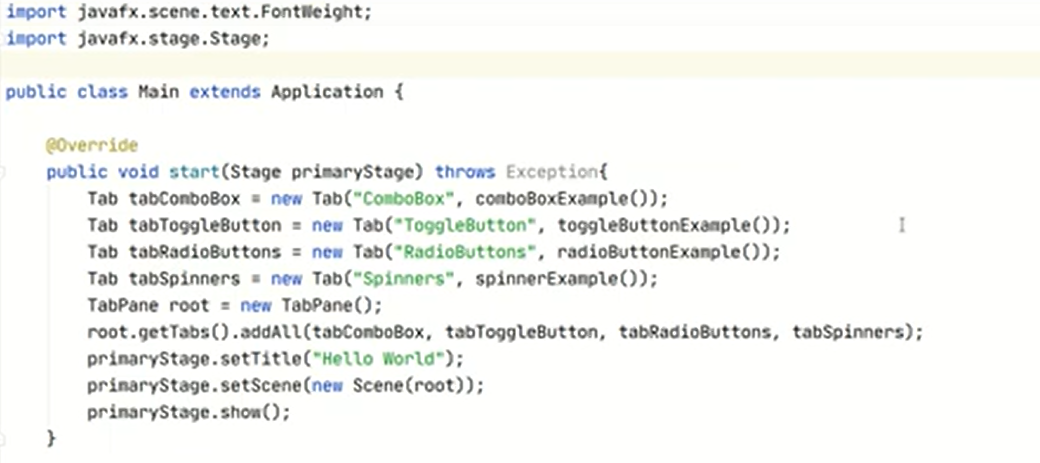
**Примеры создания компонентов ComboBox,**

**ToggleButton, RadioButton, Spinner**

Рассмотрим другие визуальные компоненты. Начнём с класса ComboBox, объекты которого позволяют делать выбор из списка элементов.

Листинг кода:

m

Сначала необходимо создать список ObservableList<T> объектов для выбора, T – класс к которому относятся эти объекты. Список передаётся как параметр конструктора класса ComboBox или метода setItems(ObservableList<T> value). Отображаются объекты как правило в виде строк. Если выбор осуществляется непосредственно из объектов класса String, то можно установить возможность ввода собственной строки, отличающейся от элементов списка, при помощи метода setEditable(boolean value) с параметром true. Получить выбранное значение можно как результат вызова метода T getValue(), а установить выбор элемента в коде программы методом void setValue(T value).

Код метода для создания ComboBox выглядит так:

private VBox comboBoxExample() {

ObservableList<String> options

= FXCollections.*observableArrayList*(

"Monday",

"Tuesday",

"Wednesday",

"Thursday",

"Friday",

"Saturday",

"Sunday"

);

ComboBox comboBox = new ComboBox(options);

comboBox.setValue("Friday");

comboBox.setEditable(true);

Label label = new Label();

label.setFont(Font.*font*(30));

label.setPrefSize(400, 50);

comboBox.setPrefSize(300, 50);

comboBox.setStyle("-fx-font-size: 30px");

Button btn = new Button();

btn.setPrefSize(300, 50);

btn.setFont(Font.*font*(30));

btn.setText("Read comboBox");

btn.setOnAction((event) -> {

if (comboBox.getValue() != null && !comboBox.getValue().toString().isEmpty()) {

label.setText("selected: " + comboBox.getValue());

}

else label.setText(" no selected item");

});

VBox vBox = new VBox();

vBox.setPadding(new Insets(50, 100, 50, 100));

vBox.setSpacing(15);

vBox.getChildren().addAll(comboBox, btn, label);

return vBox;

}

Рассмотрим примеры кнопок. Класс ToggleButton создаёт кнопку переключатель.

Внешний вид объекта класса ToggleButton и описание действия при нажатии на кнопку аналогичны свойствам обычной кнопки Button. Отличие заключается в том, что в классе ToggleButton есть метод boolean isSelected(), позволяющий определить включена кнопка или выключена.

Приведём соответствующий код:

private VBox toggleButtonExample() {

Label label = new Label("LABEL");

label.setFont(Font.*font*(20));

label.setPrefSize(400, 50);

label.setAlignment(Pos.*CENTER*);

ToggleButton btn = new ToggleButton();

btn.setFont(Font.*font*(20));

btn.setPrefSize(400, 50);

btn.setText("Push!");

btn.setOnAction((event) -> {

if (btn.isSelected()) {

label.setText("ON");

label.setFont(Font.*font*("Tahoma", FontWeight.*NORMAL*, 40));

label.setBackground(new Background(new BackgroundFill(Color.*AQUA*, CornerRadii.*EMPTY*, Insets.*EMPTY*)));

} else {

label.setText("OFF");

label.setFont(Font.*font*("Tahoma", FontWeight.*MEDIUM*, 20));

label.setBackground(new Background(new BackgroundFill(Color.*YELLOWGREEN*, CornerRadii.*EMPTY*, Insets.*EMPTY*)));

}

});

VBox vBox = new VBox(20);

vBox.setPadding(new Insets(50, 100, 50, 100));

vBox.getChildren().addAll(btn, label);

return vBox;

}

Класс RadioButton позволяет создать группу радиокнопок.

Особенностью радиокнопок является то, что они определяют выбор одного элемента из нескольких . Поэтому кнопки соответствующего набора должны относиться к одной группе, задаваемой объектом класса ToggleGroup. Принадлежность радиокнопки к группе назначается вызовом у неё метода setToggleGroup(ToggleGroup value).

