*package com.example.finitestatemachine*;  
  
*public class Application* {  
 *public static void* main(*String*[] *args*) {  
 *Main*.*main*(*args*);  
 }  
}

*package com.example.finitestatemachine*;  
  
*import gui.utils.Alerts*;  
*import javafx.application.*Application;  
*import javafx.fxml.FXMLLoader*;  
*import javafx.scene.Scene*;  
*import javafx.scene.control.Alert*;  
*import javafx.stage.Stage*;  
  
*import java.io.IOException*;  
  
*public class Main extends* Application {  
 *@Override  
 public void* start(*Stage stage*) *throws IOException* {  
 *FXMLLoader* fxmlLoader = *new* FXMLLoader(*Main*.*class*.getResource("window-view.fxml"));  
 *Scene* scene = *new* Scene(fxmlLoader.load(), 807, 300);  
 *stage*.setTitle("Finite State Machine");  
 *stage*.setScene(scene);  
 *stage*.show();  
 }  
  
 *public static void* main(*String*[] *args*) { *launch*(); }  
}

*package gui.utils*;  
  
*import javafx.scene.control.Alert*;  
*import javafx.scene.control.Alert*.*AlertType*;  
  
*public class Alerts* {  
 *public static void* showAlert(*String title*, *String header*, *String content*, *AlertType type*) {  
 *Alert* alert = *new* Alert(*type*);  
 alert.setTitle(*title*);  
 alert.setHeaderText(*header*);  
 alert.setContentText(*content*);  
 alert.show();  
 }  
}

*package com.example.finitestatemachine*;  
  
*import gui.utils.Alerts*;  
*import javafx.fxml.FXML*;  
*import javafx.fxml.FXMLLoader*;  
*import javafx.scene.*Parent;  
*import javafx.scene.Scene*;  
*import javafx.scene.control.*\*;  
*import javafx.scene.image.Image*;  
*import javafx.scene.image.ImageView*;  
*import javafx.scene.layout.Pane*;  
*import javafx.stage.Stage*;  
  
*import java.io.IOException*;  
*import java.net.URL*;  
*import java.util.*ResourceBundle;  
  
*public class Controller* { */\* Interface controller. \*/  
 /\* Declaration of the FXML items. \*/  
 @FXML  
 private Button* testButton; *// Button to start the test.  
  
 @FXML  
 private TextField* textInput; *// Text field to input the automata.  
  
 @FXML  
 private TextArea* textInfo; *// Text area to show the operations.  
  
 @FXML  
 private Label* opLabel; *// Label to show the Operations/Operações text.  
  
 @FXML  
 private ImageView* creatorLogo; *// ImageView to show the creator logo.  
  
 @FXML  
 ImageView* afdImage; *// ImageView to show the finite state design.  
  
 @FXML  
 private Label* outputLabel; *// Label to show below the text area the result for the automata.  
  
 @FXML  
 private Button* helpButton; *// Button to show the help page.  
  
 @FXML  
 private Label* helpLabel; *// Label to show the text indicating the user to use the help button.  
  
 @FXML  
 private Label* creatorFooter; *// Label to show the credits about the creator on the footer.  
  
 @FXML  
 private Button* preset1; *// Button to show an example of an acceptable automata.  
  
 @FXML  
 private Button* preset2; *// Button to show an example of a non-acceptable automata.  
 /\* Declaration of the FXML items. \*/  
  
 @FXML  
 protected void* onPreset1ButtonClickAction() {  
 *String* automata = "0100101101";  
 operations(automata);  
 }  
  
 *@FXML  
 protected void* onPreset2ButtonClickAction() {  
 *String* automata = "01101001";  
 operations(automata);  
 }  
  
 *@FXML  
 protected void* onTestButtonClickAction() {  
 *String* automata = textInput.getText();  
 *if* (automata.equals("")) {  
 *Alerts*.*showAlert*("Cadeia inexistente", *null*, """  
 Por favor, insira uma cadeia antes de iniciar o teste.  
 """, *Alert*.*AlertType*.WARNING);  
 automata = textInput.getText();  
 } *else if* (!automataValidation(automata)) {  
 *Alerts*.*showAlert*("Cadeia inválida", *null*, """  
 A cadeia inserida é inválida. Por favor, certifique-se de que a cadeia possui apenas números (i.e: 00101000110).  
   
