        // bouton à choix
        JComboBox c = new JComboBox();
        c.addItem("avion");
        c.addItem("bus");
        c.addItem("taxi");
        c.addItem("train");

        // question 2 :
        // vecteur contenant les villes
        list1 = new Vector<String>();
        list1.addElement("Oran");
        list1.addElement("Alger");
        list1.addElement("Constantine");
        list1.addElement("Mostaganem");
        list1.addElement("Relizane");
        list1.addElement("Tiaret");
    }
}
```

```

list1.addElement("Tlemcen");

// affectation vecteur à la liste
JList textlist1 = new JList(list1);

// type de la sélection dans la liste
textlist1.setSelectionMode(ListSelectionMode.SINGLE_SELECTION);

// ajouter scroll à la liste
JScrollPane scrollpane = new JScrollPane(textlist1);

// dimension de la liste
scrollpane.setPreferredSize(new Dimension(200,100));

// question 3 :
// Bouton confirmer
JButton suite = new JButton("Confirmer");

// Evènement lors du clic sur le bouton (changement de couleur en vert)
//*****
suite.addActionListener(new ActionListener() {    /**EVENEMENT EXEMPLE
    public void actionPerformed(ActionEvent e) {
        //System.out.println("bonjour");
        suite.setBackground(Color.green);
    }
//*****
});

// question 4 :
// Text field
JTextField texte= new JTextField(16);
texte.setPreferredSize(DEFAULT_TEXT_FIELD_DIMENSION);

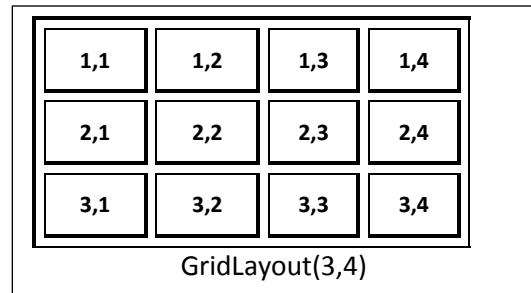
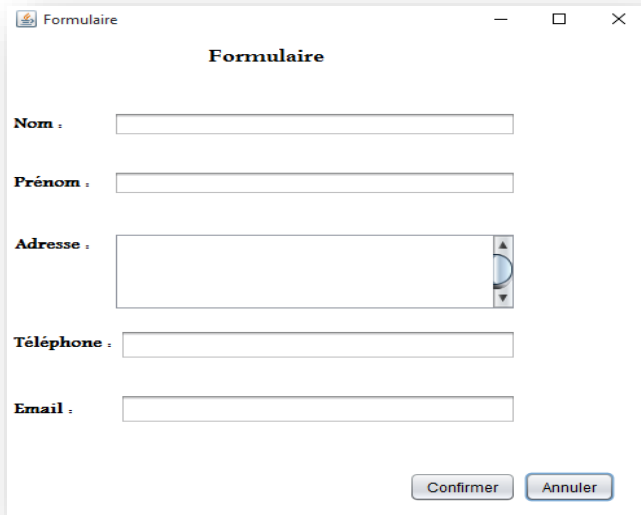
// Remplissage de la panel conteneur1
conteneur1.add(c, BorderLayout.WEST);
conteneur1.add(scrollpane, BorderLayout.CENTER); /* la liste des villes */
conteneur1.add(suite, BorderLayout.EAST);
conteneur1.add(texte, BorderLayout.SOUTH);
// Taille Bordure entre BorderLayout panel conteneur1
conteneur1.setBorder(new EmptyBorder(10,10,10,10));

// calcul des dimensions de la fenêtre
this.pack();
}

public static void main(String[] args) {
    // TODO code application logic here
    Exercice01 ex01=new Exercice01();
// Affichage de la fenêtre
    ex01.setVisible(true);
}
}

```

## Exercice02 :



```
import java.awt.*; /* Bibliothèque */
import java.awt.event.ActionEvent; /* Bibliothèque Action des événements */
import java.awt.event.ActionListener;
import java.util.Vector; /* Bibliothèque vecteur de liste des villes */
import javax.swing.*; /* Bibliothèque graphique */
import javax.swing.border.EmptyBorder;

public class Exercice02 extends JFrame {
    protected Vector<String> adresselist;

    public Exercice02 () {

        super("Formulaire"); // l'intitulé de la fenêtre
        // pour la fermer
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // panel conteneur principal
        /* type de découpage de panel est GridLayout */
        JPanel conteneur2 = (JPanel) this.getContentPane();
        conteneur2.setLayout(new GridLayout(7, 2)); /* type de découpage de panel est GridLayout */

