

Programmation Orientée Objet Java



Licence 2 Informatique Tours 2018-2019

Nizar Messai

nizar.messai@univ-tours.fr



Contenu du module

- Notions de base de la POO
 - Classe, objet, encapsulation, h ritage
- Abstraction et polymorphisme
 - Classe abstraite, interface, polymorphisme, liaison dynamique
- G n ricit 
 - Types g n riques, classes g n riques
- Gestion des exceptions
- Structures de donn es
 - Collections, listes, tables de hachage
- Gestions des flux : fichiers et bases de donn es
- Interface graphique et programmation  v nementielle (JavaFX)

Aujourd'hui ...

7

Interface graphique

Programmation événementielle
JavaFX

Intervenant : Maxime Bourreau



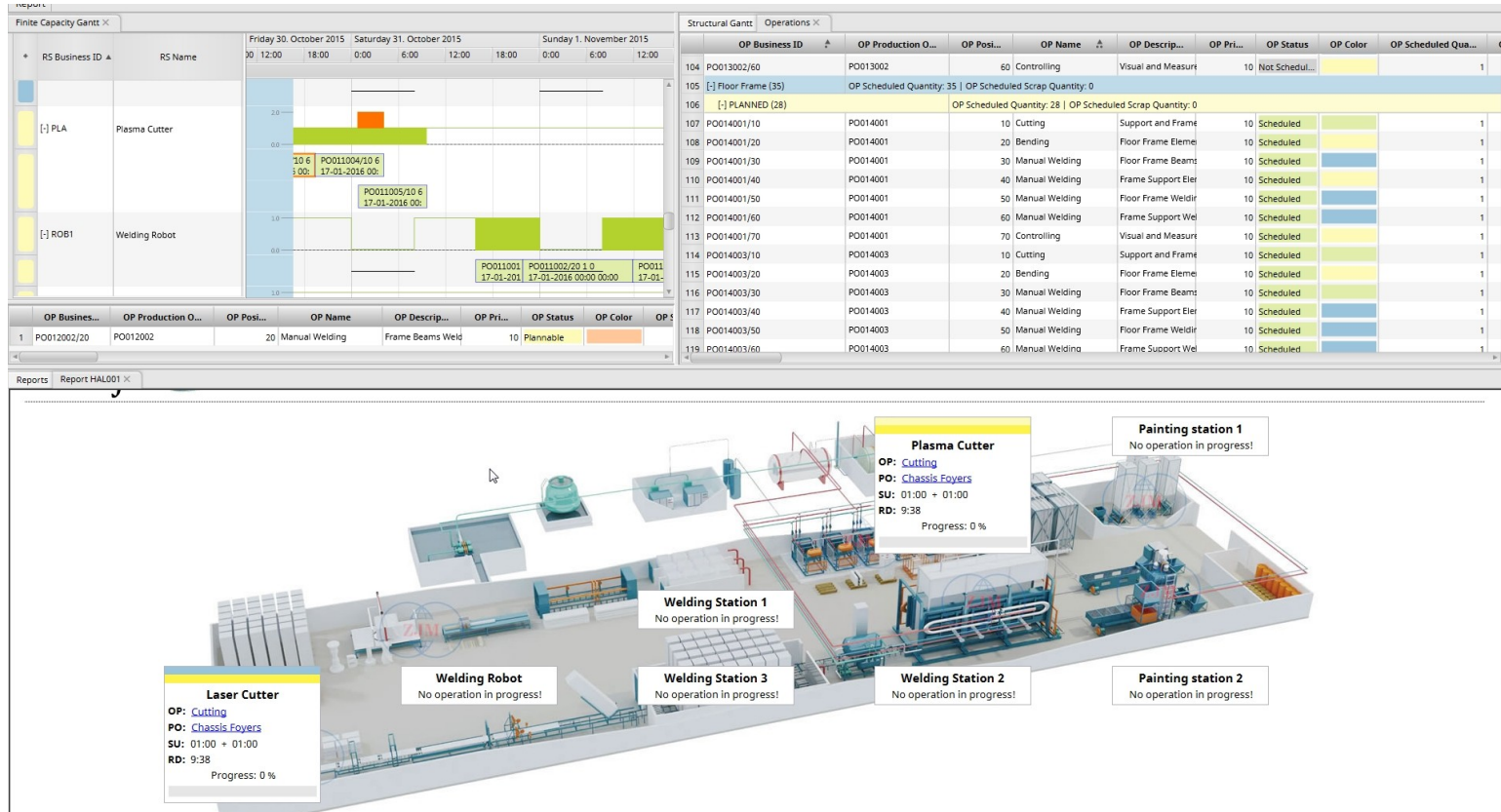


Pourquoi  tudier la programmation des interfaces graphiques ?

- Omnipr sentes
 - Ordinateur, smartphone
 - Informatique embarqu e, objets connect s
- Revoir les notions de POO
- Aborder d'autres notions : design patterns, programmation syst me et r seau



Pourquoi étudier la programmation des interfaces graphiques ?





Historique

- 1963 : Sketchpad (*Ivan Sutherland*)
 - Logiciel de dessin technique et artistique
 - Utilisation d'un stylet
- 1972 : Smalltalk (*Alan Kay, Dan Ingals, Adele Goldberg*)
 - Langage de programmation orienté objet et environnement de développement graphique
- 1984/85 : Diffusion des interfaces graphiques (Macintosh, Microsoft Windows, Atari ST, etc...)
- 1990 : World Wide Web (*Tim Berners-Lee, Robert Cailliau*)



Librairies d'interfaces graphiques de Java

Plusieurs g n rations technologiques :

- AWT (1995)
- SWING (1996)
- SWT (2003, d velopp  par IBM pour eclipse)
- JavaFX 1 (2008)
- JavaFX 2 (2011)

Le cours va porter sur JavaFX 8 ( volution de JavaFX 2) : sur internet, ne prenez pas d'informations sur JavaFX 1, JavaFXScript ou FXScript

Attention   l'autocompl tion


Biblioth ques d'interfaces graphiques de Java

Attention   l'autocompl tion

```
Button b = new Button();
```

 Button cannot be resolved to a type

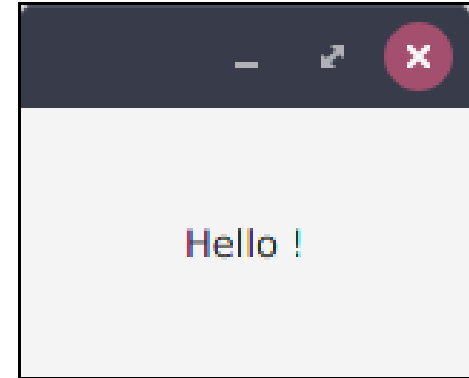
10 quick fixes available:

- ⌚ [Import 'Button' \(javafx.scene.control\)](#) 
- ⌚ [Import 'Button' \(java.awt\)](#)
- Ⓞ [Create class 'Button'](#)
- ➡ [Change to 'ButtonBar' \(javafx.scene.control\)](#)
- ➡ [Change to 'ButtonBase' \(javafx.scene.control\)](#)
- ➡ [Change to 'ButtonGroup' \(javax.swing\)](#)
- ➡ [Change to 'ButtonSkin' \(com.sun.javafx.scene.control.skin\)](#)
- ➡ [Change to 'ButtonType' \(javafx.scene.control\)](#)



Premier exemple

- Initialisation de JavaFX dans un programme Java
- Affichage d'une fen tre





```
public class Main extends Application {
```

```
    public static void main(String[] args) {
        Application.launch(args);
    }
```

```
@Override
```

```
public void start(Stage primaryStage) {
```

```
    }
}
```

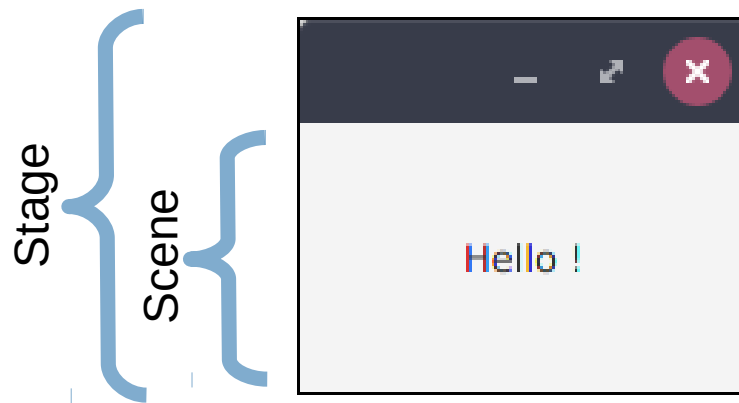
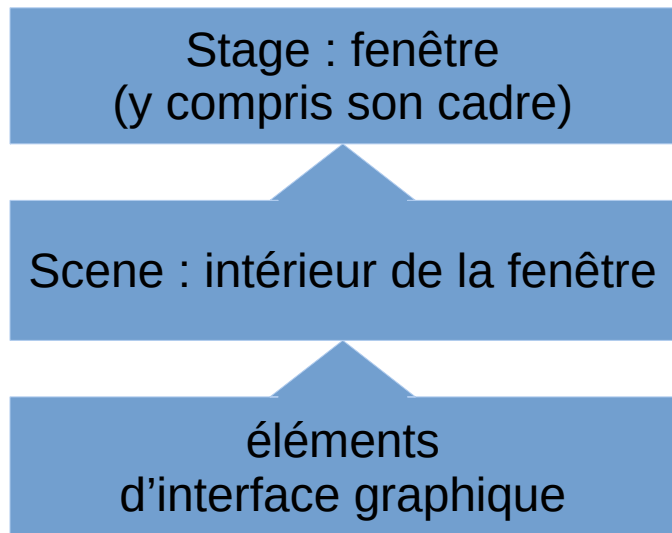
Javafx.application.Application

Point d'entr e Java

Point d'entr e JavaFX



Affichage d'une fenêtre





```
public class Main extends Application {
```

```
    public static void main(String[] args) {
        Application.launch(args);
    }
```

```
@Override
```

```
public void start(Stage primaryStage) {
```

```
    Label label = new Label("Hello !");
    label.setAlignment(Pos.CENTER);
    Scene scene = new Scene(label, 300, 250);
```

```
    primaryStage.setScene(scene);
    primaryStage.show();
```

```
    }
}
```

Fen tre principale

Label ( tiquette texte)

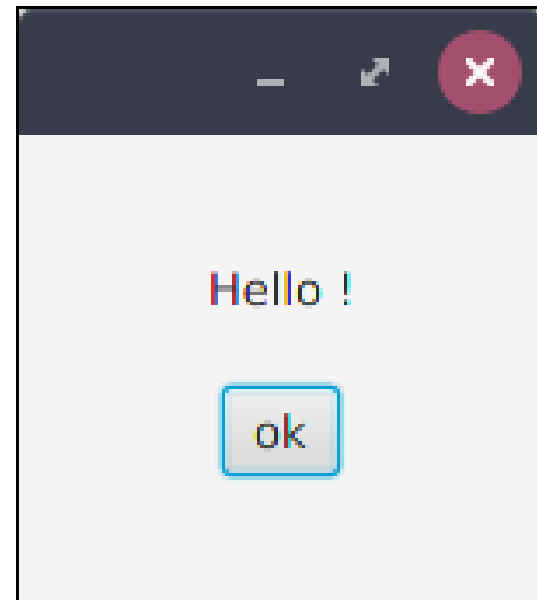
Dimensions int rieures

Scene
(int rieur d'une fen tre)



Deuxi me exemple

- Plusieurs  l ments d'interface graphique
- Gestion d'un clic sur un bouton

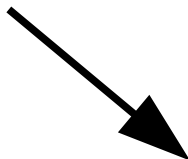




Deuxi me exemple

Comment placer plusieurs  l ments dans l'interface graphique ?

```
Label label = new Label("Hello !");
```



Le constructeur Scene ne peut
recevoir qu'un seul  l ment

```
Scene scene = new Scene(label, 300, 250);
```

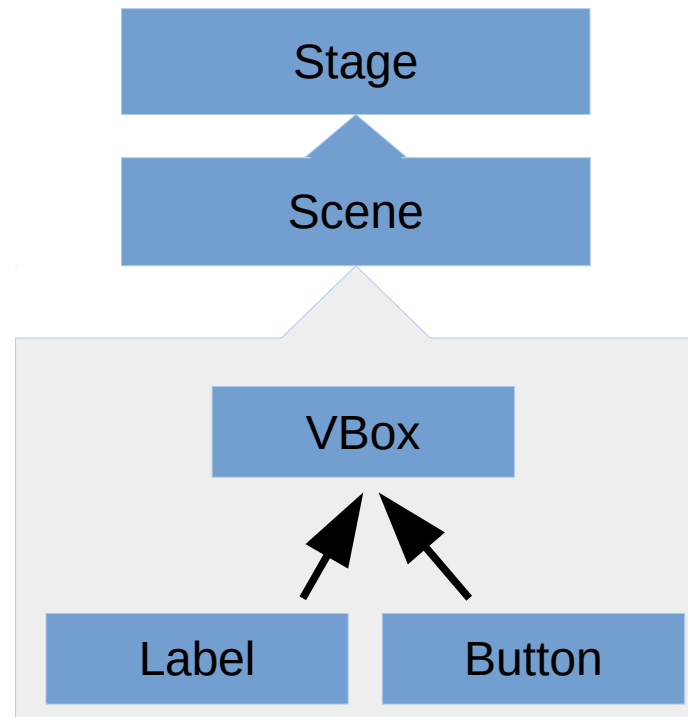
```
primaryStage.setScene(scene);  
primaryStage.show();
```