 OBS: Certifique-se de utilizar o alfabeto {0, 1} (ou seja, apenas números 1 e 0). Caso contrário, a cadeia será negada e o teste não será iniciado.  
 """, *Alert*.*AlertType*.WARNING);  
 automata = textInput.getText();  
 } *else* {  
 operations(automata);  
 }  
 }  
  
 *@FXML  
 protected void* onHelpButtonClickAction() {  
 *Alerts*.*showAlert*("Finite State Machine", *null*, """  
 Este simulador de autômatos finitos determinísticos aceita cadeias com quantidades ímpares de números 1.  
  
 As operações realizadas pelo AFD são mostradas no quadro ao lado. Abaixo do quadro será exibida a informação de aceitação ou negação do AFD inserido.  
   
 Alfabeto: {0, 1} | Estados: {q0, q1}  
 Estado Inicial: {q0} | Estado de Aceitação: {q1}  
   
 Para ver um rápido exemplo do funcionamento deste simulador, faça uso dos botões "Preset 1" e "Preset 2". O botão "Preset 1" irá exemplificar uma cadeia válida (aceita), enquanto o botão "Preset 2" irá exemplificar uma cadeia inválida (negada).  
  
 Para iniciar o teste, insira um AFD no bloco de texto "Informe o autômato" e clique no botão Testar que fica logo abaixo do mesmo.  
 """, *Alert*.*AlertType*.INFORMATION);  
 }  
  
 *private boolean* automataValidation(*String automata*) {  
 *boolean* result = *true*;  
 *for* (*int* i = 0; i < *automata*.length(); i++) {  
 *if* (!*Character*.*isDigit*(*automata*.charAt(i))) {  
 result = *false*;  
 } *else if* (*automata*.charAt(i) != '1' && *automata*.charAt(i) != '0') {  
 result = *false*;  
 }  
 }  
 *return* result;  
 }  
  
 *private void* operations(*String automata*) {  
 *int* countOp = 2;  
 *String* currentState = "";  
 textInfo.setText("Cadeia informada: " + *automata*);  
 textInfo.appendText("\nEstado Inicial: q0");  
 *if* (*automata*.charAt(0) == '0') {  
 textInfo.appendText("\n1ª Operação: q0 -> q0");  
 currentState = "q0";  
 }  
 *else if* (*automata*.charAt(0) == '1') {  
 textInfo.appendText("\n1ª Operação: q0 -> q1");  
 currentState = "q1";  
 }  
 *if* (*automata*.length() > 1) {  
 *for* (*int* i = 1; i < *automata*.length(); i++) {  
 *if* (currentState.equals("q0") && *automata*.charAt(i) == '0') {  
 textInfo.appendText("\n" + countOp + "ª Operação: q0 -> q0");  
 currentState = "q0";  
 }  
 *else if* (currentState.equals("q0") && *automata*.charAt(i) == '1') {  
 textInfo.appendText("\n" + countOp + "ª Operação: q0 -> q1");  
 currentState = "q1";  
 }  
 *else if* (currentState.equals("q1") && *automata*.charAt(i) == '0') {  
 textInfo.appendText("\n" + countOp + "ª Operação: q1 -> q1");  
 currentState = "q1";  
 }  
 *else if* (currentState.equals("q1") && *automata*.charAt(i) == '1') {  
 textInfo.appendText("\n" + countOp + "ª Operação: q1 -> q0");  
 currentState = "q0";  
 }  
 countOp++;  
 }  
 }  
 textInfo.appendText("\nEstado Final: " + currentState);  
 *if* (currentState.equals("q1")) {  
 outputLabel.setText("Cadeia aceita!");  
 } *else* {  
 outputLabel.setText("Cadeia negada!");  
 }  
 }  
  
 *public void* initialize(*URL url*, ResourceBundle *rb*) {  
 *Image* cl = *new* Image(getClass().getResourceAsStream("com/example/finitestatemachine/cl\_gnl.png"));  
 *Image* afd = *new* Image(getClass().getResourceAsStream("com/example/finitestatemachine/afd\_rps.png"));  
 creatorLogo.setImage(cl);  
 afdImage.setImage(afd);  
 }  
  
