

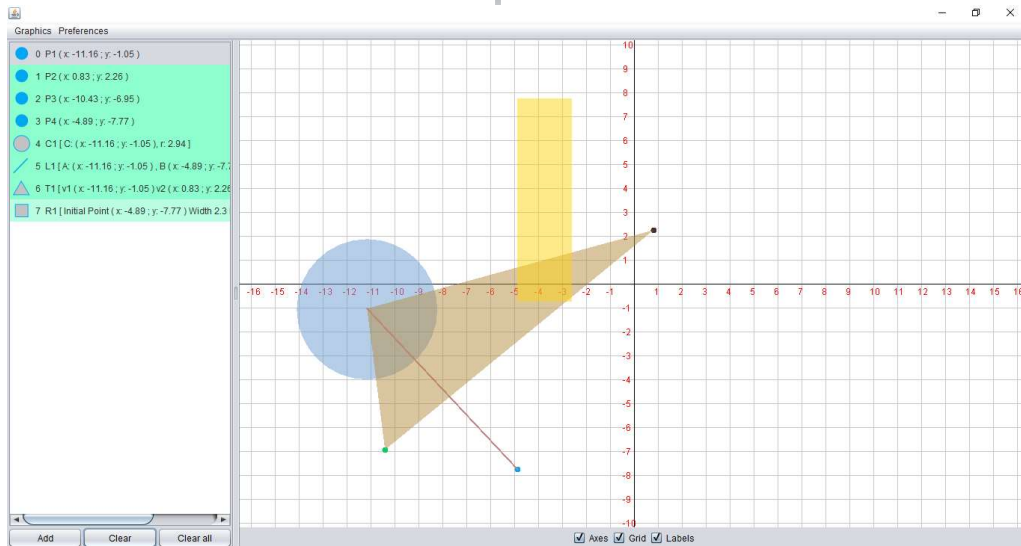
Rapport Technique N° 1 à rendre pour le
vendredi 21 février

Agustín Cartaya

Partie 1

Graphics	3
	3
Description	3
L'application vous permet de :	3
Fenêtres d'application et utilitaires	4
Fenêtre initiale (JFrame)	4
Arbre	5
Éléments	6
Barre d'outils (JMenuBar).	6
Menu graphique (JMenu)	6
Préférences de menu (JMenu)	6
Liste (JList)	6
Add (JButton)	6
Clear (JButton)	6
Clear All (JButton)	6
Axes (JCheckBox)	7
Grilles (JCheckBox)	7
Étiquettes (JCheckBox)	7
Form pour Ajouter des elements (JDialog)	8
Arbre	9
Éléments	10
Liste des éléments à ajouter (JComboBox)	10
Extrait (JButton)	10
Ajouter (JButton)	10
Annuler (JButton)	10
Liste d'information (JDialog)	11
Arbre	11
Éléments	11
Zone d'information (JTextArea)	11
Filtres (JCheckBox)	11
Compteur d'objets affichés (JLabel)	11
Formulaire d'édition couleur (JDialog)	12
Arbre	12
Éléments	12
Boutons de couleur (JButton)	12
Formulaire d'édition de police (JDialog)	13
Abol Abol	13
Éléments	13
Boutons (JButton)	13

Graphics



Description

Graphics est une application développée en Java dont la fonctionnalité est de graphiquer des éléments 2D sur un plan composé de deux axes de coordonnées (X,Y).

L'application vous permet de :