Deuxi me exemple

Comment placer plusieurs  l ments dans l'interface graphique ?

```
public void start(Stage primaryStage) {  
  
    VBox vbox = new VBox();  
  
    Label label1 = new Label("Bonjour");  
    vbox.getChildren().add(label1);  
  
    Button btn1 = new Button("ok");  
    vbox.getChildren().add(btn1);  
  
    Scene scene = new Scene(vbox, 300, 250);  
    primaryStage.setScene(scene);  
    primaryStage.show();  
}
```





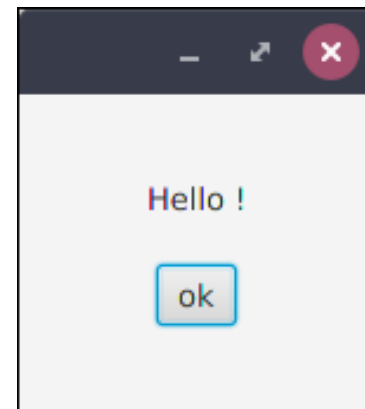
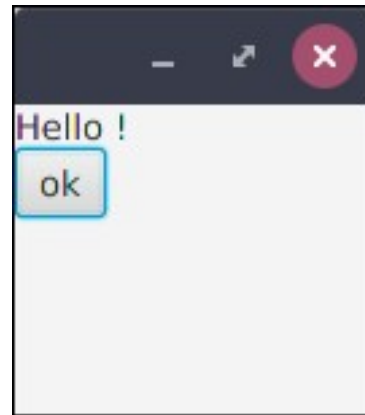
Deuxi me exemple

Param trage de la disposition VBox

```
public void start(Stage primaryStage) {
    VBox vbox = new VBox();
    vbox.setAlignment(Pos.CENTER);
    vbox.setSpacing(20);
    Label label1 = new Label("Bonjour");
    vbox.getChildren().add(label1);

    Button btn1 = new Button("ok");
    btn1.setOnAction(new HandlerBtn1());
    vbox.getChildren().add(btn1);

    Scene scene = new Scene(vbox, 300, 250);
    primaryStage.setScene(scene);
    primaryStage.show();
}
```





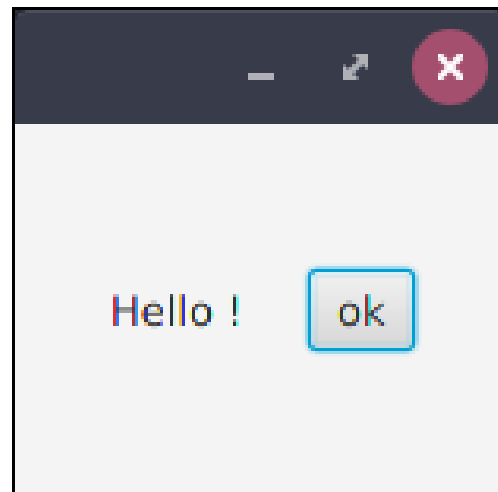
Deuxi me exemple

Autre exemple de disposition : HBox (r partition horizontale)

```
public void start(Stage primaryStage) {
    HBox hbox = new HBox();
    hbox.setAlignment(Pos.CENTER);
    hbox.setSpacing(20);
    Label label1 = new Label("Bonjour");
    hbox.getChildren().add(label1);

    Button btn1 = new Button("ok");
    btn1.setOnAction(new HandlerBtn1());
    hbox.getChildren().add(btn1);

    Scene scene = new Scene(hbox, 300, 250);
    primaryStage.setScene(scene);
    primaryStage.show();
}
```





Deuxi me exemple

Autre exemple de disposition : GridPane

```
GridPane gridPane = new GridPane();
gridPane.setAlignment(Pos.CENTER);
gridPane.setHgap(30);
gridPane.setVgap(15);

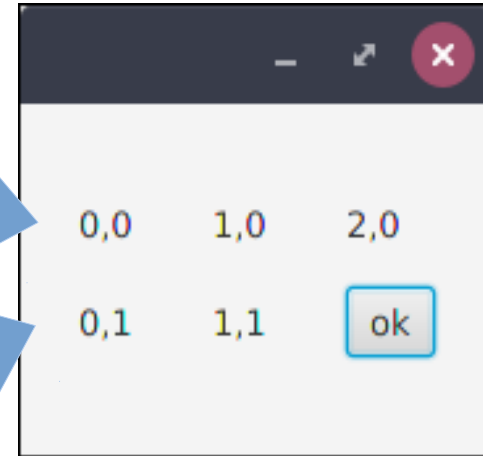
gridPane.add(new Label("0,0"), 0, 0);
gridPane.add(new Label("1,0"), 1, 0);
gridPane.add(new Label("2,0"), 2, 0);

gridPane.add(new Label("0,1"), 0, 1);
gridPane.add(new Label("1,1"), 1, 1);

Button btn1 = new Button("ok");
gridPane.add(btn1, 2, 1);
```

Ligne 0

Ligne 1



Il n'est pas n cessaire de remplir toutes les cellules



Principales dispositions

Packages `javafx.scene.layout.*`



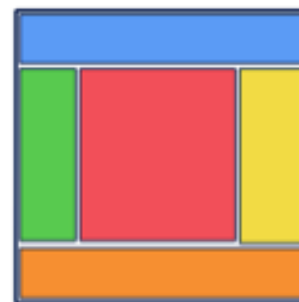
VBox



TilePane



GridPane



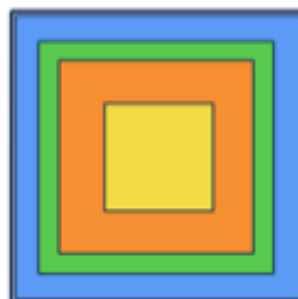
BorderPane



HBox



FlowPane



StackPane



AnchorPane



Gestion d'un clic sur un bouton

```
Button btn1 = new Button("ok");

btn1.setOnAction(
);

vbox.getChildren().add(btn1);
```

