

Secteur Tertiaire Informatique  
Filière « Etude et développement »

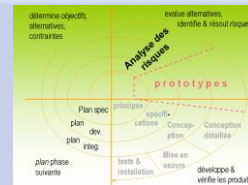
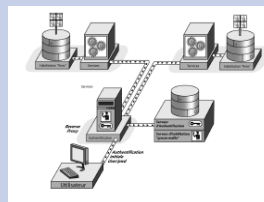
**TP – Développer des interfaces graphiques**

**IHM Java FX – Radio buttons**

Apprentissage

Mise en situation

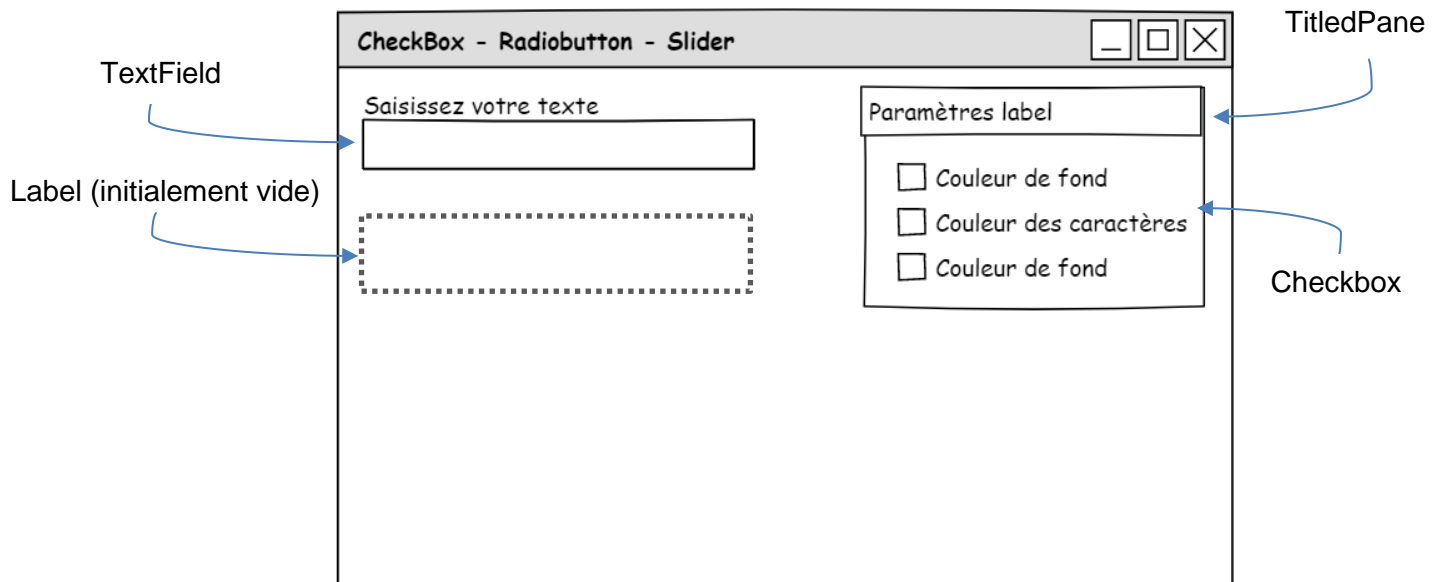
Evaluation



## 1. MANIPULATION DES GROUPES DE CONTROLES

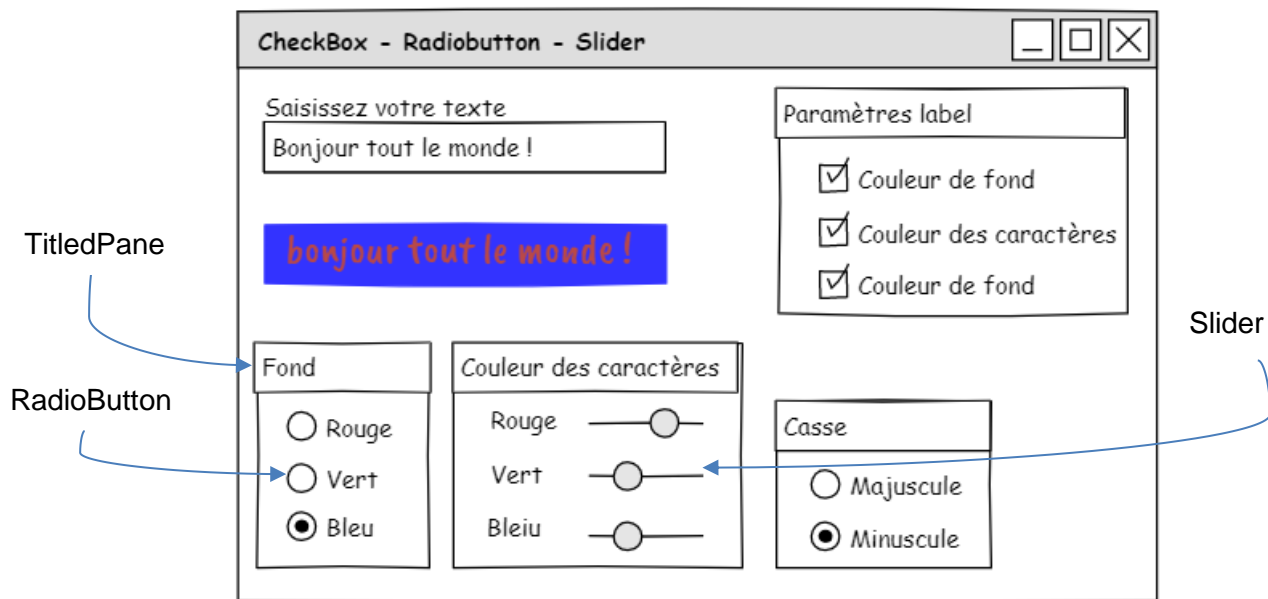
L'objectif est de mettre en place des contrôles tels des **cases à cocher** (classe « Checkbox »), des **boutons radio** (classe « RadioButton ») et des **barres coulissantes** (classe « Slider »).

Ci-dessous une maquette en fil de fer (« wireframe ») du résultat attendu au lancement du programme.



En cours de programme l'utilisateur pourra :

- Cocher les **checkboxes** qui l'intéressent pour afficher les paramètres de mise en forme souhaités (cf. figure ci-dessous) ;
- Saisir du texte à afficher qui permettra de remplir le label qui devra être situé sous le champ indiqué par « Saisissez votre texte » ;
- Choisir des paramètres de mise en forme parmi les options proposées (cf. figure ci-dessous).



Initialement, la fenêtre offre simplement à l'utilisateur la saisie de texte qui sera recopié caractères à caractères dans un label juste en dessous.

**Au premier caractère saisi** par l'utilisateur, les cases à cocher situées à droite deviennent accessibles.

Les cases à cocher sont initialement décochées. Dès que l'utilisateur en cochera une le « TitledPane » associé devra apparaître en dessous.

