

UNIVERSITÉ ABDELMALEK ESSAADI
FACULTÉ POLYDISCIPLINAIRE DE LARACHE



Département Informatique
FILIÈRE SMI

Module : M33

Semestre : S6

Intitulé : Programmation événementielle en JAVA

PARTIE II

ASSURER PAR : PR HAIMOUDI

ANNÉE UNIVERSITAIRE 2021/2022

Présentation des packages AWT et Swing

- ▶ Deux packages permettent de gérer les interfaces graphiques.
- ▶ AWT et SWING.
- ▶ AWT utilise des composants lourds, c'est à dire utilisant les ressources du système d'exploitation.
- ▶ Swing utilise des composants dits légers n'utilisant pas ces ressources.
- ▶ Swing est plus robuste que l'AWT, plus portable, et plus facile à utiliser.
- ▶ Swing ne remplace pas complètement AWT mais fournit des composants d'interface plus performants.
- ▶ Chaque composant AWT a son équivalent Swing dans le package javax.swing.
- ▶ Nous nous intéresserons surtout aux possibilités développées dans le package swing.

Présentation des packages AWT et Swing

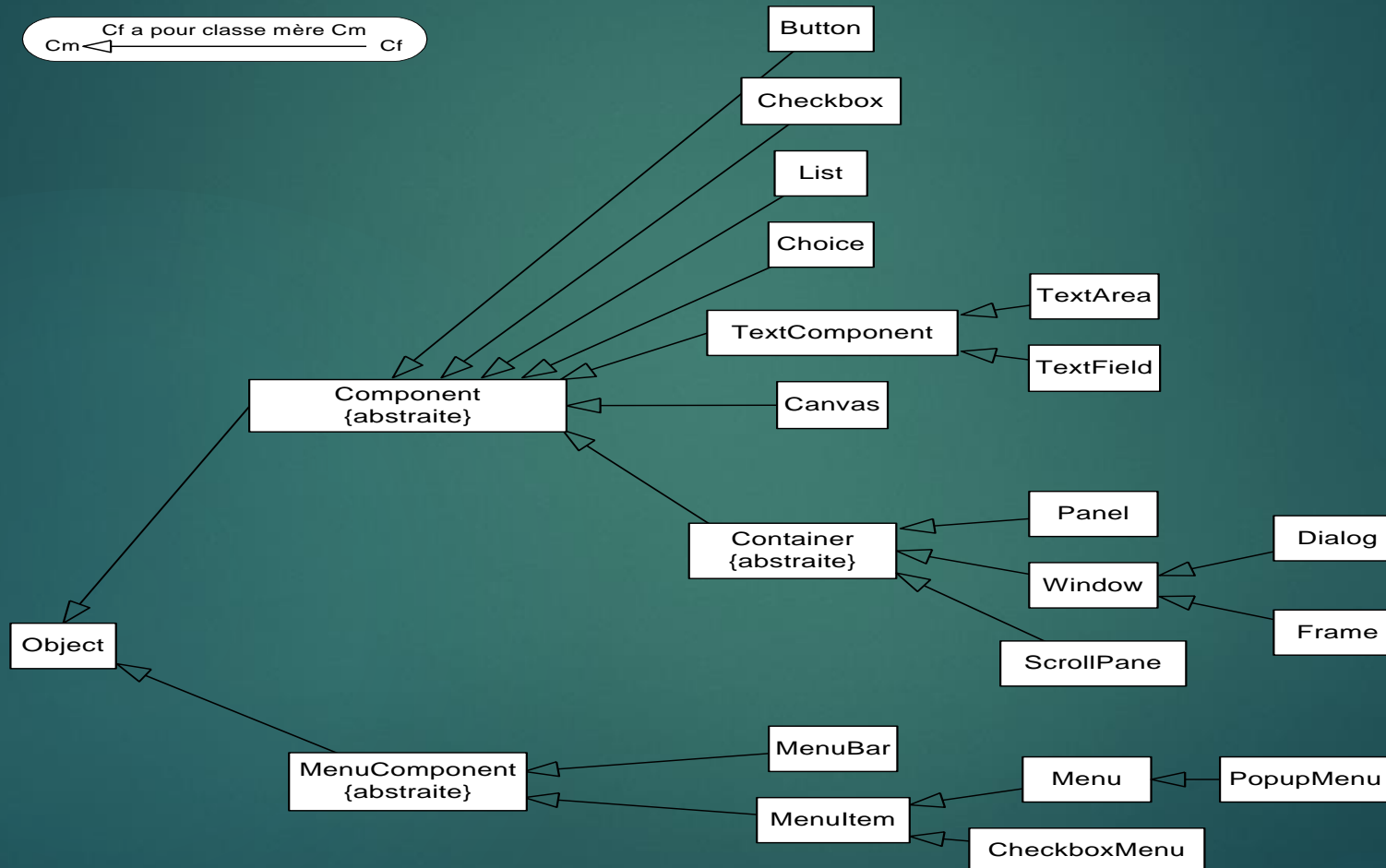
L'interface graphique avec AWT

- ▶ Les classes du toolkit AWT (Abstract Windows Toolkit) permettent d'écrire des interfaces graphiques indépendantes du système d'exploitation sur lesquelles elles vont fonctionner.
- ▶ Cette librairie utilise le système graphique de la plateforme d'exécution (Windows, MacOS, X-Window) pour afficher les objets graphiques.
- ▶ Le toolkit contient des classes décrivant les composants graphiques, les polices, les couleurs et les images.

Présentation des packages AWT et Swing

L'interface graphique avec AWT

► Les la hiérarchie des classes AWT



Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

Pour utiliser un composant, il faut créer un nouvel objet représentant le composant et l'ajouter à un de type conteneur qui existe avec la méthode `add()`.

► Exemple :

```
package ex_component;
import java.awt.event.*;
import java.awt.*;
public class Maplet extends Frame {