        // question 1 :
        // création des labels
        JLabel videlab = new JLabel(" "); // label vide pour afficher les messages GridLayout11
        JLabel Formulairerelab = new JLabel("Formulaire");
        JLabel nomlab = new JLabel("Nom :");
        JLabel prenomlab = new JLabel("Prénom :");
        JLabel adresselab = new JLabel("Adresse :");
        JLabel telephonelab = new JLabel("Téléphone :");
        JLabel emailab = new JLabel("Email :");

        // création des TextField
        JTextField nomtext = new JTextField(16);
        JTextField prenomtext = new JTextField(16);

        // vecteur contenant les adresses
        adresselist = new Vector<String>();
        adresselist.addElement("Oran");
        adresselist.addElement("Alger");
```

```

adresselist.addElement("Constantine");
adresselist.addElement("Mostaganem");
adresselist.addElement("Relizane");
adresselist.addElement("Tiaret");
adresselist.addElement("Tlemcen");

// affectation vecteur à la liste
JList textadresselist = new JList(adresselist);

// type de la sélection dans la liste
textadresselist.setSelectionMode(ListSelectionMode.SINGLE_SELECTION);

// ajouter scroll à la liste
JScrollPane scrollpane = new JScrollPane(textadresselist);
scrollpane.setPreferredSize(new Dimension(200,100));

// création des TextField
JTextField telephonetext = new JTextField(16);
JTextField emailtext = new JTextField(16);

// question 2 :
// création des boutons confirmer et annuler
JButton confirmboutton = new JButton("Confirmer");
JButton annulerboutton = new JButton("Annuler");

// Remplissage du panel conteneur 2
conteneur2.add(videlab); // label vide pour afficher les messages GridLayout11
conteneur2.add(Formulairelab);
conteneur2.add(nomlab);
conteneur2.add(nomtext);
conteneur2.add(prenomlab);
conteneur2.add(prenomtext);
conteneur2.add(adresselab);
conteneur2.add(scrollpane); /* la liste des adresses */
conteneur2.add(telephonelab);
conteneur2.add(telephonetext);
conteneur2.add(emaillab);
conteneur2.add(emailtext);
conteneur2.add(confirmboutton);
conteneur2.add(annulerboutton);
// Taille Bordure entre BorderLayout panel conteneur1
conteneur2.setBorder(new EmptyBorder(10,10,10,10));

/*
o Si les champs sont vides des messages d'erreurs s'afficheront,
o Si les champs sont correctement remplis un message de confirmation est affiché.
*/

// Evènement : clic sur le bouton confirme si les champs sont vides alors focus
//*****
confirmboutton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (nomtext.getText().isEmpty()) {
            videlab.setText("le nom est vide");
            nomtext.grabFocus();
        }
        else
            if (prenomtext.getText().isEmpty()) {
                videlab.setText("le prénom est vide");
                prenomtext.grabFocus();
            }
    }
}

```

```

else
  if (telephonetext.getText().isEmpty()) {
    videlab.setText("le télép est vide");
    telephonetext.grabFocus();
  }
  else
    if (emailtext.getText().isEmpty()) {
      videlab.setText("le mail est vide");
      emailtext.grabFocus();
    }
    else { /** afficher la fenêtre affichant c'est confirmé
      videlab.setText(" ");
      /** création fenêtre
      JFrame jf = new JFrame("Confirmation");
      JPanel panneau5 = new JPanel();
      JPanel panneau6 = new JPanel();
      JButton bouton5 = new JButton("Ok"); // bouton ok fenêtre
      JLabel label5 = new JLabel("C'est confirmé");

      // panel conteneur panneau 5
      /* type de découpage de panel est BorderLayout */
      BorderLayout bl = new BorderLayout();
      panneau5.setLayout(bl);
      panneau5.add(label5, BorderLayout.NORTH );
      panneau5.add(bouton5, BorderLayout.SOUTH );
      jf.getContentPane().add(panneau5);
      jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      jf.setVisible(true);
      jf.pack();

      bouton5.addActionListener(new ActionListener() { // évènement ok
        public void actionPerformed(ActionEvent e) {
          System.exit(0);
        } // fin actionPerformed