Приведём код создания двух групп радиокнопок из примера:

private HBox radioButtonExample() {

HBox root = new HBox();

GridPane pane1 = new GridPane();

pane1.setHgap(40);

pane1.setVgap(110);

pane1.setPadding(new Insets(20, 20, 20, 20));

Label labelImage1 = new Label();

labelImage1.setPrefSize(210, 270);

pane1.add(labelImage1, 1, 0, 1, 3);

ToggleGroup group1 = new ToggleGroup();

RadioButton buttonDog1 = new RadioButton("dog");

buttonDog1.setToggleGroup(group1);

buttonDog1.setFont(Font.*font*(20));

buttonDog1.setOnAction((event) -> {

Image imageDog = new Image(getClass().getResourceAsStream("images/dog.jpeg"));

ImageView imvDog = new ImageView(imageDog);

labelImage1.setGraphic(imvDog);

});

pane1.add(buttonDog1, 0, 0);

RadioButton buttonHouse1 = new RadioButton("house");

buttonHouse1.setToggleGroup(group1);

buttonHouse1.setFont(Font.*font*(20));

buttonHouse1.setOnAction(( event) -> {

Image imageHouse = new Image(getClass().getResourceAsStream("images/house.jpeg"));

ImageView imvHouse = new ImageView(imageHouse);

labelImage1.setGraphic(imvHouse);

});

pane1.add(buttonHouse1, 0, 1);

RadioButton buttonTree1 = new RadioButton("tree");

buttonTree1.setToggleGroup(group1);

buttonTree1.setFont(Font.*font*(20));

buttonTree1.setOnAction((event) -> {

Image imageTree = new Image(getClass().getResourceAsStream("images/tree.jpeg"));

ImageView imvTree = new ImageView(imageTree);

labelImage1.setGraphic(imvTree);

});

pane1.add(buttonTree1, 0, 2);

root.getChildren().add(pane1);

Separator separator = new Separator();

separator.setOrientation(Orientation.*VERTICAL*);

root.getChildren().add(separator);

GridPane pane2 = new GridPane();

pane2.setHgap(40);

pane2.setVgap(110);

pane2.setPadding(new Insets(20, 20, 20, 20));

Label labelImage2 = new Label();

labelImage2.setPrefSize(210, 270);

pane2.add(labelImage2, 1, 0, 1, 3);

ToggleGroup group2 = new ToggleGroup();

RadioButton buttonDog2 = new RadioButton("dog");

buttonDog2.setToggleGroup(group2);

buttonDog2.setFont(Font.*font*(20));

pane2.add(buttonDog2, 0, 0);

RadioButton buttonHouse2 = new RadioButton("house");

buttonHouse2.setToggleGroup(group2);

buttonHouse2.setFont(Font.*font*(20));

pane2.add(buttonHouse2, 0, 1);

RadioButton buttonTree2 = new RadioButton("tree");

buttonTree2.setToggleGroup(group2);

buttonTree2.setFont(Font.*font*(20));

pane2.add(buttonTree2, 0, 2);

Button btn = new Button("Ok");

btn.setFont(Font.*font*(20));

btn.setOnAction(( event) -> {

if (buttonDog2.isSelected()) {

Image imageDog = new Image(getClass().getResourceAsStream("images/dog.jpeg"));

ImageView imvDog = new ImageView(imageDog);

labelImage2.setGraphic(imvDog); }

if (buttonHouse2.isSelected()) {

Image imageHouse = new Image(getClass().getResourceAsStream("images/house.jpeg"));

ImageView imvHouse = new ImageView(imageHouse);

labelImage2.setGraphic(imvHouse); }

if (buttonTree2.isSelected()) {

Image imageTree = new Image(getClass().getResourceAsStream("images/tree.jpeg"));

ImageView imvTree = new ImageView(imageTree);

labelImage2.setGraphic(imvTree); }

});

pane2.add(btn, 0, 3, 2, 1);

root.getChildren().add(pane2);

return root;}

Для ввода чисел можно использовать специальные счётчики Spinner<Integer> и Spinner<Double>.

Объект класса Spinner<T> представляет собой однострочное текстовое поле, которое позволяет выбрать число или значение объекта из упорядоченной последовательности. Поле содержит пару крошечных кнопок со стрелками для пошагового перемещения по элементам последовательности. Клавиши со стрелками вверх / вниз также переключают элементы. В конструкторе счетчика чисел, например, целых Spinner(int min, int max, int initialValue, int amountToStepBy), можно задать границы перемещения, начальное значение и шаг. Вызвав метод setEditable(boolean value) с параметром true, можно разрешить вводить допустимое значение непосредственно в счетчик.

Приведём код метода, в котором создаются два счетчика:

private VBox spinnerExample() {

Spinner<Integer> spInt = new Spinner(-100, 100, 0);

spInt.setStyle("-fx-font-size: 30px");

spInt.setPrefWidth(170);

Spinner<Double> spDouble = new Spinner(-100.0, 100.0, 1.0, 0.1);

spDouble.setEditable(true);

spDouble.setStyle("-fx-font-size: 30px");

spDouble.setPrefWidth(170);

Label label = new Label();

label.setFont(Font.*font*(30));

label.setPrefSize(400, 100);

Button btn = new Button();

btn.setText("Ok");

btn.setFont(Font.*font*(30));

btn.setOnAction((event) -> {

label.setText("integer spinner: "+spInt.getValue()+"\ndouble spinner: "+spDouble.getValue());

});

VBox vBox = new VBox(20);

vBox.setPadding(new Insets(50, 100, 50, 100));

vBox.getChildren().addAll(spInt, spDouble, btn, label);

return vBox;}