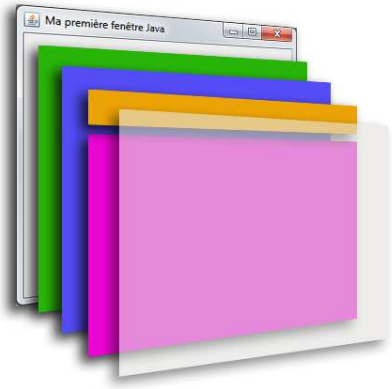
Classe JFrame (fenêtre)

- setTitle(String titre);
- setSize(int largeur, int hauteur);
- setLocation(int gauche, int haut);
- setResizable(boolean redimmensionnable);
- setVisible(boolean visible);
- setAlwaysOnTop(boolean tjrsPremierPlan);
- setDefaultCloseOperation(int operation);

Opérations sur fermeture de fenêtre

- DO_NOTHING_ON_CLOSE
  (définie dans WindowConstants)
- HIDE_ON_CLOSE
  (définie dans WindowConstants)
- DISPOSE_ON_CLOSE
  (définie dans WindowConstants)
- EXIT_ON_CLOSE (définie dans JFrame)
  à n'utiliser que pour la fenêtre principale d'une application graphique

Structure d'une fenêtre



- RootPane : le conteneur principal
- LayeredPane : un panneau avec un conteneur et une barre de menu
- MenuBar : la barre de menu, quand il y en a une
- ContentPane : le conteneur des composants de la fenêtre
- GlassPane : couche utilisée pour intercepter les actions de l'utilisateur avant qu'elles ne parviennent aux composants.

Autres fenêtres

- JWindow
  une fenêtre sans décoration
- JDialog
  une boîte de dialogue
- JFileChooser
  la boîte de dialogue prédéfinie
  pour ouvrir ou enregistrer des fichiers

Classe JPanel (Panneau)

- Conteneur pour des composants
- Zone d'affichage avec multiples possibilités de dessin applicables à son contexte graphique :
  - `setFont(Font f)`
  - `drawString(String txt, int x, int y)`
  - `setColor(Color c)`
  - `fillRect(int x, int y, int l, int h)`
  - …

  utilisées dans sa méthode `paintComponent(Graphics g)`

Dessin d'un composant

- Pour dessiner un composant **awt**, on redéfinit
  - **void paint(Graphics g)**
- Pour dessiner un composant **swing**, on peut redéfinir :
  - **void paintComponent(Graphics g)**
  - **void paint(Graphics g)**

Dessin d'un composant…

- La méthode **paint** d'un composant appelle dans l'ordre :
  - **paintComponent** qui dessine effectivement ce composant
  - **paintBorder** qui dessine les bordures
  - **paintChildren** qui dessine les composants contenus en appelant leur méthode **paint** si le composant est un conteneur

Mise à jour de l'affichage

- Pour mettre à jour l'affichage, on appelle la méthode **repaint**, qui elle appellera **paint**.
- **void repaint()** passe par un RepaintManager qui se charge de "planifier" l'affichage.

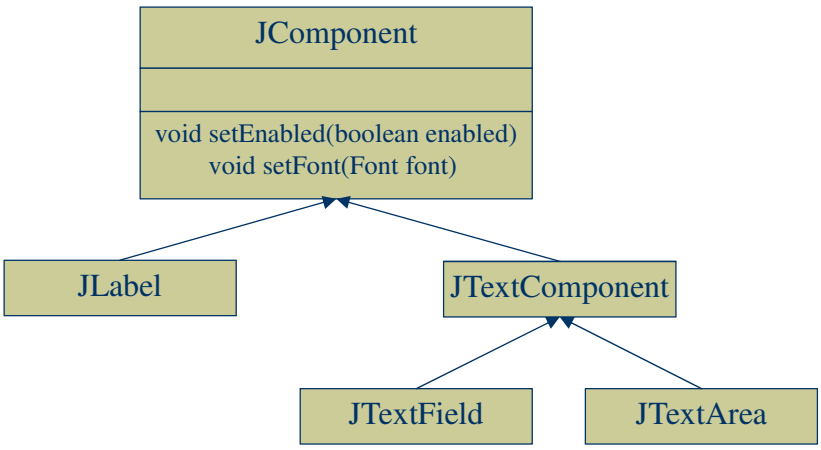
JButton

- Création avec un des constructeurs :
  - `JButton(Icon icon)`
  - `JButton(String text)`
- (Dés)Activation avec
  `setEnabled(boolean enabled)`
- Association d'une action avec
  `addActionListener(ActionListener al)`