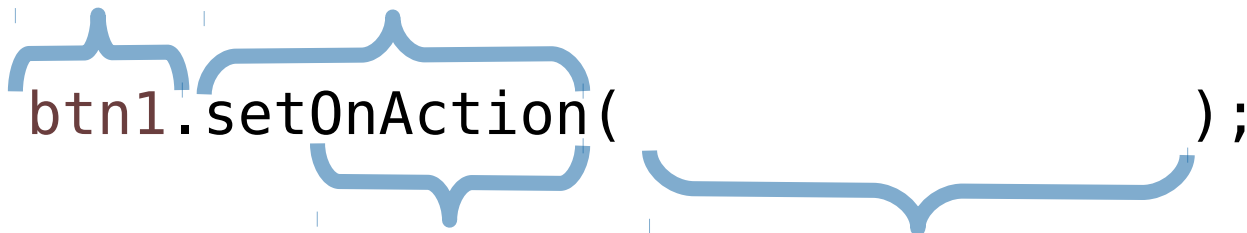


Gestion d'un clic sur un bouton

Element recevant le clic

Méthode pour déclarer le gestionnaire
de l'évènement `onAction`

```
btn1.setOnAction( );
```



une méthode pour chaque
type d'évènement

objet gestionnaire de l'évènement



Gestion d'un clic sur un bouton

```
public void start(Stage primaryStage) {
    VBox vbox = new VBox();
    vbox.setAlignment(Pos.CENTER);
    vbox.setSpacing(20);
    Label label1 = new Label("Bonjour");
    vbox.getChildren().add(label1);
    Button btn1 = new Button("ok");
    btn1.setOnAction(
    vbox.getChildren().add(btn1);
    Scene scene = new Scene(vbox, 300, 250);
    primaryStage.setScene(scene);
    primaryStage.show();
}
```


objet gestionnaire de l'évènement

doit implémenter l'interface `EventHandler<ActionEvent>`



Gestion d'un clic sur un bouton

```
public void start(Stage primaryStage) {
    VBox vbox = new VBox();
    vbox.setAlignment(Pos.CENTER);
    vbox.setSpacing(20);
    Label label1 = new Label("Bonjour");
    vbox.getChildren().add(label1);
    Button btn1 = new Button("ok");
    btn1.setOnAction(new HandlerBtn1());
    vbox.getChildren().add(btn1);
    Scene scene = new Scene(vbox, 300, 250);
    primaryStage.setScene(scene);
    primaryStage.show();
}
```



```
class HandlerBtn1 implements EventHandler<ActionEvent>
{
    @Override
    public void handle(ActionEvent event) {
        System.out.println("clic sur ok");
    }
}
```



Gestion d'un clic sur un bouton

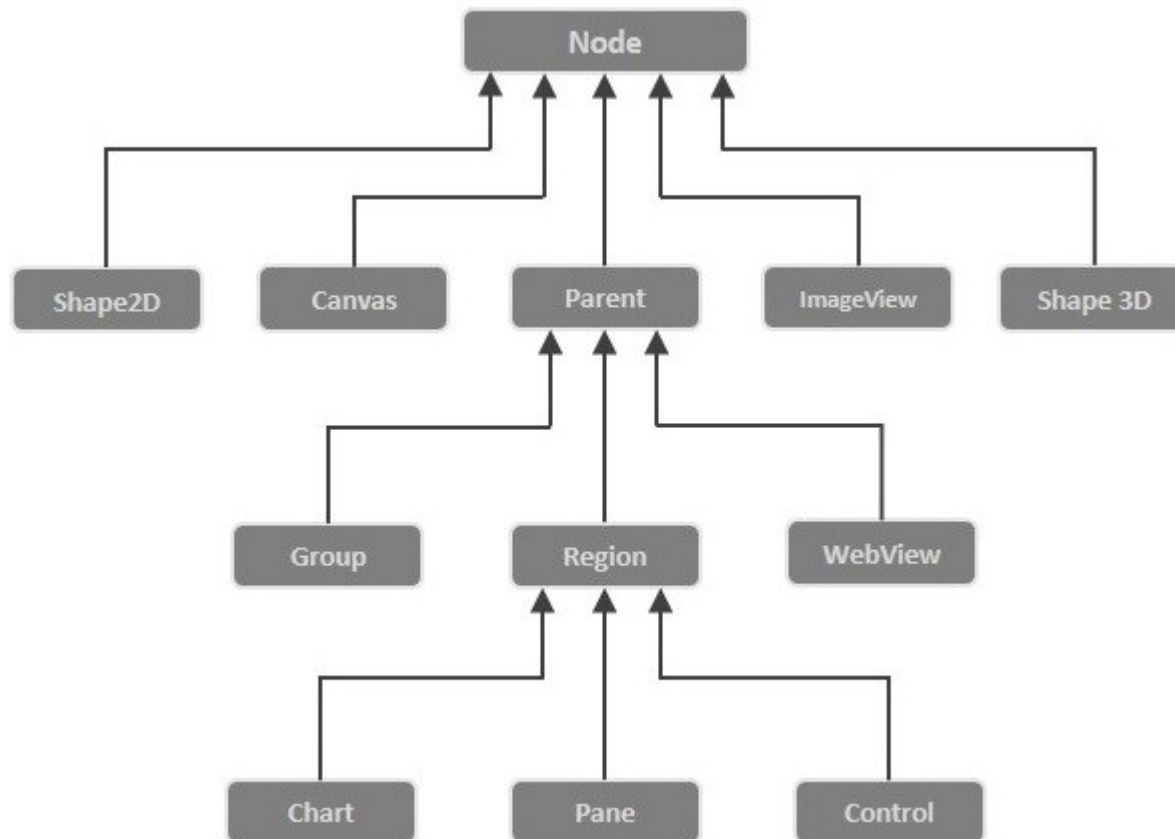
```
btn1.setOnAction(new HandlerBtn1());
class HandlerBtn1 implements EventHandler<ActionEvent> {
    @Override
    public void handle(ActionEvent event) {
        System.out.println("clic sur ok");
    }
}
```

Utilisation
d'une classe
anonyme

```
btn1.setOnAction(
    new EventHandler<ActionEvent>() {
        @Override
        public void handle(ActionEvent event) {
            System.out.println("clic sur ok");
        }
    }
);
```




Elements graphiques de JavaFX

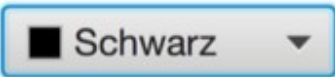
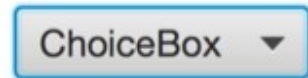




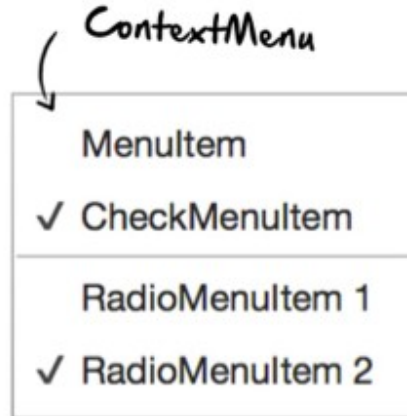
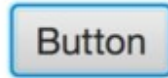
Principaux contrôles de JavaFX (1/3)



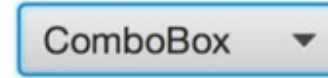
Accordion



ColorPicker



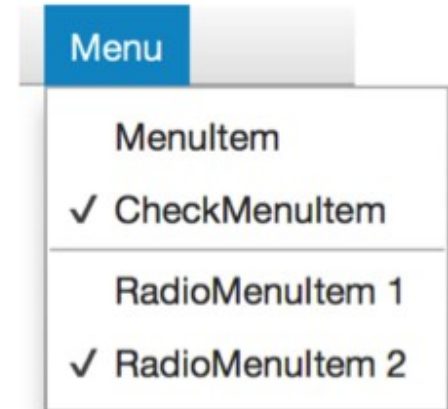
ContextMenu



DatePicker



Label





Principaux contrôles de JavaFX (2/3)