} */\* Interface controller. \*/*

<?*xml version*="1.0" *encoding*="UTF-8"?>  
  
<?*import* javafx.scene.control.Button?>  
<?*import* javafx.scene.control.Label?>  
<?*import* javafx.scene.control.TextArea?>  
<?*import* javafx.scene.control.TextField?>  
<?*import* javafx.scene.image.Image?>  
<?*import* javafx.scene.image.ImageView?>  
<?*import* javafx.scene.layout.Pane?>  
<?*import* javafx.scene.text.Font?>  
  
<Pane *fx:id*="Window" *maxHeight*="-Infinity" *maxWidth*="-Infinity" *minHeight*="-Infinity" *minWidth*="-Infinity" *prefHeight*="300.0" *prefWidth*="803.0" *xmlns*="http://javafx.com/javafx/17" *xmlns:fx*="http://javafx.com/fxml/1" *fx:controller*="com.example.finitestatemachine.Controller">  
 <children>  
 <ImageView *fx:id*="afdImage" *fitHeight*="474.0" *fitWidth*="612.0" *layoutX*="134.0" *layoutY*="-96.0" *pickOnBounds*="true" *preserveRatio*="true">  
 <image>  
 <Image *url*="@afd\_rps.png" />  
 </image>  
 </ImageView>  
 <Button *fx:id*="testButton" *layoutX*="80.0" *layoutY*="217.0" *mnemonicParsing*="false" *onAction*="#onTestButtonClickAction" *prefHeight*="35.0" *prefWidth*="71.0" *text*="Testar" *textAlignment*="CENTER" />  
 <TextField *fx:id*="textInput" *layoutX*="41.0" *layoutY*="181.0" *promptText*="Informe o autômato" />  
 <TextArea *fx:id*="textInfo" *layoutX*="565.0" *layoutY*="42.0" *prefHeight*="200.0" *prefWidth*="200.0" *wrapText*="true" />  
 <Label *fx:id*="outputLabel" *layoutX*="620.0" *layoutY*="244.0" *prefHeight*="21.0" *prefWidth*="101.0" *textAlignment*="JUSTIFY" *wrapText*="true" />  
 <Button *fx:id*="helpButton" *layoutX*="212.0" *layoutY*="181.0" *mnemonicParsing*="false" *onAction*="#onHelpButtonClickAction" *text*="Ajuda (!)" />  
 <Label *fx:id*="helpLabel" *layoutX*="41.0" *layoutY*="107.0" *prefHeight*="61.0" *prefWidth*="200.0" *text*="Para utilizar o programa de maneira correta, utilize o botão de Ajuda." *textAlignment*="CENTER" *wrapText*="true" />  
 <Label *fx:id*="creatorFooter" *layoutX*="170.0" *layoutY*="284.0" *text*="Criado por: Danilo &quot;giga&quot; Silva | Ciência da Computação - Universidade Católica de Pernambuco (Unicap).">  
 <font>  
 <Font *size*="10.0" />  
 </font>  
 </Label>  
 <ImageView *fx:id*="creatorLogo" *fitHeight*="91.0" *fitWidth*="108.0" *layoutX*="87.0" *layoutY*="10.0" *nodeOrientation*="INHERIT" *pickOnBounds*="true" *preserveRatio*="true">  
 <image>  
 <Image *url*="@cl\_gnl.png" />  
 </image>  
 </ImageView>  
 <Label *fx:id*="opLabel" *layoutX*="628.0" *layoutY*="14.0" *text*="Operações" *textAlignment*="CENTER">  
 <font>  
 <Font *name*="System Bold" *size*="15.0" />  
 </font>  
 </Label>  
 <Button *fx:id*="preset1" *layoutX*="343.0" *layoutY*="222.0" *mnemonicParsing*="false" *onAction*="#onPreset1ButtonClickAction" *text*="Preset 1" />  
 <Button *fx:id*="preset2" *layoutX*="440.0" *layoutY*="222.0" *mnemonicParsing*="false" *onAction*="#onPreset2ButtonClickAction" *text*="Preset 2" />  
 </children>  
</Pane>