    public Maplet(String s){
        super(s);
        Button bt = new Button("HALLO");
        setSize(200,200);
        add(bt);
        setLayout(new FlowLayout());
        show();
    }
    public static void main(String[] args) {
        Maplet frm = new Maplet("Exfrm");
        frm.addWindowListener(new WindowAdapter(){public void windowClosing(WindowEvent e){System.exit(0);}});
    }
}
```

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- Les étiquettes.

Il faut utiliser un objet de la classe `java.awt.Label`

```
Label la = new Label( );la.setText(« une etiquette »);//  
ou Label la = new Label(« une etiquette »);
```

Il est possible de créer un objet de la classe `java.awt.Label` en précisant l'alignement du texte.

```
Label la = new Label(« etiquette », Label.RIGHT);
```

Le texte à afficher et l'alignement peuvent être modifiés dynamiquement lors de l'exécution :

```
la.setText(« nouveau texte »);  
la.setAlignment(Label.LEFT);
```

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ Il faut utiliser un objet de la classe `java.awt.Button`
- ▶ Cette classe possède deux constructeurs :

Constructeur	Rôle
<code>Button()</code>	
<code>Button(String)</code>	Permet de préciser le libellé du bouton

- ▶ Exemple:

```
Button bouton = new Button();
```

```
bouton.setLabel("bouton");// ou
```

```
Button bouton = new Button("bouton");
```

- ▶ Le libellé du bouton peut être modifié dynamiquement grâce à la méthode `setLabel()` :

```
bouton.setLabel(« nouveau libellé »);
```

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ **Les panneaux.**
- ▶ Les panneaux sont des conteneurs qui permettent de rassembler des composants et de les positionner grâce à un gestionnaire de présentation. Il faut utiliser un objet de la classe `java.awt.Panel`.
- ▶ Par défaut le gestionnaire de présentation d'un panel est de type `FlowLayout`.