1/10

Pagination



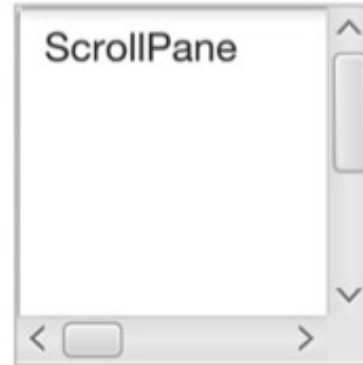
PasswordField



ProgressIndicator



ProgressBar



SplitPane

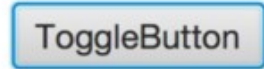


Principaux contrôles de JavaFX (3/3)

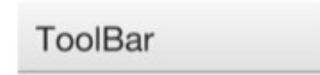
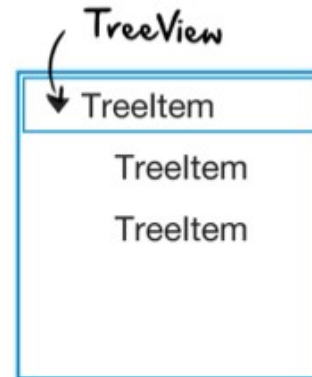


TableColumn	Tab
TableCell	Ta
TableCell	Ta

Table



Some Node
with a tooltip



TreeTableView

TableColumn	Tab
▼ TreeTableCell	Tree
TreeTableCell	Tree
TreeTableCell	Tree



Autres éléments de JavaFX

Graphiques



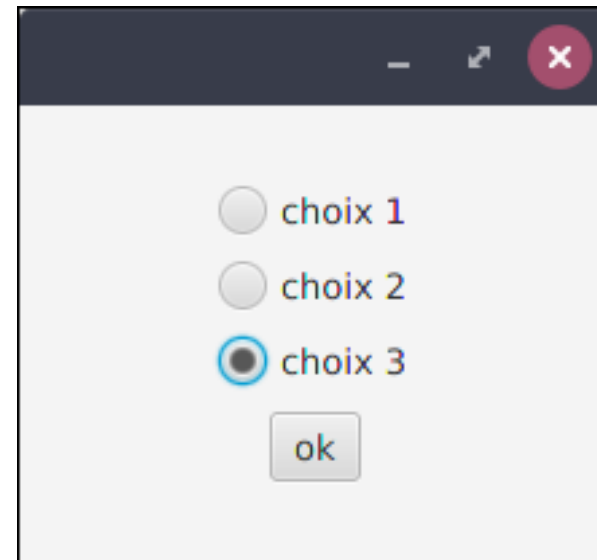
Stock Monitoring, 2010





3eme exemple

- Boutons radios
- Ne doit autoriser qu'un seul choix



choix 1

choix 2

choix 3

ok



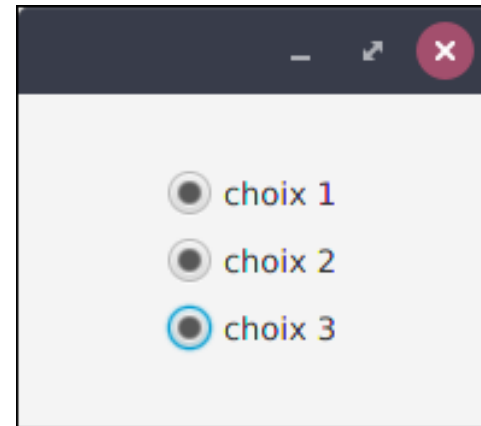
3eme exemple

```
RadioButton rb1 = new RadioButton("choix 1");
vbox.getChildren().add(rb1);
RadioButton rb2 = new RadioButton("choix 2");
vbox.getChildren().add(rb2);
RadioButton rb3 = new RadioButton("choix 3");
vbox.getChildren().add(rb3);
```

Pour qu'un seul choix reste actif :

- Déclarer un ToggleGroup
- Lier chaque RadioButton à ce ToggleGroup

```
ToggleGroup tg1 = new ToggleGroup();
rb1.setToggleGroup(tg1);
rb2.setToggleGroup(tg1);
rb3.setToggleGroup(tg1);
```





3eme exemple

Conna tre la valeur choisie : r f rence de l'objet RadioButton

```
Button btnOk = new Button("ok");
vbox.getChildren().add(btnOk);

btnOk.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        System.out.println(tg1.getSelectedToggle());
    }
});
```



RadioButton@2ad87d4d[styleClass=radio-button]'choix 3'



3eme exemple

Connaître la valeur choisie : Valeur attachée à chaque RadioButton

```
RadioButton rb1 = new RadioButton("choix 1");
rb1.setUserData(1);
RadioButton rb2 = new RadioButton("choix 2");
rb2.setUserData(2);
RadioButton rb3 = new RadioButton("choix 3");
rb3.setUserData(3);
```

Type d'objet quelconque

```
btnOk.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        Toggle t = tgl.getSelectedToggle();
        System.out.println("Valeur sélectionnée = " + t.getUserData());
    }
});
```

Valeur sélectionnée = 3



4eme exemple

- Champ de saisie



4eme exemple

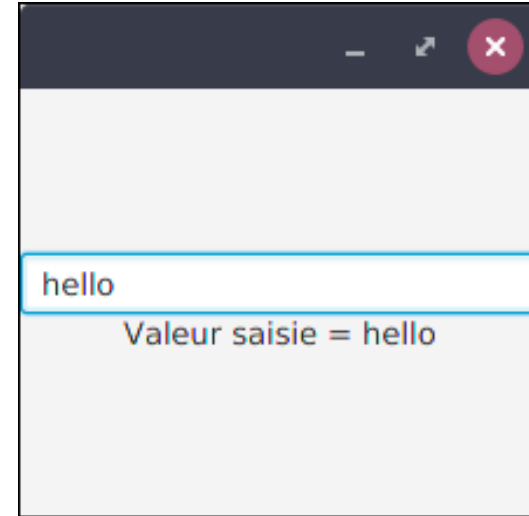
```
TextField textField = new TextField("");
Label label = new Label("");

VBox vbox = new VBox();
vbox.setAlignment(Pos.CENTER);

vbox.getChildren().addAll(textField, label);
```



