

* Observable của JavaFX chỉ có 2 phương thức là addListener và removeListener không có phương thức làm nhiệm vụ notify cho Observer.
* addListener đảm nhiệm vai trò như phương thức đăng kí cho Observer
* removeListener đảm nhiệm vai trò như hủy đăng kí cho Observer
* JavaFX cả Observeble và Observer đều sử dụng các Interface để giao tiếp không giống như mô hình tổng quát có thể sử dụng extends
* Bất cứ khi nào dữ liệu trong Observable được cập nhật, hàm onChanged sẽ notify các thay đổi xuống cho các Observer.
* Thay vì hàm notify độc lập, method addListener nhận vào các InvalidationListener kiểm tra sự thay đổi trước vào sau đối với dữ liệu để quyết định xem có yêu cầu cập nhật hay không.
* Mã nguồn sau minh họa cho điều đó.

package org.o7planning.javafx.choicebox;  
  
import org.o7planning.javafx.model.Language;  
  
import javafx.application.Application;  
import javafx.beans.value.ChangeListener;  
import javafx.beans.value.ObservableValue;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.geometry.Insets;  
import javafx.scene.Scene;  
import javafx.scene.control.ChoiceBox;  
import javafx.scene.control.Label;  
import javafx.scene.layout.FlowPane;  
import javafx.stage.Stage;  
  
public class ChoiceBoxEventDemo extends Application {  
  
 @Override  
 public void start(Stage primaryStage) {  
  
 Language vietnamese = new Language("vi", "Vietnamese", "Xin Chào");  
 Language english = new Language("en", "English", "Hello");  
 Language russian = new Language("ru", "Russian", "привет");  
  
 FlowPane root = new FlowPane();  
  
 Label label = new Label("Select Language:");  
 Label greetingLabel = new Label("");  
  
 ObservableList<Language> languages //  
 = FXCollections.observableArrayList(vietnamese, english, russian);  
  
 ChoiceBox<Language> choiceBox = new ChoiceBox<Language>(languages);  
  
 ChangeListener<Language> changeListener = new ChangeListener<Language>() {  
  
 @Override  
 public void changed(ObservableValue<? extends Language> observable, //  
 Language oldValue, Language newValue) {  
 if (newValue != null) {  
 greetingLabel.setText(newValue.getGreeting());  
 }  
 }  
 };  
 // Sự kiện khi thay đổi Item trên ChoiceBox  
 choiceBox.getSelectionModel().selectedItemProperty().addListener(changeListener);  
  
 root.setPadding(new Insets(10));  
  
 root.getChildren().addAll(label, choiceBox, greetingLabel);  
 root.setPadding(new Insets(10));  
 root.setHgap(10);  
  
 primaryStage.setTitle("ChoiceBox (o7planning.org)");  
 Scene scene = new Scene(root, 400, 300);  
 primaryStage.setScene(scene);  
 primaryStage.show();  
 }  
  
 public static void main(String[] args) {  
 launch(args);  
 }  
  
}