      }); // fin évènement ok
    } // else

  } // fin actionPerformed
} // fin Evènement : clic sur le bouton confirme

// Evènement : clic sur le bouton annuler pour vider les champs
//*****

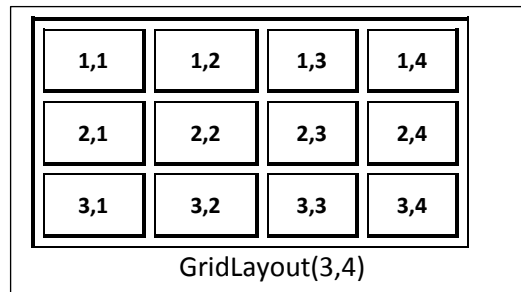
annulerboutton.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent e) {
    videlab.setText("");
    nomtext.setText("");
    nomtext.grabFocus();
    prenomtext.setText("");
    telephonetext.setText("");
    emailtext.setText("");
  }
} //*****
}); // fin Evènement : clic sur le bouton annuler

// calcul des dimensions de la fenêtre

```

```
    this.pack();
}
public static void main(String[] args) {
    // TODO code application logic here
    Exercice02 ex02=new Exercice02();
    ex02.setVisible(true);
}
}
```

### Exercice03 :



```
import java.awt.*; /* Bibliothèque */
import java.awt.event.ActionEvent; /* Bibliothèque Action des évènements */
import java.awt.event.ActionListener;
import java.util.Vector; /* Bibliothèque vecteur de liste des villes */
import javax.swing.*; /* Bibliothèque graphique */
import javax.swing.border.EmptyBorder;

public class Exercice03 extends JFrame {
    protected Vector<String> list1;
    public Exercice03 () {

        super("Authentification"); // l'intitulé de la fenêtre

        // pour la fermer
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // panel conteneur principal
        /* type de découpage de panel est GridLayout */
        JPanel conteneur3 = (JPanel) this.getContentPane();
        conteneur3.setLayout(new GridLayout(4, 2)); /* type de découpage de panel est GridLayout */

        // question :
        // création des labels
        JLabel videlab = new JLabel(" "); // label vide pour afficher les messages GridLayout11
        JLabel authentiflab = new JLabel("Authentification ");
        JLabel utilisateurlab = new JLabel("Nom d'utilisateur :");
        JLabel motpasselab = new JLabel("Mot de passe:");
        JLabel vide2lab = new JLabel(" ");

        // création des TextField
        JTextField utilisateurtext= new JTextField(8);
        JTextField motpassetext= new JTextField(8);

        // création du bouton verifier
        JButton verifierboutton = new JButton("Vérifier");

        // Remplissage du panel conteneur3
        conteneur3.add(videlab);
        conteneur3.add(authentiflab);
        conteneur3.add(utilisateurlab);
        conteneur3.add(utilisateurtext);
        conteneur3.add(motpasselab);
        conteneur3.add(motpassetext);
        conteneur3.add(vide2lab);
        conteneur3.add(verifierboutton);
    }
}
```

```

// Evènement : clic sur le bouton verifier si les champs sont vides alors focus
//*****
verifierboutton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (utilisateurtext.getText().isEmpty()) {
            videlab.setText("le nom d'utilisateur est vide");
            utilisateurtext.grabFocus();
        }
        else
            if (motpassetext.getText().isEmpty()) {
                videlab.setText("le mot de passe est vide");
                motpassetext.grabFocus();
            }
                else {
                    /** afficher la fenetre affichant c'est correcte ou non
                    videlab.setText(" ");
                    /** création fenetre
                    JFrame jf = new JFrame("Confirmation");
                    // panel conteneur panneau 5
                    JPanel panneau5 = new JPanel();
                    JButton bouton5 = new JButton("Ok");
                    JLabel label5 = new JLabel();
                    String texte1="username";
                    String texte2="password";
                    Integer texte3=utilisateurtext.getText().trim().compareTo(texte1);
                    Integer texte4=motpassetext.getText().trim().compareTo(texte2);
                    if (texte3==0) {
                        if (texte4==0)
                            label5.setText("informations correctes");
                        else
                            label5.setText("password non correcte");
                    }
                    else
                        label5.setText("user non correcte");
                    panneau5.add(bouton5);
                    panneau5.add(label5);
                    /* type de découpage de panel est BorderLayout */
                    BorderLayout bl = new BorderLayout();
                    Panneau5.setLayout(bl);
                    Panneau5.add(label5, BorderLayout.NORTH );
                    Panneau5.add(label5, BorderLayout.SOUTH );
                    jf.getContentPane().add(panneau5);
                    jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                    jf.setVisible(true);
                    jf.pack();
                    bouton5.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                            System.exit(0);
                        } // fin actionPerformed
                    });
                } // fin else
            } // fin actionPerformed
//*****
});