Ajout de plusieurs Node   la fois



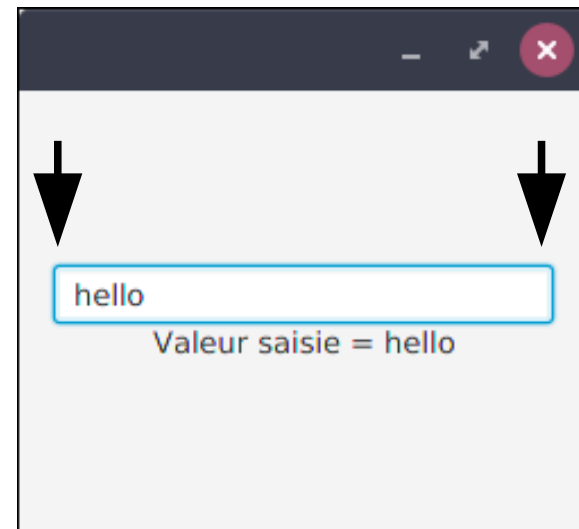


4eme exemple

```
TextField textField = new TextField("");
Label label = new Label("");

VBox vbox = new VBox();
vbox.setAlignment(Pos.CENTER);
vbox.setPadding(new Insets(0, 15, 0, 15));
vbox.getChildren().addAll(textField, label);
```

Marges int rieures
« inside offsets »





4eme exemple

```
textField.textProperty().addListener(
    new ChangeListener<String>() {
        @Override
        public void changed(
            ObservableValue<? extends String> observable,
            String oldValue,
            String newValue) {

            if (!newValue.isEmpty()) {
                label1.setText("Valeur saisie = " + newValue);
            } else {
                label1.setText("");
            }
        }
    }
);
```



4eme exemple

```
textField.textProperty().addListener(
    new ChangeListener<String>() {
        @Override
        public void changed(
```

Objet de type javafx.beans.property.StringProperty

- valeur saisie dans le champ texte
- chaque changement d clenche un appel de m thode

```
        label1.setText("Valeur saisie = " + newValue);
    } else {
        label1.setText("");
    }
}
);
```



4eme exemple

```
textField.textProperty().addLi
```

```
new ChangeListener<String>() {
```

```
@Override
```

```
public void changed(
```

```
ObservableValue<? extends String> observable,
```

```
String oldValue,
```

```
String newValue) {
```

```
if (!newValue.isEmpty()) {
```

```
label1.setText("Valeur saisie");
```

```
} else {
```

```
label1.setText("");
```

```
}
```

```
}
```

```
}
```

```
);
```

M thode appel e   chaque changement

Valeur avant
le dernier changement

Valeur la plus r cente



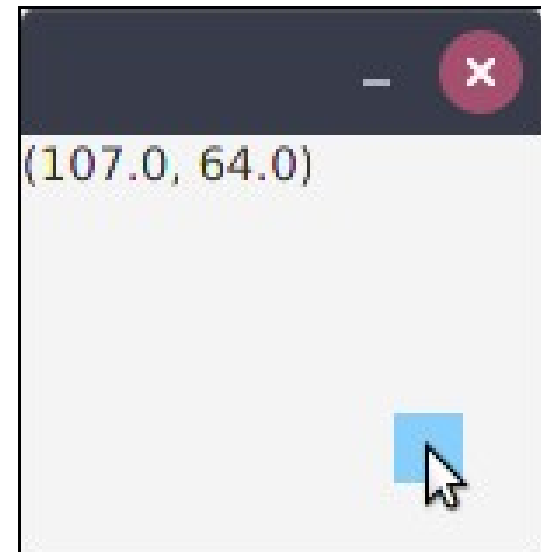
5eme exemple

Dessin et d placement d'une forme g om trique

Source et cible d'un  v nement

Cycle de vie complet d'un programme JavaFX

Boite de dialogue





5eme exemple

```
BorderPane borderPane = new BorderPane();
```

```
Pane pane = new Pane();
```

```
Label label = new Label();
```

```
borderPane.setTop(label);
```

```
borderPane.setCenter(pane);
```

```
double w = 20;
```

```
Rectangle rect = new Rectangle(w, w, Color.LIGHTSKYBLUE);
```

```
pane.getChildren().add(rect);
```



5eme exemple

Comment d placer le rectangle avec la souris ?

```
rect.setOnMouseDragged(new EventHandler<MouseEvent>() {
    public void handle(MouseEvent event) {
        double x = event.getSceneX() - pane.getLayoutX();
        double y = event.getSceneY() - pane.getLayoutY();
        label.setText("(" + x + ", " + y + ")");
        if (x < w/2) x = w/2;
        if (x > pane.getWidth()-w/2) x = pane.getWidth()-w/2;
        if (y < w/2) y = w/2;
        if (y > pane.getHeight()-w/2) y = pane.getHeight()-w/2;
        rect.setTranslateX(x - w/2);
        rect.setTranslateY(y - w/2);
    }
});
```



5eme exemple

Comment d placer le rectangle avec la souris ?

```
rect.setOnMouseDragged(new EventHandler<MouseEvent>() {
    public void handle(MouseEvent event) {
        {
            double x = event.getSceneX() - pane.getLayoutX();
            double y = event.getSceneY() - pane.getLayoutY();
        }
    }
});
```

Position de la souris relative au pane =
position de la souris relative   la scene - position du pane relative   la scene

```
});
```



5eme exemple

Comment d placer le rectangle avec la souris ?

```
rect.setOnMouseDragged(new EventHandler<MouseEvent>() {
    public void handle(MouseEvent event) {
        pane.setLayoutX();
        pane.setLayoutY();
        Emp che la sortie du rectangle hors du pane
        pane.setLayoutX(x + " ");
        if (x < w/2) x = w/2;
        if (x > pane.getWidth()-w/2) x = pane.getWidth()-w/2;
        if (y < w/2) y = w/2;
        if (y > pane.getHeight()-w/2) y = pane.getHeight()-w/2;
    }
});
```



5eme exemple

Comment d placer le rectangle avec la souris ?

```
rect.setOnMouseDragged(new EventHandler<MouseEvent>() {
    public void handle(MouseEvent event) {
        double x = event.getSceneX() - pane.getLayoutX();
        double y = event.getSceneY() - pane.getLayoutY();
        label.setText("(" + x + ", " + y + ")");
        if (x < w/2) x = w/2;
        if (y > pane.getHeight() - w/2) y = pane.getHeight() - w/2;
        rect.setTranslateX(x - w/2);
        rect.setTranslateY(y - w/2);
    }
});
```

Positionne le centre du rectangle sous la souris



5eme exemple

D tecter la fermeture de la fen tre principale

```
primaryStage.setOnCloseRequest(new EventHandler<WindowEvent>() {  
    public void handle(WindowEvent event) {
```

Gestion de l' v nement onCloseRequest
sur l'objet primaryStage

```
    }  
};
```