Autres composants

Etiquettes	JLabel
Zone de texte	JTextField
Cases à cocher	JCheckBox
Bouton radio	JRadioButton
Barre de menus	JMenuBar
Menu	JMenu
Item de menu	JMenuItem

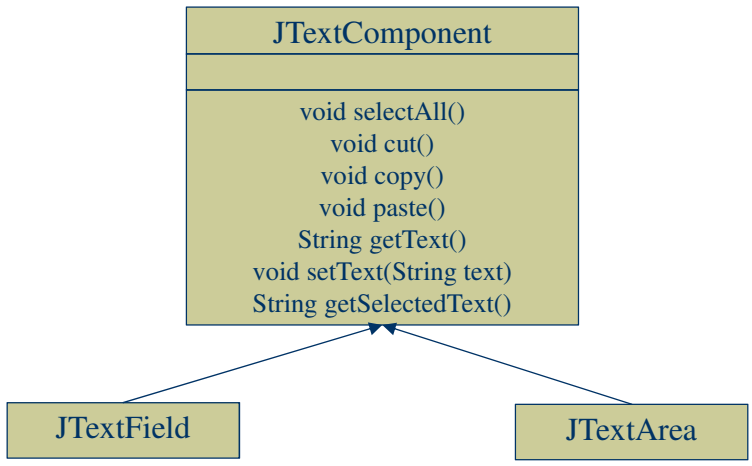
JComponent



JLabel

- Une étiquette peut afficher un texte et/ou une image
  JLabel(Icon image*, int horizontalAlignment*)
  JLabel(String text*, int horizontalAlignment*)
  JLabel(String text, Icon icon, int horizontalAlignment)
- On peut fixer l'alignement horizontal et vertical
  void **setHorizontalAlignment**(int alignment)
  void **setVerticalAlignment**(int alignment)
- On peut changer le texte ou l'image
  void **setText**(String text)
  void **setIcon**(Icon icon)

JTextComponent



JTextField & JTextArea

- ➔**JTextField**(int columns)
- ➔**JTextField**(String text, int columns)

- ➔**JTextArea**(int rows, int columns)
- ➔**JTextArea**(String text, int rows, int columns)

JFormattedTextField

- classe des zones de texte typées

➔**JFormattedTextField(**Object value)

- void setFocusLostBehavior(int)
  - COMMIT COMMIT_OR_REVERT
  - PERSIST REVERT
- Object getValue()

JSlider

- classe des jauges

➔**JSlider(**int min, int max, int value)
➔**JSlider**(int orientation, int min, int max, int value)

- void **setPaintTicks**(boolean b);
- void **setPaintLabels**(boolean b);
- void **setMinorTickSpacing**(int n)
- void **setMajorTickSpacing**(int n)
- void **setValue**(int n)
- int **getValue**()

JCheckBox

- classe des cases à cocher

➔**JCheckBox**(Icon ic, boolean sel)
➔**JCheckBox**(String txt, boolean sel)
➔**JCheckBox**(String txt, Icon ic, boolean sel)

- public boolean **isSelected**()
- public void **setSelected**(boolean b)

JRadioButton

- classe des boutons radio

➔**JRadioButton**(Icon ic, boolean sel)
➔**JRadioButton**(String txt, boolean sel)
➔**JRadioButton**(String txt, Icon ic, boolean sel)

- public boolean **isSelected**()
- public void **setSelected**(boolean b)

ButtonGroup

Conteneur d'instances de JRadioButton (et JRadioButtonMenuItem, JToggleButton) qui lorsque l'utilisateur en sélectionne une, déselectionne automatiquement les autres.