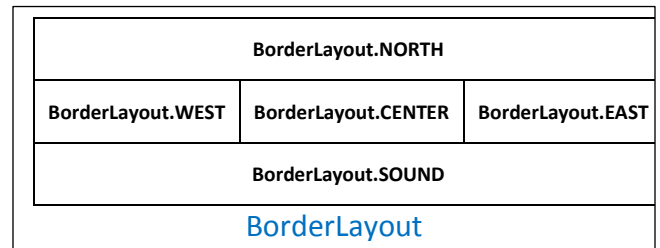
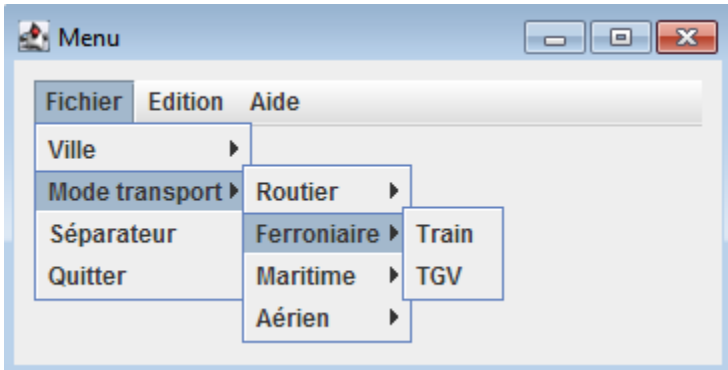
```



```
// calcul des dimensions de la fenetre
this.pack();
}
public static void main(String[] args) {
    // TODO code application logic here
    Exercice03 ex03=new Exercice03();
    ex03.setVisible(true);
}
}
```

## TPIHM 02 2021/2022

### Exercice01 :



```

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Vector;
import javax.swing.*;
import javax.swing.border.EmptyBorder;

/* Bibliothèque */
/* Bibliothèque Action des événements */
/* Bibliothèque vecteur de liste des villes */
/* Bibliothèque graphique */

public class Exercice01t02 extends JFrame {
    public Exercice01t02 () {

        super("Menu");

        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // remplissage du panel principal
        /* type de découpage de panel est BorderLayout */
        JPanel conteneur1 = (JPanel) this.getContentPane();
        conteneur1.setLayout(new BorderLayout(20,20));

        // Création et Déclaration des menus et sous menus
        JMenuBar barreDeMenu = new JMenuBar(); // Barre générale des menus
        JMenu menu1 = new JMenu("Fichier"); // Menu Fichier
        JMenu smenu1 = new JMenu("Ville"); // Sous menu Ville du menu Fichier
        JMenu ssmenu11 = new JMenu("Nord"); // Sous sous menu Nord du sous menu Ville
        JMenuItem item11 = new JMenuItem("Alger"); // item Alger sous sous menu Nord
        JMenu ssmenu12 = new JMenu("Est"); // .....
        JMenuItem item12 = new JMenuItem("Annaba"); // .....
        JMenu ssmenu13 = new JMenu("Ouest");
        JMenuItem item13 = new JMenuItem("Oran");
        JMenu ssmenu14 = new JMenu("Sud");
        JMenuItem item14 = new JMenuItem("Adrar");

        JMenu smenu2 = new JMenu("Mode transport"); // Sous menu Mode transport du menu Fichier
        JMenu ssmenu21 = new JMenu("Routier"); // Sous sous menu Routier du sous menu Mode transport
        JMenuItem item21 = new JMenuItem("Bus"); // item Bus sous sous menu Routier
        JMenuItem item22 = new JMenuItem("Taxi"); // .....
        JMenu ssmenu22 = new JMenu("Ferronaire"); // .....
        JMenuItem item23 = new JMenuItem("Train");
        JMenuItem item24 = new JMenuItem("TGV");
        JMenu ssmenu23 = new JMenu("Maritime");
        JMenuItem item25 = new JMenuItem("Bateau");
        JMenuItem item26 = new JMenuItem("Navette");
        JMenu ssmenu24 = new JMenu("Aérien");
        JMenuItem item27 = new JMenuItem("Avion");
    }
}

```

```

JMenuItem item28 = new JMenuItem("Helicopter");

JMenuItem item3 = new JMenuItem("Séparateur");// Sous menu du menu Fichier
JMenuItem item4 = new JMenuItem("Quitter"); // Sous menu du menu Fichier