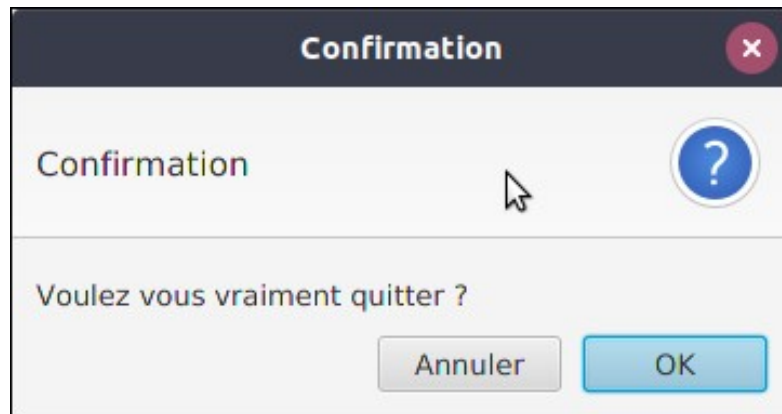
5eme exemple

D tecter la fermeture de la fen tre principale

```
primaryStage.setOnCloseRequest(new EventHandler<WindowEvent>() {
    public void handle(WindowEvent event) {
        Alert alert = new Alert(AlertType.CONFIRMATION);
        alert.setContentText("Voulez vous vraiment quitter ?");

        Optional<ButtonType> result = alert.showAndWait();
    }
});
```

Affichage d'une boite de dialogue demandant confirmation





5eme exemple

D tecter la fermeture de la fen tre principale

```
primaryStage.setOnCloseRequest(new EventHandler<WindowEvent>() {
    public void handle(WindowEvent event) {
        Alert alert = new Alert(AlertType.CONFIRMATION);
        alert.setContentText("Voulez vous vraiment quitter ?");

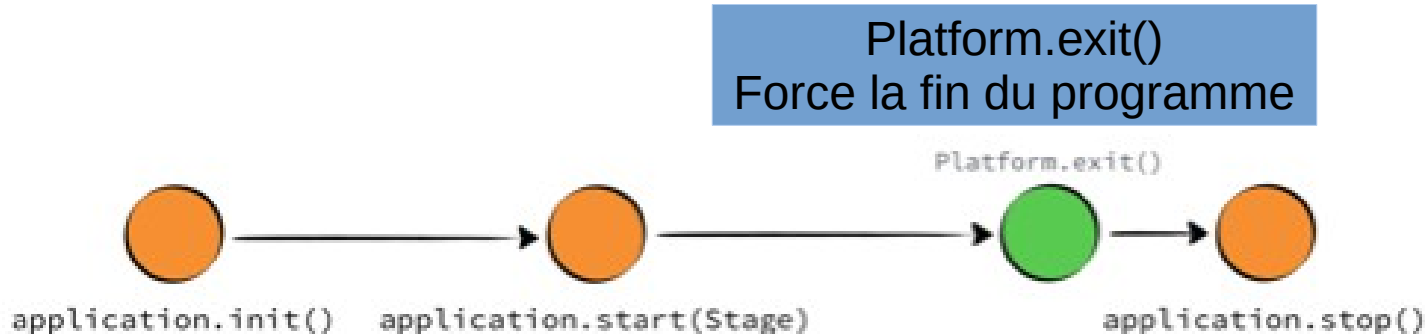
        Optional<ButtonType> result = alert.showAndWait();
        if (!result.equals(Optional.of(ButtonType.OK))) {
            event.consume();
        }
    }
});
```

Si r ponse n gative,
annulation de l' v nement onCloseRequest



5eme exemple

- Cycle de vie complet d'un programme JavaFX





5eme exemple

Notions de source et cible d'un  v nement

```
borderPane.setOnMouseClicked(  
    new EventHandler<MouseEvent>() {  
        public void handle(MouseEvent ev) {  
            System.out.println("source = " + ev.getSource());  
            System.out.println("target = " + ev.getTarget());  
        }  
    }  
);
```



```
source = BorderPane@5d964414  
target = Rectangle[x=0.0, y=0.0, width=20.0, height=20.0,  
fill=0x87cefa]
```

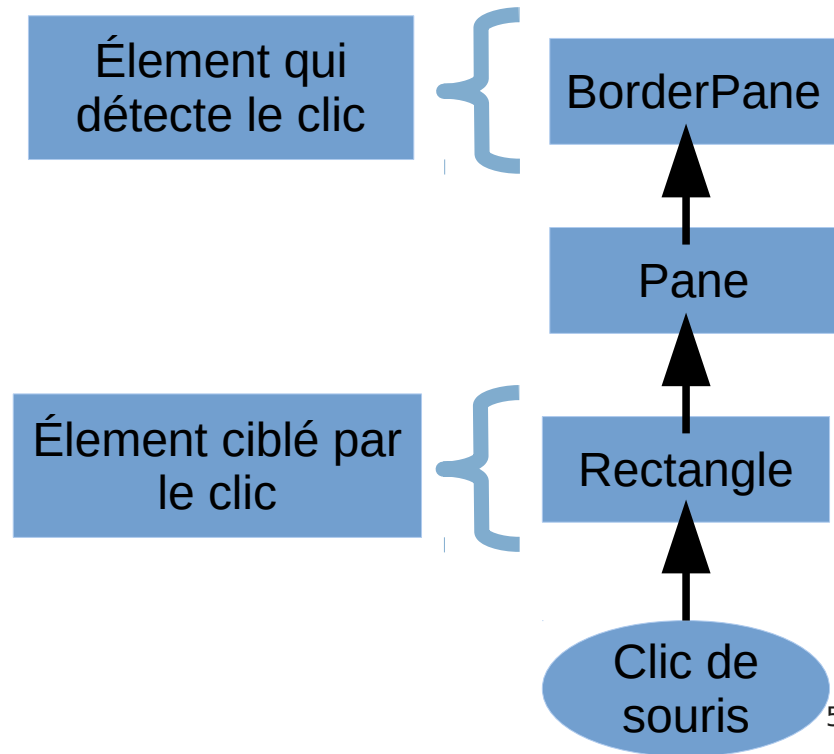


5eme exemple

Notions de source et cible d'un évènement

```
borderPane.setOnMouseClicked(  
    new EventHandler<MouseEvent>() {  
        public void handle(MouseEvent ev) {  
            System.out.println(  
                "source = " + ev.getSource());  
            System.out.println(  
                "target = " + ev.getTarget());  
        }  
    }  
);
```

source = BorderPane@5d964414
target = Rectangle





Récapitulatif sur les évènements et les propriétés observables



R capitulatif sur les  v nements

objet source
de l' v nement

- Stage
- Scene
- Button
- etc...

d claration d'un listener :  couter l' v nement

- Stage.setOnCloseRequest
- Stage.setOnHidding
- Stage.setOnHidden
- etc...
- Button.setAction
- Button.setOnMouseMoved
- etc...
- TextField.setOnMouseMoved
- TextField.setOnKeyPressed
- TextField.setOnKeyReleased
- TextField.setOnKeyTyped
- etc...

chaque objet source
poss de son jeu d' v nement



R capitulatif sur les  v nements

objet source
de l' v nement

- Stage
- Scene
- Button
- etc...

d claration d'un listener

- Stage.setOnCloseRequest
- Stage.setOnHidding
- Stage.setOnHidden
- etc...

- Button.setAction
- Button.setOnMouseMoved
- etc...