Constructeur	Role
<code>Panel()</code>	Créer un panneau avec un gestionnaire de présentation de type <code>FlowLayout</code>
<code>Panel(LayoutManager)</code>	Créer un panneau avec le gestionnaire précisé en paramètre

```
Panel p = new Panel();
```

- ▶ L'ajout d'un composant au panel se fait grâce à la méthode `add()`.
`p.add(new Button("bouton"));`

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ Les listes déroulantes (combobox).
- ▶ Il faut utiliser un objet de la classe `java.awt.Choice`
- ▶ Cette classe ne possède qu'un seul constructeur qui ne possède pas de paramètres.

```
Choice maCombo = new Choice();
```

- ▶ Les méthodes `add()` et `addItem()` permettent d'ajouter des éléments à la combo.

```
maCombo.addItem("element1");// ou
```

```
maCombo.add("element2");
```

- ▶ Plusieurs méthodes permettent la gestion des sélections :

```
void select( int ); void select(String); int countItems( ); String  
getItem( int );
```

```
String getSelectedItem(); int getSelectedIndex( );
```

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

► La classe **TextComponent**.

- La classe `TextComponent` est la classe des mères des classes qui permettent l'édition de texte : `TextArea` et `TextField`.
- Elle définit un certain nombre de méthodes dont ces classes héritent.

Méthodes	Role
<code>String getSelectedText();</code>	Renvoie le texte sélectionné
<code>int getSelectionStart();</code>	Renvoie la position de début de sélection
<code>int getSelectionEnd();</code>	Renvoie la position de fin de sélection
<code>String getText();</code>	Renvoie le texte contenu dans l'objet
<code>boolean isEditable();</code>	Retourne un boolean indiquant si le texte est modifiable
<code>void select(int start, int end);</code>	Sélection des caractères situés entre start et end
<code>void selectAll();</code>	Sélection de tout le texte
<code>void setEditable(boolean b);</code>	Autoriser ou interdire la modification du texte
<code>void setText(String s);</code>	Définir un nouveau texte

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ Les champs de texte.
 - ▶ Il faut déclarer un objet de la classe `java.awt.TextField`
- ▶ Il existe plusieurs constructeurs :

Constructeurs	Role
<code>TextField();</code>	
<code>TextField(int);</code>	prédetermination du nombre de caractères à saisir
<code>TextField(String);</code>	avec texte par défaut
<code>TextField(String, int);</code>	avec texte par défaut et nombre de caractères à saisir

- ▶ Cette classe possède quelques méthodes utiles :
`String getText(); int getColumns(); void setEchoCharacter();`
`TextField tf = new TextField(10);`
`tf.setEchoCharacter('*');`
`tf.setEchoChar('*');`

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ Les zones de texte multilignes.
Il faut déclarer un objet de la classe `java.awt.TextArea`.
- ▶ Il existe plusieurs constructeurs :

Constructeur	Role
<code>TextArea()</code>	
<code>TextArea(int, int)</code>	avec prédetermination du nombre de lignes et de colonnes
<code>TextArea(String)</code>	avec texte par défaut
<code>TextArea(String, int, int)</code>	avec texte par défaut et taille

- ▶ Les principales méthodes sont :
`String getText(); String getSelectedText(); int getRows(); int getColumns();`
`void insertText(String, int); void setEditable(boolean); void appendText(String);`
`void replaceText(String, int, int);`

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ Les listes.
 - ▶ Il faut déclarer un objet de la classe `java.awt.List`.
- ▶ Il existe plusieurs constructeurs :

Constructeur	Role
<code>List()</code>	
<code>List(int)</code>	Permet de préciser le nombre de lignes affichées
<code>List(int, boolean)</code>	Permet de préciser le nombre de lignes affichées et l'indicateur de sélection multiple

- ▶ Les principales méthodes sont :

```
void addItem(String); void addItem(String, int); void delItem(int); void clear()
void delItems(int, int); void replaceItem(String, int); int countItems();

int getRows(); String getItem(int); void select(int); void deselect(int);
setMultipleSelections(boolean); int getSelectedIndex( ); int getVisibleIndex()
int[] getSelectedIndexes( ); boolean isSelected(int); String getSelectedItem( );
String[] getSelectedItems(); void makeVisible(int)
```

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ Les cases à cocher.
 - ▶ Il faut déclarer un objet de la classe `java.awt.Checkbox`
- ▶ Il existe plusieurs constructeurs :