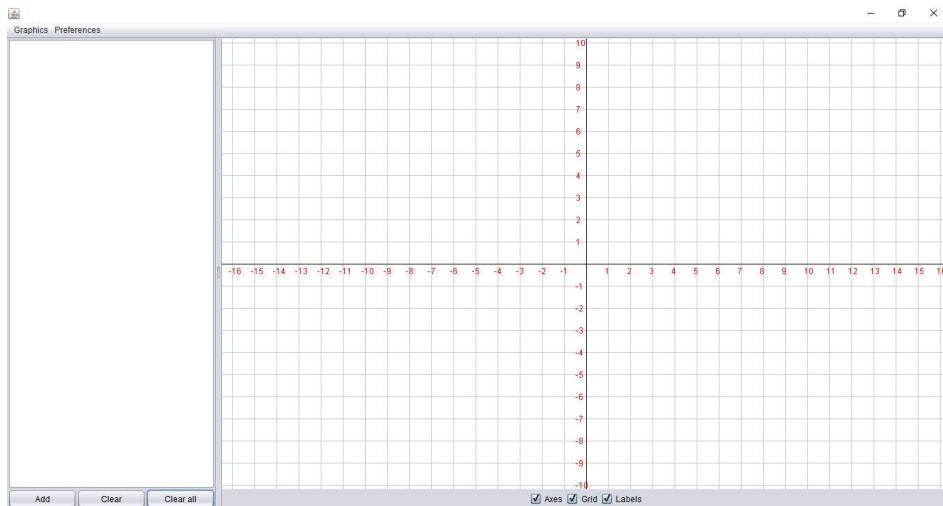
- Graphiquer éléments tels que :
 - Points. Avec leurs coordonnées X,Y
 - Segments. Avec les coordonnées de leurs extrémités
 - Triangles. Avec les coordonnées de leurs 3 sommets
 - Cercles. Avec les coordonnées de leurs centres et la valeur de leurs rayons
 - Rectangles. Avec les coordonnées de leurs coins supérieur gauche et les valeurs de leurs hauteurs et largeurs
- Obtenir les informations des éléments tracés : l'application a la possibilité d'afficher les informations de tous ses éléments, y compris leurs noms, leur type, si l'emplacement, et leur zone et périmètre (si possible) dans un panneau
- Gérer les couleurs: L'application dispose d'un panneau pour gérer la plupart des couleurs qui peuvent être consultés dans l'application, y compris les couleurs de l'espace graphique et la liste des éléments.
- Gérer les polices: L'application dispose d'un panneau pour gérer les polices qui vous permet de changer la police, la taille et le style à certains éléments de l'espace graphique ou la liste des éléments

- Gérer l'espace graphique: L'application a différentes cases à cocher qui vous permettent de montrer ou de cacher des éléments de l'espace graphique

Fenêtres d'application et utilitaires

L'application dispose actuellement d'une fenêtre principale (JFrame) qui contient l'espace graphique et la liste des éléments, puis 4 fenêtres secondaires (JDialog) destiné à afficher, ajouter ou modifier. les fenêtres seront détaillées ci-dessous.

Fenêtre initiale (JFrame)



Il s'agit de la fenêtre initiale du programme qui a une barre d'outils et un panneau de séparation (JSplitPane) qui contiendra l'espace graphique et la liste des éléments sur chacun de ses côtés.

Arbre

JFrame Grapics

- ✓ (BorderLayout)
- ✓ JMenuBar menuBar
 - JMenu jMenuItem1 "Graphics"
 - JMenuItem addDefaultElements "Add default Elements"
 - JMenuItem showElementDescriptions "Elements Description"
 - JMenu jMenuItem2 "Preferences"
 - JMenuItem showColorsForm "Colors"
 - JMenuItem showFontform "Fonts"
- ✓ JSplitPane jSplitPane1, CENTER
 - JPanel jPanel1
 - (BorderLayout)
 - JPanel jPanel3, CENTER
 - (GridLayout [1,1])
 - JScrollPane jScrollPane1
 - JList elementsList
 - JPanel jPanel4, SOUTH
 - (GridLayout [2,1])
 - JPanel jPanel2
 - (FlowLayout)
 - JButton jbAdd "Add"
 - JButton clear "Clear"
 - JButton clearAll "Clear All"
 - JPanel jPanel4
 - (FlowLayout)
 - JRadioButton lock "Lock"
 - JRadioButton unlock "Lnlock"
 - JPanel jPanelCenter
 - (BorderLayout)
 - JPanel jPanel5, SOUTH
 - (FlowLayout)
 - JCheckBox axesVisibility "Axes"
 - JCheckBox gridVisibility "Grids"
 - JCheckBox labelsVisibility "Labels"
 - GraphicSpace, CENTER

Éléments

Barre d'outils (JMenuBar).