- TextField.setOnMouseMove
- TextField.setOnKeyPressed
- TextField.setOnKeyReleased
- TextField.setOnKeyTyped
- etc...

type du listener
correspondant   l' v nement

EventHandler<WindowEvent>

EventHandler<ActionEvent>

EventHandler<MouseEvent>

EventHandler<KeyEvent>



R capitulatif sur les  v nements

type du listener
correspondant   l' v nement

EventHandler<WindowEvent>

EventHandler<ActionEvent>

EventHandler<MouseEvent>

EventHandler<KeyEvent>

```
src.setOnMouseMoved(  
    new EventHandler<MouseEvent>() {  
        public void handle(MouseEvent ev) {
```

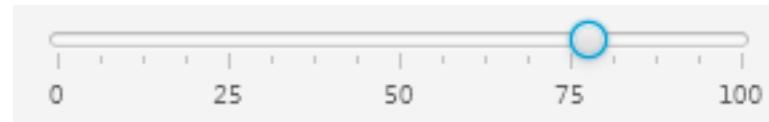
Informations sur l' v nement



R capitulatif sur les propri t s observables

De plus, un  l ment graphique peut poss der une ou plusieurs propri t s

- TextField
 - texte saisie
 - texte s lectionn 
- Slider (curseur)
 - valeur s lectionn e sur le curseur
- Stage (fen tre)
 - hauteur
 - largeur



Récapitulatif sur les propriétés observables

Il est possible de suivre toutes les modifications d'une propriété

```
primaryStage.heightProperty().addListener(  
    new ChangeListener<Number>() {
```

Nom de la propriété

```
});  
primaryStage.widthProperty().addListener(  
    new ChangeListener<Number>() {
```

Type du listener

```
});
```

R  capitulatif sur les propri  t  s observables

Il est possible de suivre toutes les modifications d'une propri  t  

```
primaryStage.heightProperty().addListener(
    new ChangeListener<Number>() {
        public void changed(ObservableValue<? extends Number> observable,
            Number oldValue, Number newValue) {
            System.out.println("Hauteur stage = " + newValue);
        }
    });

primaryStage.widthProperty().addListener(
    new ChangeListener<Number>() {
        public void changed(ObservableValue<? extends Number> observable,
            Number oldValue, Number newValue) {
            System.out.println("Largeur stage = " + newValue);
        }
    });
```

Ancienne et
nouvelle valeur



Compl ments de cours



Compl ments

Case   cocher

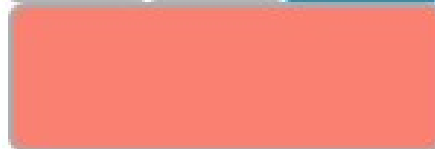
<https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/checkbox.htm>

Toogle button (bouton bistable)

<https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/toggle-button.htm>

☒ Security
☒ Project
☐ Chart

Priority:



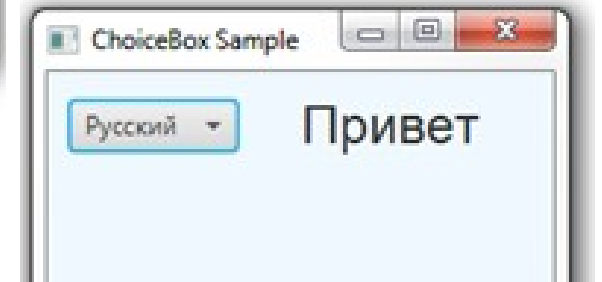
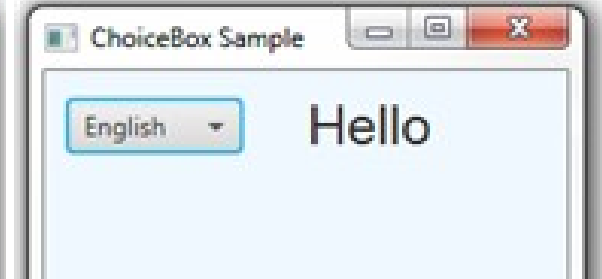
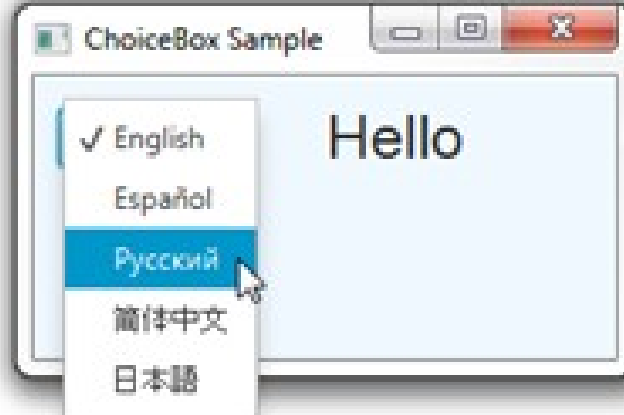
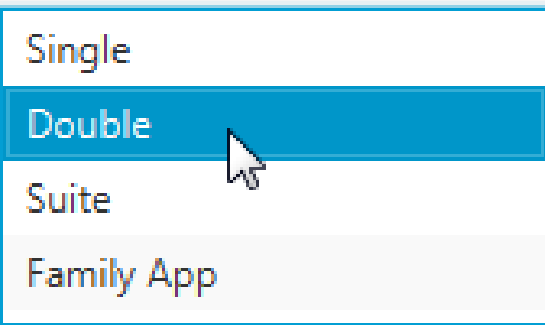


Compl ments

Liste de choix / liste de choix d roulante

<https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/list-view.htm>

<https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/choice-box.htm>





Compléments

Barre de menus

https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/menu_controls.htm





Compléments

Conteneur avec barres de défilement

<https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/scrollpane.htm>





Compl ments

Table

<https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/table-view.htm>

Affichage arborescent

<https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/tree-view.htm>

Table View Sample

Address Book

First Name	Last Name	Email
Jacob	Smith	jacob.smith@example.com
Isabella	Johnson	isabella.johnson@example.com
Ethan	Williams	ethan.williams@example.com
Emma	Jones	emma.jones@example.com
Michael	Brown	michael.brown@example.com

Tree View Sample

- MyCompany Human Resources
 - Accounts Department
 - Jacob Smith
 - Isabella Johnson
 - Add Employee
 - Emma Jones
 - Michael Brown



Compléments

Propriétés observables

<https://docs.oracle.com/javase/8/javafx/properties-binding-tutorial/binding.htm>

Gestion des événements

<https://docs.oracle.com/javase/8/javafx/events-tutorial/events.htm>

Tâches en arrière-plan

<https://docs.oracle.com/javase/8/javafx/interoperability-tutorial/concurrency.htm>

Librairie de tests d'intégration : TestFX

<https://github.com/TestFX/TestFX/wiki>

Site de questions/réponses spécialisé dans l'UX (expérience utilisateur)

<https://ux.stackexchange.com/>



Origine des images

- Page 5 : jaxenter.com
- Pages 19, 26 à 28, 49 : dzone.com
- Pages 29, 60 à 64 : oracle.com