➔**ButtonGroup**()

- public void **add**(AbstractButton b)
- public void **clearSelection**()

JMenuBar

- classe des barres de menus

➔**JMenuBar**()

- JMenu **add**(JMenu menu)
- int **getMenuCount**()
- JMenu **getMenu**(int index)

JMenu

- classe des menus

➔**JMenu**(String s)

- JMenuItem **add**(JMenuItem menuItem)
- void **addSeparator**()
- int **getItemCount**()
- JMenuItem **getItem**(int pos)

JMenuItem

- classe des éléments de menu

➔**JMenuItem**(Icon icon)
➔**JMenuItem**(String text)
➔**JMenuItem**(String text, Icon icon)

- void **setEnabled**(boolean b)

JCheckBoxMenuItem

- classe des éléments de menu cochables

➔**JCheckBoxMenuItem**(Icon icon)
➔**JCheckBoxMenuItem**(String text)
➔**JCheckBoxMenuItem**(String text, Icon icon)
➔**JCheckBoxMenuItem**(String text, Icon icon, boolean b)

- void **setEnabled**(boolean b)
- void **setState**(boolean b)
- boolean **getState**()

JRadioButtonMenuItem

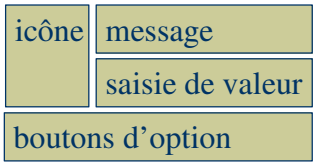
- classe des éléments de menu sélectionnables

➔**JRadioButtonMenuItem**(Icon icon, *boolean selected*)
➔**JRadioButtonMenuItem**(String text, *boolean selected*)
➔**JRadioButtonMenuItem**(String text, Icon icon)

- void **setEnabled**(boolean b)
- void **setSelected**(boolean b)

JOptionPane

- Propose des dialogues usuels par un appel en une seule ligne de code.
- Les dialogues sont tous modaux et comportent généralement 4 zones :



Méthodes de JOptionPane …

static void **showMessageDialog**(
Component parentComponent,
Object message, String title, int messageType)

static int **showConfirmDialog**(
Component parentComponent,
Object message, String title, int optionType)

static String **showInputDialog**(
Component parentComponent, Object message,
String title, int messageType)

… et leur rôle

- Les différentes méthodes prédéfinies permettent d'obtenir un dialogue fonction du type d'interaction souhaitée :

Nom de méthode	Description
showConfirmDialog	Demande une confirmation par oui/non/annuler
showInputDialog	Demande une donnée
showMessageDialog	Indique à l'utilisateur que quelquechose s'est produit.

Quelques paramètres des méthodes de JOptionPane

- **message** : en général une chaîne de caractères
- **messageType** qui peut être :
  ERROR_MESSAGE INFORMATION_MESSAGE
  WARNING_MESSAGE QUESTION_MESSAGE
  PLAIN_MESSAGE
- **optionType** qui définit les boutons affichés :
  YES_NO_OPTION YES_NO_CANCEL_OPTION
  OK_CANCEL_OPTION DEFAULT_OPTION
- **title**
- **initialValue**

Valeur retournée

- Lorsqu'elle renvoie une valeur, la méthode show…Dialog retourne l'une des valeurs :
  - YES_OPTION
  - NO_OPTION
  - CANCEL_OPTION
  - OK_OPTION
  - CLOSED_OPTION

JDialog

- Cette classe peut être dérivée pour produire des dialogues personnalisés

➔**JDialog**(Dialog owner,
String title,
boolean modal)

- protected void **dialogInit**()

JFileChooser

- classe des dialogues d'ouverture et d'enregistrement de fichiers

➔**JFileChooser**()
➔**JFileChooser**(String currentDirectoryPath)

- void setFileSelectionMode(int mode)
  FILES_ONLY DIRECTORIES_ONLY
  FILES_AND_DIRECTORIES
- void **setFileFilter**(FileFilter filter)
- int **showOpenDialog**(Component parent)
- int **showSaveDialog**(Component parent)

JFileChooser…

- les méthodes showOpenDialog et showSaveDialog peuvent renvoyer :
  - JFileChooser.CANCEL_OPTION
  - JFileChooser.APPROVE_OPTION
  - JFileChooser.ERROR_OPTION

  quand un choix est validé par l'utilisateur
  - File **getSelectedFile**()
- méthodes utiles de File
  - String **getName**()
  - String **getParent**()

FileNameExtensionFilter

- Implantation de la classe abstraite FileFilter utilisée dans les dialogues de choix de fichier pour restreindre l'affichage à certains fichiers

➔**FileNameExtensionFilter**(String description, String... extensions)