JMenu menu2 = new JMenu("Edition"); // Menu Edition
JMenuItem item5 = new JMenuItem("Précédent"); // item Précédent du menu Edition
JMenuItem item6 = new JMenuItem("Annuler"); // item Annuler du menu Edition
JMenuItem item7 = new JMenuItem("Supprimer"); // item Supprimer du menu Edition

JMenu menu3 = new JMenu("Aide"); // Menu Aide
JMenuItem item8 = new JMenuItem("A-propos"); // item A-propos du menu Aide

// Remplissage et relie des menus et sous menus
menu1.add(smenu1); // Remplir le menu 1
menu1.add(smenu2);
menu1.add(item3);
menu1.add(item4);

smenu1.add(ssmenu11); // Remplir le sous menu 1
smenu1.add(ssmenu12);
smenu1.add(ssmenu13);
smenu1.add(ssmenu14);
ssmenu11.add(item11); // Remplir le sous sous menu 11
ssmenu12.add(item12); // Remplir le sous sous menu 12
ssmenu13.add(item13); // Remplir le sous sous menu 13
ssmenu14.add(item14); // Remplir le sous sous menu 14

smenu2.add(ssmenu21); // Remplir le sous menu 2
smenu2.add(ssmenu22);
smenu2.add(ssmenu23);
smenu2.add(ssmenu24);
ssmenu21.add(item21); // Remplir le sous sous menu 21
ssmenu21.add(item22); // Remplir le sous sous menu 22
ssmenu22.add(item23); // Remplir le sous sous menu 23
ssmenu22.add(item24); // Remplir le sous sous menu 24
ssmenu23.add(item25); // Remplir le sous sous menu 25
ssmenu23.add(item26); // Remplir le sous sous menu 26
ssmenu24.add(item27); // Remplir le sous sous menu 27
ssmenu24.add(item28); // Remplir le sous sous menu 28

menu2.add(item5); // Remplir le menu 2
menu2.add(item6);
menu2.add(item7);

menu3.add(item8); // Remplir le menu 3

barreDeMenu.add(menu1); // Remplir la Barre générale des menus
barreDeMenu.add(menu2);
barreDeMenu.add(menu3);

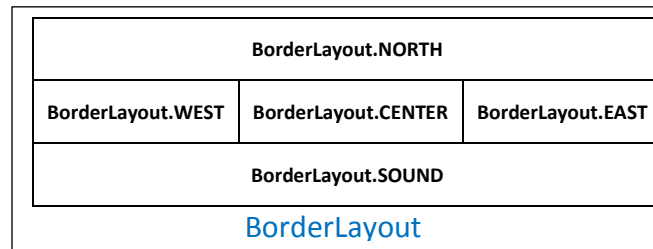
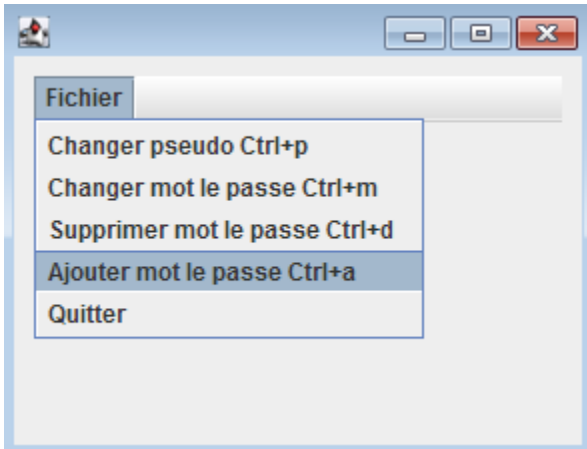
// Evenement lors du clique sur le menu
//*****
item4.addActionListener(new ActionListener() { //**EVENEMENT EXEMPLE
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});
//*****
});

```

```
// Remplissage de la panel conteneur1
    conteneur1.add(barreDeMenu, BorderLayout.NORTH);
// Taille Bordure entre BorderLayout panel conteneur1
    conteneur1.setBorder(new EmptyBorder(10,10,10,10));

// calcul des dimentions de la fenetre
this.pack();
}
    public static void main(String[] args) {
        // TODO code application logic here
        Exercice01t02 ex03=new Exercice01t02();
        ex03.setVisible(true);
    }
}
```

## Exercice02 :



```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Vector;
import javax.swing.*;
import javax.swing.border.EmptyBorder;