Constructeur	Role
<code>Checkbox()</code>	
<code>Checkbox(String)</code>	avec une étiquette
<code>Checkbox(String,boolean)</code>	avec une étiquette et un état
<code>Checkbox(String,CheckboxGroup,boolean)</code>	avec une étiquette, dans un groupe de cases à cocher et un état

- ▶ Les principales méthodes sont :
`void setLabel(String); void setState(boolean); boolean getState();`
`String getLabel();`

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ Les boutons radio.
 - ▶ Déclarer un objet de la classe `java.awt.CheckboxGroup`.

```
CheckboxGroup rb;  
Checkbox cb1 = new Checkbox(« etiquette 1 », rb, etat1_boolean);  
Checkbox cb2 = new Checkbox(« etiquette 2 », rb, etat1_boolean);  
Checkbox cb3 = new Checkbox(« etiquette 3 », rb, etat1_boolean);
```

- ▶ Les principales méthodes sont :

Méthodes	Role
Checkbox getCurrent()	retourne l'objet Checkbox correspondant à la réponse sélectionnée il faut utiliser la méthode getSelectedCheckbox()
void setCurrent(Checkbox)	Coche le bouton radio passé en paramètre il faut utiliser la méthode setSelectedCheckbox()

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

► La classe Canvas.

- C'est un composant sans fonction particulière : il est utile pour créer des composants graphiques personnalisés.
- Il est nécessaire d'étendre la classe Canvas pour en redéfinir la méthode `Paint()`

```
syntaxe : =  
import java.awt.*;  
public class MonCanvas extends  
Canvas {  
    public void paint(Graphics g) {  
        g.setColor(Color.black);  
        g.fillRect(10, 10, 100,50);  
        g.setColor(Color.green);  
        g.fillOval(40, 40, 10,10);  
    }  
}
```

```
import java.applet.*;  
import java.awt.*;  
public class AppletButton extends  
Applet {  
    MonCanvas mc = new  
MonCanvas();  
    public void paint(Graphics g){  
        super.paint(g);  
        mc.paint(g);  
    }  
}
```


Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ Les conteneurs.
 - ▶ Les conteneurs sont des objets graphiques qui peuvent contenir d'autres objets graphiques, incluant éventuellement des conteneurs. Ils héritent de la classe Container.
- ▶ Un composant graphique doit toujours être incorporé dans un conteneur :

Conteneur	Rôle
Panel	conteneur sans fenêtre propre. Utile pour ordonner les contrôles
Window	fenêtre principale sans cadre ni menu. Les objets descendants de cette classe peuvent servir à implémenter des menus
Dialog (descendant de Window)	réaliser des boîtes de dialogue simples
Frame (descendant de Window)	classe de fenêtre complètement fonctionnelle
Applet (descendant de Panel)	pas de menu. Pas de boîte de dialogue sans être incorporée dans une classe Frame.

- ▶ L'insertion de composant dans un conteneur se fait grâce à la méthode `add(Component)` de la classe Container.

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ Le conteneur Panel.
 - ▶ C'est essentiellement un objet de rangement pour d'autres composants.
- ▶ La classe Panel possède deux constructeurs :

Constructeur	Role
Panel()	
Panel(LayoutManager)	Permet de préciser un layout manager

```
Panel p = new Panel( );  
  
Button b = new Button(« bouton »);  
  
p.add( b);
```

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

- ▶ Le conteneur Frame.
 - ▶ Ce conteneur permet de créer des fenêtres d'encadrement.
 - ▶ Il hérite de la classe Window qui ne s'occupe que de l'ouverture de la fenêtre.
 - ▶ Window ne connaît pas les menus ni les bordures qui sont gérés par la classe Frame.
 - ▶ Dans une applet, elle n'apparaît pas dans le navigateur mais comme une fenêtre indépendante.