Menu graphique (JMenu)

Contient des options d'édition ou d'informations pour l'espace graphique

Ajouter des éléments par défaut (JMenuitem)

Option qui vous permet d'ajouter les éléments prédéfinis à l'espace graphique, ce sont:

- 4 points avec des coordonnées aléatoires et 3 d'entre eux avec des couleurs aléatoires, en plus le premier point est réglé pour ne pas être visible,
- 2 cercles C1 et C2 avec leurs centres en P1 et P2 respectivement et rayons aléatoires
- 2 lignes L1 et L2 dont les extrémités sont P1-P4 et P2-P3 respectivement
- 1 Triangle dont les extrémités sont P1,P2,P3
- 1 rectangle avec dont l'extrémité supérieure gauche est P4 et la largeur et la longueur sont aléatoires

Description des éléments (JMenuitem)

Possibilité d'afficher les informations du panneau sur les éléments du graphique

Préférences de menu (JMenu)

Contient des options pour la personnalisation des fenêtres

Colors (JMenuitem)

Option qui vous permet d'afficher le panneau pour l'édition de couleur

Fonts (JMenuitem)

Option qui vous permet d'afficher le panneau pour l'édition de polices

Liste (JList)

Liste qui contient tous les éléments ajoutés à l'espace graphique, permet la sélection des éléments et lorsque la souris est sur l'un d'eux, il change de couleur dans l'espace graphique

Add (JButton)

Un bouton qui vous permet d'ajouter un nouvel élément à l'espace graphique. Déplier le formulaire pour ajouter le nouvel élément

Clear (JButton)

Bouton qui vous permet de supprimer les éléments sélectionnés dans la liste

Clear All (JButton)

Un bouton qui vous permet de supprimer tous les éléments de la liste

Axes (JCheckBox)

CheckBox qui permet d'afficher ou de ne pas afficher les axes principaux dans l'espace graphique

Grilles (JCheckBox)

CheckBox qui permet d'afficher ou de guider les lignes dans l'espace graphique

Étiquettes (JCheckBox)

CheckBox qui permet d'afficher la numérotation des divisions dans l'espace graphique

Lock (JRadioButton)

Si ce bouton radio est sélectionné, aucun élément ne peut être ajouté ou supprimé.

Unlock (JRadioButton)

Si ce bouton radio est sélectionné, des éléments peuvent être ajoutés et supprimés.

Form pour Ajouter des elements (JDialog)

The image displays four separate JDialog windows, each designed for adding a specific geometric element to a graphical workspace. Each window features a title bar with a close button (X) and a small icon on the left. The main content area of each dialog is divided into two parts: a left panel for input fields and a right panel for a graphical preview.

- Point Dialog:** The title bar shows a dropdown menu set to "Point". The left panel contains two text input fields labeled "Point X" and "Point Y", each with a placeholder "(Ex: 1,8)". Below these is a cyan-colored text box with the message "The values have to be number and the comas are accepted by points". At the bottom left is a "Color" label with a small color selection button. The right panel shows a blue circle. At the bottom are "Add" and "Cancel" buttons.
- Line Dialog:** The title bar shows a dropdown menu set to "Line". The left panel contains four text input fields labeled "ExtremeA X", "ExtremeA Y", "ExtremeB X", and "ExtremeB Y", each with a placeholder "(Ex: 1,8)". Below these is a cyan-colored text box with the message "The points formed by ExtremeA and ExtremeB have to be different". At the bottom left is a "Color" label with a small color selection button. The right panel shows a blue line segment. At the bottom are "Add" and "Cancel" buttons.
- Circle Dialog:** The title bar shows a dropdown menu set to "Circle". The left panel contains four text input fields: "Point X" and "Point Y" (with placeholder "(Ex: 1,8)"), and "radius" (with placeholder "(Ex: 1,8)"). There is also a "Fill" checkbox. Below these is a cyan-colored text box with the message "The radius have to be bigger than zero". At the bottom left is a "Color" label with a small color selection button. The right panel shows a blue circle with a grey fill. At the bottom are "Add" and "Cancel" buttons.
- Triangle Dialog:** The title bar shows a dropdown menu set to "Triangle". The left panel contains six text input fields: "VertexA X", "VertexA Y", "VertexB X", "VertexB Y", "VertexC X", and "VertexC Y", each with a placeholder "(Ex: 1,8)". There is also a "Fill" checkbox. Below these is a cyan-colored text box with the message "All the vertex formed by the points A, B, C have to be different and they can't be on the same line". At the bottom left is a "Color" label with a small color selection button. The right panel shows a blue triangle with a grey fill. At the bottom are "Add" and "Cancel" buttons.
- Rectangle Dialog:** The title bar shows a dropdown menu set to "Rectangle". The left panel contains four text input fields: "Init X", "Init Y", "Width", and "Height", each with a placeholder "(Ex: 1,8)". There is also a "Fill" checkbox. Below these is a cyan-colored text box with the message "The radius have to be bigger than zero". At the bottom left is a "Color" label with a small color selection button. The right panel shows a blue rectangle with a grey fill. At the bottom are "Add" and "Cancel" buttons.