/* Bibliothèque */
/* Bibliothèque Action des événements */

/* Bibliothèque vecteur de liste des villes */
/* Bibliothèque graphique */

public class Exercice02t2 extends JFrame {
    public Exercice02t2 () {

        super("");

        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // remplissage du panel principal
        /* type de découpage de panel est BorderLayout */
        JPanel conteneur1 = (JPanel) this.getContentPane();
        conteneur1.setLayout(new BorderLayout(20,20));

        JLabel utilisateurlab = new JLabel("Nom d'utilisateur :");
        JLabel motpasselab = new JLabel("Mot de passe:");

        JTextField utilisateurtext= new JTextField(8);
        JTextField motpasetext= new JTextField(8);

        // Déclaration des menus et sous menus
        JMenuBar barreDeMenu = new JMenuBar(); // Barre générale des menus
        JMenu menu1 = new JMenu("Fichier"); // Menu Fichier
        JMenuItem item1 = new JMenuItem("Changer pseudo : ctrl+p");// item
        JMenuItem item2 = new JMenuItem("Changer mot le passe : ctrl+m"); // item
        JMenuItem item3 = new JMenuItem("Supprimer mot le passe : ctrl+d"); // item
        JMenuItem item4 = new JMenuItem("Ajouter mot le passe : ctrl+a"); // item
        JMenuItem item5 = new JMenuItem("Quitter"); // item
        menu1.add(item1); // Remplir le menu 1
        menu1.add(item2);
        menu1.add(item3);
        menu1.add(item4);
        menu1.add(item5);

        barreDeMenu.add(menu1); // Remplir la Barre générale des menus
```

```

/*
 * Si le mot de passe est supprimé vous ne pouvez pas le changer,
 * Si un mot de passe existe vous ne pouvez pas l'ajouter.
 */

// Evènement lors du clic sur le menu
//*****
utilisateurtext.setText("nd");
motpassetext.setText("mp");
item1.addActionListener(new ActionListener() {    /**EVENEMENT change pseudo
    public void actionPerformed(ActionEvent e) {
        utilisateurtext.setText("nd2");
    }
//*****
});

item1.setAccelerator(KeyStroke.getKeyStroke(
    KeyEvent.VK_P,KeyEvent.CTRL_MASK)); /**EVENEMENT pour raccourcie ctrl+P

item2.addActionListener(new ActionListener() {    /**EVENEMENT change mot de passe
    public void actionPerformed(ActionEvent e) {
        if (motpassetext.getText().trim().compareTo("")==1)
            motpassetext.setText("mp3");
    }
//*****
});

Item2.setAccelerator(KeyStroke.getKeyStroke(
    KeyEvent.VK_M,KeyEvent.CTRL_MASK)); /**EVENEMENT pour raccourcie ctrl+M

item3.addActionListener(new ActionListener() {    /**EVENEMENT supprime mot de passe
    public void actionPerformed(ActionEvent e) {
        motpassetext.setText("");
    }
//*****
});

Item3.setAccelerator(KeyStroke.getKeyStroke(
    KeyEvent.VK_D,KeyEvent.CTRL_MASK)); /**EVENEMENT pour raccourcie ctrl+D

item4.addActionListener(new ActionListener() {    /**EVENEMENT ajoute mot de passe
    public void actionPerformed(ActionEvent e) {
        if (motpassetext.getText().trim().compareTo("")==0)
            motpassetext.setText("mp2");
    }
//*****
});

Item4.setAccelerator(KeyStroke.getKeyStroke(
    KeyEvent.VK_A,KeyEvent.CTRL_MASK)); /**EVENEMENT pour raccourcie ctrl+A

item5.addActionListener(new ActionListener() {    /**EVENEMENT EXEMPLE
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
//*****
});

```

```
// Remplissage de la panel conteneur1
```

```
conteneur1.add(barreDeMenu, BorderLayout.NORTH);
conteneur1.add(utilisateurlab, BorderLayout.WEST);
conteneur1.add(utilisateurtext, BorderLayout.CENTER);
conteneur1.add(motpasselab, BorderLayout.EAST);
conteneur1.add(motpassetext, BorderLayout.SOUTH);
// Taille Bordure entre BorderLayout panel conteneur1
conteneur1.setBorder(new EmptyBorder(10,10,10,10));

// calcul des dimensions de la fenetre
this.pack();
}
    public static void main(String[] args) {
    // TODO code application logic here
        Exercice02t2 ex03=new Exercice02t2();
        ex03.setVisible(true);
    }
}
```