Constructeur	Role
Frame()	Exemple : <code>Frame f = new Frame();</code>
Frame(String)	Precise le nom de : <code>Frame f = new Frame(« titre »);</code>

- ▶ Les principales méthodes sont :

Méthodes	Role
setCursor(int)	changer le pointeur de la souris dans la fenêtre Exemple : f.setCursor(Frame.CROSSHAIR_CURSOR); utiliser la méthode setCursor(Cursor).
int getCursorType()	déterminer la forme actuelle du curseur utiliser la méthode getCursor().
Image getIconImage()	déterminer l'icône actuelle de la fenêtre
MenuBar getMenuBar()	déterminer la barre de menus actuelle
String getTitle()	déterminer le titre de la fenêtre
boolean isResizable()	déterminer si la taille est modifiable
void remove(MenuComponent)	Supprimer un menu
void setIconImage(Image);	définir l'icône de la fenêtre
void setMenuBar(MenuBar)	Définir la barre de menu
void setResizable(boolean)	définir si la taille peut être modifiée
void setTitle(String)	définir le titre de la fenêtre

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Les composants graphiques

► Les menus.

- Il faut insérer les menus dans des objets de la classe Frame (fenêtre d'encadrement).
- Il faut créer une barre de menu et l'affecter à la fenêtre d'encadrement.
- Il faut ensuite créer les entrées de chaque menu et les rattacher à la

```
import java.awt.*;  
public class MaFrame extends Frame {  
    public MaFrame() {  
        super();  
        setTitle(« Titre de la Fenetre » );  
        setSize(300, 150);  
        MenuBar mb = new MenuBar();  
        setMenuBar(mb);  
        Menu m = new Menu("un menu");  
        mb.add(m);  
        m.add(new MenuItem(" 1er element "));  
        m.add(new MenuItem(" 2eme element "));  
        Menu m2 = new Menu("sous menu");  
        CheckboxMenuItem cbm1 = new
```

```
CheckboxMenuItem(" menu item 1.3.1 ");  
m2.add(cbm1);  
        cbm1.setState(true);  
        CheckboxMenuItem cbm2 = new  
        CheckboxMenuItem(" menu item 1.3.2 ");  
        m2.add(cbm2);    m.add(m2);  
        pack();  
        show(); // affiche la fenetre    }  
  
        public static void main(String[] args) {  
            new MaFrame();  
        }  
    }  
}
```

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Positionnement de composants

Les *layout managers*, plus ou moins simples à utiliser, dont le rôle est de gérer la position des éléments sur la fenêtre. Tous ces layout managers se trouvent dans le package java.awt.

- ▶ il existe plusieurs sortes de *layout managers*.
 - ▶ **BorderLayout** : Pratique si vous voulez placer vos composants de façon simple par rapport à une position cardinale de votre conteneur.
 - ▶ **GridLayout** : permet d'ajouter des composants suivant une grille définie par un nombre de lignes et de colonnes. Les éléments sont disposés à partir de la case située en haut à gauche. Dès qu'une ligne est remplie, on passe à la suivante.
 - ▶ **BoxLayout** : ranger les composants à la suite soit sur une ligne, soit sur une colonne.
 - ▶ **CardLayout** : gérer vos conteneurs comme un tas de cartes (les uns sur les autres), et basculer d'un contenu à l'autre en deux temps, trois clics. Le principe est d'assigner des conteneurs au layout en leur donnant un nom afin de les retrouver plus facilement, permettant de passer de l'un à l'autre sans effort.
 - ▶ **GridBagLayout** : ce layout se présente sous la forme d'une grille à la façon d'un tableau Excel : vous devez positionner vos composants en vous servant des coordonnées des cellules (qui sont définies lorsque vous spécifiez leur nombre).

Présentation des packages AWT et Swing

L'interface graphique avec AWT

Positionnement de composants

► Exemple : **BorderLayout**.

```
import java.awt.*;
public class MaFrame extends Frame {
    public MaFrame() {
        super();
        setTitle("Titre de la Fenetre ");
        setSize(300, 150);
        setLayout(new BorderLayout());
        add("North", new Button(" bouton haut "));
        add("South", new Button(" bouton bas "));
        add("West", new Button(" bouton gauche "));
        add("East", new Button(" bouton droite "));
        add("Center", new Button(" bouton milieu "));
        pack();
        show(); // affiche la fenetre
    }
    public static void main(String[] args) {
        new MaFrame();
    }
}
```