Les parties suivantes détaillent le fonctionnement de chaque option de modification de texte.

### 1.1 COULEUR DE FOND

La coche de la case à cocher « Couleur du fond » fait apparaître le groupe d'options « Fond ». Un élément coché dans ce groupe d'options modifie la couleur de fond du label (vous pourrez passer par une **propriété CSS** appliquée au « Label » concerné, <https://docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html>).

Le CSS (pour « Cascading Style Sheet » ou « Feuille de style en cascade ») a initialement été utilisé pour mettre en forme des pages web et est rapidement devenu un standard. JavaFX a adopté ce standard afin de pouvoir modifier ses composants graphiques. Vous verrez plus tard le fonctionnement du CSS,

La méthode à utiliser pour modifier le CSS d'un composant graphique est : « `setStyle()` », cette méthode est héritée de la classe « Node » ([par ici](#) pour une illustration des héritages de classes au sein de JavaFX).

Par exemple, pour modifier le style d'un bouton il serait possible d'utiliser le code suivant :

```
Button boutonColor = new Button("Color");
boutonColor.setStyle("-fx-background-color: slateblue; -fx-text-fill: white;");
```

La propriété CSS « **-fx-background-color** », permet de modifier la couleur de fond d'un composant graphique et la propriété « **-fx-text-fill** » permet de donner une couleur au texte.