Ce formulaire permet d'ajouter les différents éléments disponibles à l'espace graphique. Il contient un espace dynamique qui change en fonction de l'élément que vous souhaitez ajouter et en bas on peut trouver deux boutons Cancel et Add qui annulent ou ajoutent l'élément respectivement

Arbre

JDialog FormAddElement (JDialog)

- ✓ (BorderLayout)
- ✓ JPanel jPanel1, NORTH
 - (FlowLayout)
 - JComboBox addElementList
- ✓ JPanel jPanel2, CENTER
 - (BorderLayout)
 - JPanel jPanel7, CENTER
 - (BorderLayout)
 - JPanel jPanel5, SOUTH
 - (FlowLayout)
 - JLabel jLabel1 "Color"
 - JButton colorSelected " "
 - JPanel panelForm, CENTER
 - (GridLayout [1,1])
 - (Contenido dinámico)
 - JPanel jPanel4, EAST
 - (GridLayout [1,1])
 - JButton preview ""
- ✓ JPanel jPanel6, SOUTH
 - (FlowLayout)
 - JButton addElement "Add"
 - JButton cancel "Cancel"

Éléments

Liste des éléments à ajouter (JComboBox)

Cette liste contient les noms des éléments qui peuvent être ajoutés à l'espace graphique, lorsque vous le déployez et sélectionnez l'un d'eux, il modifie le formulaire en fonction des données dont l'élément a besoin et affiche une image de l'élément à ajouter dans le bouton de prévisualisation

Extrait (JButton)

Dans ce bouton, nous pouvons voir l'aperçu de l'élément que vous souhaitez ajouter

Ajouter (JButton)

Appuyer sur ce bouton ajoute l'élément à l'espace graphique et ferme le formulaire, en cas d'erreur une fenêtre contextuel s'affiche indiquant l'erreur

Annuler (JButton)

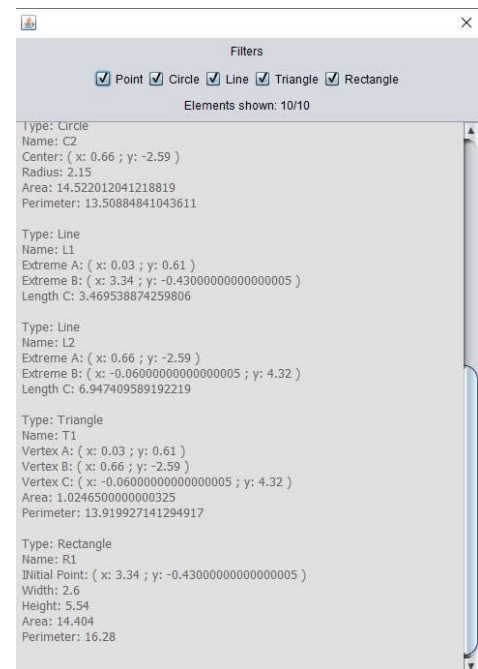
Appuyer sur ce bouton ferme le formulaire sans ajouter d'éléments

Liste d'information (JDialog)

Cette liste contient les informations des éléments dans l'espace graphique, il a également une série de filtres pour afficher uniquement les éléments désirés

Arbre

- ✓ JDialog Information
 - (BorderLayout)
 - JPanel jPanel1, NORTH
 - (GridLayout [3,1])
 - JPanel jPanel2
 - (FlowLayout)
 - JLabel jLabel1 "Filters"
 - JPanel jPanelFilters
 - (FlowLayout)
 - (Contenido dinámico)
 - JPanel jPanel3
 - (FlowLayout)
 - JLabel ElementsShown "(texte dynamique)"
 - JScrollPane jScrollPane1, CENTER
 - JTextArea elementsDescription "(texte dynamique)"



Éléments

Zone d'information (JTextArea)

C'est là que toutes les informations de l'article seront affichées

Filtres (JCheckBox)