La coche de la case à cocher « Couleur des caractères » fait apparaître les barres coulissantes de réglage de la couleur des caractères.

A vous de gérer **les évènements** pour que le fond change en fonction de ce qui est sélectionné.

## 1.2 COULEUR DES CARACTERES

Les barres coulissantes correspondant aux différentes couleurs sont des objets de la classe « Slider » ([https://docs.oracle.com/javafx/2/ui\\_controls/slider.htm](https://docs.oracle.com/javafx/2/ui_controls/slider.htm)).

La couleur suivra le [codage RGB](#) permettant de coder une couleur sur 3 nombres aux valeurs comprises entre **0 et 255**.

Chaque « slider » devra donc permettre de régler les valeurs de **0 à 255** (à vous de voir comment borner les valeurs en vous [consultant la documentation](#)).

Pour modifier la couleur d'un texte de label vous pouvez passer par une propriété CSS (« fx-text-fill ») ou utiliser la méthode « setTextFill() ».

### 1.2.1 Méthode CSS

Si vous optez pour la méthode CSS il vous faudra adapter la propriété suivante :

```
-fx-text-fill: rgb(255,0,0)
```

« **rgb(<integer-rouge>, <integer-vert>, <integer-bleu>)** » est une **fonction CSS**.

### 1.2.2 Modification par l'utilisation de méthode JavaFX

La méthode « **setTextFill()** » de la classe « **Label** » (héritée de « Labeled ») permet de modifier uniquement la valeur de la propriété « text-fill ».

Voilà comment elle est définie (extrait de la documentation accessible à l'adresse suivante pour le JDK 18 :

[https://openjfx.io/javadoc/18/javafx.controls/javafx/scene/control/Labeled.html#setTextFill\(javafx.scene.paint.Paint\)](https://openjfx.io/javadoc/18/javafx.controls/javafx/scene/control/Labeled.html#setTextFill(javafx.scene.paint.Paint))).

```
public final void setTextFill(Paint value)
```

Sets the value of the property textFill.

Property description:

The [Paint](#) used to fill the text.

Default value:

```
Color.BLACK
```

La méthode prend en paramètre un objet de la classe « **Paint** » qui peut être créé en utilisant une méthode « static » de cette même classe, méthode appelée « **valueOf(String str)** ».

Cette méthode transforme une **représentation en chaîne de caractère** en objet de « **Paint** ».

### 1.2.3 Construction de la chaîne de caractère représentant un couleur

Quel que soit l'approche choisie, il vous faudra construire une chaîne de caractère représentant la couleur choisie par l'utilisateur.

Pour se faire, créez une fonction avec un nom qui vous semble pertinent. L'objectif de la fonction est de transformer 3 valeurs décimales en une chaîne de caractères.

Question à se poser pour la construction de la fonction :

- Quels sont les paramètres ?
- Quel est le type de retour ?
- Que dois-je concaténer ?

Jeu de test :

- Entrée : 3 entiers représentant la couleur rouge → 255, 0, 0
- Sortie : la chaîne suivante

```
"rgb(255, 0, 0)"
```

### 1.3 CASSE DE LA CHAÎNE DE CARACTERES

La coche de la case à cocher Casse fait apparaître le groupe d'options Casse. Un élément coché dans ce groupe d'options fera passer en minuscules/majuscules le texte recopié.

Pour transformer la casse du texte vous pouvez utiliser des méthodes de la classe String : [« toLower\(\) »](#) et [« toUpper\(\) »](#).

## **CREDITS**

### **ŒUVRE COLLECTIVE DE l'AFPA**

**Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services**

### **Equipe de conception (IF, formateur, mediatiseur)**

Michel Coulard – Formateur Evry

Chantal Perrachon – IF Neuilly sur Marne

**Date de mise à jour : 11/07/2024**

## **Reproduction interdite**

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »

Programmation orientée objet – Première application

Afpa © 2022 – Section Tertiaire Informatique – Filière « Etude et développement » 6/10