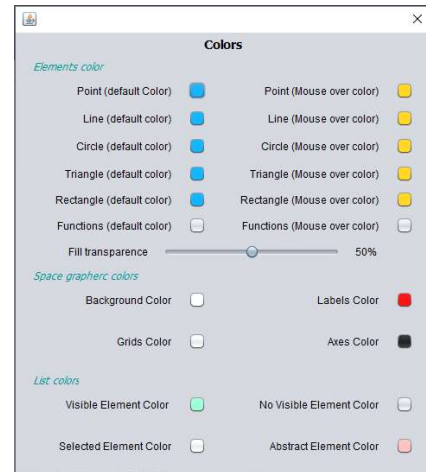
Filtres qui apparaîtront dynamiquement en fonction des types d'éléments que nous avons dans notre espace graphique, ceux-ci nous aideront à montrer ou non ces éléments dans la liste d'informations

Compteur d'objets affichés (JLabel)

Indique le nombre d'éléments affichés et le nombre total d'éléments

Formulaire d'édition couleur (JDialog)

Ce formulaire vous permet de modifier les couleurs de la fenêtre, y compris celles de la liste des éléments et celles de l'espace graphique.



Arbre

- ✓ JDialog FormColor
 - (BorderLayout)
 - JPanel jPanel2, NORTH
 - (FlowLayout)
 - JLabel jLabel1 "Colors"
 - JPanel jPanel1, CENTER
 - (GridLayout [1,4])
 - JPanel jPanel3
 - (BorderLayout)
 - JPanel jPanel6, NORTH
 - (FlowLayout)
 - JLabel jLabel1 "Elements colors"
 - JPanel jPanel7, CENTER
 - (GridLayout [2,3])
 - JPanel jPanel4
 - (FlowLayout)
 - JLabel jLabel1 "Point (default Color)"
 - JButton pointDefaultColor
- .
- .
- .
- .

C'est long et répétitif, je le ferai plus tard...

Éléments

Boutons de couleur (JButton)

Chaque bouton change la couleur de ce qui indique, appelant un JColorChooser pour sélectionner la couleur

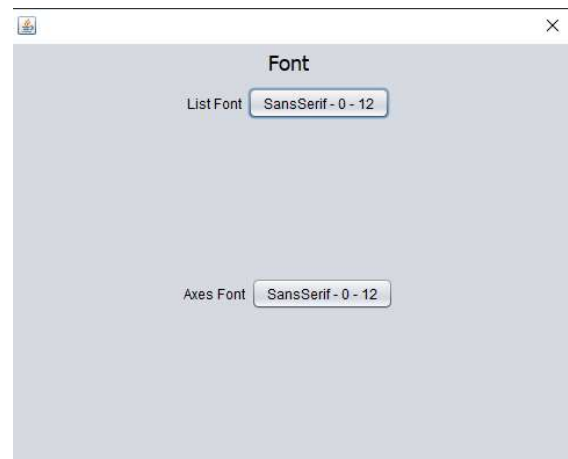
Formulaire d'édition de police (JDialog)

Ce formulaire vous permet de modifier la Police dans la fenêtre, y compris ceux de la liste des éléments et ceux de l'espace graphique.

Abol Abol

JDialog FormFonts

- ✓ (BorderLayout)
- ✓ JPanel jPanel1, NORTH
 - (FlowLayout)
 - JLabel jLabel1, "Font"
- ✓ JPanel jPanel2, CENTER
 - (GridLayout [2,1])
 - JPanel jPanel3
 - (FlowLayout)
 - JLabel jLabel2, "List Font"
 - JButton listFont "(Texto dinamico)"
 - JPanel jPanel4
 - (FlowLayout)
 - JLabel jLabel3, "Axes Font"
 - JButton axesFont "(Texto dinámico)"



Éléments

Boutons (JButton)

Ces boutons affichent un panneau qui vous permet de choisir la police, le style et la taille de la police, puis l'ajouter au composant indiqué, le texte du bouton change indiquant la police sélectionnée

Partie 2:

utilisation de JComboBox et JList

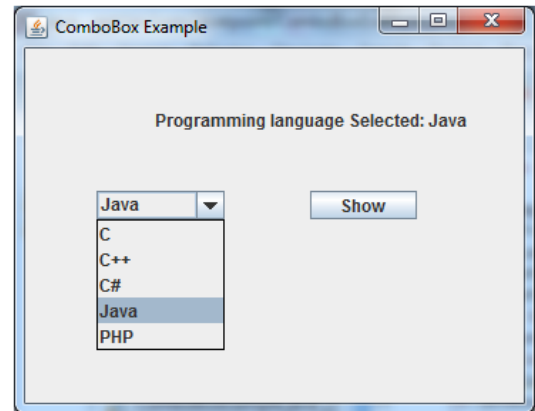
JComboBox

Fonctionnement d'une JComboBox

une JComboBox est une liste déroulante d'éléments qui vous permet de sélectionner l'un d'entre eux, par exemple:

La création d'une JComboBox se fait à l'aide de la classe JComboBox appartenant au package javax.swing. Cette classe a une surcharge de constructeur mais dans cette définition, nous nous concentrerons uniquement sur le constructeur par défaut. La création d'un JComboBox se fait avec la ligne de code suivante:

```
JComboBox miComboBox = new JComboBox();
```



Méthodes les plus importantes d'une JComboBox:

addItem(E ele): Permet d'ajouter un élément à la comboBox demande un paramètre un objet du type qui enregistre le comboBox dans notre cas sera de type String.

```
miComboBox.addItem( "Elemento 1" );
```

removeItem(E ele): permet de supprimer un élément de la comboBox en demandant un objet du type qui enregistre la comboBox.

```
miComboBox.removeItem ( "Elemento 1" );
```

removeItemAt(int index): permet de supprimer un élément de la comboBox en demandant la position de l'objet par paramètre.

```
miComboBox.removeItem ( 0 );
```

removeAllItems(): permet de supprimer tous les éléments de la comboBox

```
miComboBox.removeAllItems ();
```

getSelectedItem(): permet d'obtenir l'élément sélectionné, cette méthode retourne un objet du type qui enregistre la comboBox dans notre cas ce sera une String, mais il faut faire un casting

```
String itemSelected = (String)miComboBox.getSelectedItem ();
```

getSelectedIndex(): permet d'obtenir la position de l'élément sélectionné

```
int indexSelected = miComboBox.getSelectedIndex ();
```

setSelectedItem (E obj): Sélectionne un élément.

```
miComboBox.setSelectedItem ("Elemento 1");
```

setSelectedIndex(int index): permet de sélectionner un élément en passant sa position par paramètre.

```
miComboBox.setSelectedIndex (0);
```

getItemCount(): renvoie la quantité d'éléments contenus dans la comboBox

```
int indexSelected = miComboBox.getItemCount (0);
```

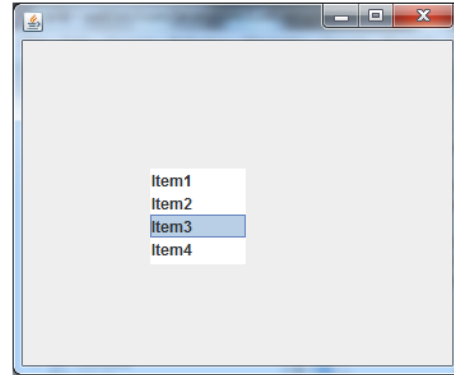
JList

Une JList est une liste d'éléments type menu, qui peuvent être sélectionnés (permet une sélection multiple) et interagir avec eux.

Pour la création d'une JList, il faut avoir un ListModel qui est responsable de déregistrer les éléments de la JList.

La création d'une JList et de son ListModel respectif se fait avec les lignes de code suivantes (le modèle par défaut sera utilisé):

```
DefaultListModel dlm = new DefaultListModel();  
JList miLista = new JList ( dlm );
```



Méthodes les plus importantes d'une JList:

setModel(ListModel mod): définir le modèle de la liste
miLista.setModel(dlm);

addElement(E ele): ajoute un élément à la listeModel
dlm.addElement ("Element 0");

remove(int index): supprimer l'élément de listModel en position «index»
dlm.remove (0);

removeAll(int index): supprimer tous les éléments de la listeModel
dlm.removeAll (0);

getSelectedIndex(): récupère la position de l'élément sélectionné dans la liste
int pos = miLista.getSelectedIndex ();

getSelectedIndices(): récupère les positions des éléments sélectionnés dans la liste
int[] pos = miLista.getSelectedIndices ();