НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ПРОГРАМА

**Розробка інформаційної бази даних (каталога) еталонних моделей обертання твердого тіла**

Текст програми

2071180. 00001 – 01 33

Аркушів 8

Представники розроблювача:

Керівник розробки

професор

Плаксій Ю.А.

Виконавець

ст. гр.І-22б

Ігнатьєв Я.В.

Харків 2017

package sample;  
  
import javafx.beans.value.ChangeListener;  
import javafx.beans.value.ObservableValue;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXMLLoader;  
import javafx.fxml.Initializable;  
import javafx.fxml.FXML;  
  
import javafx.scene.Node;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.ListCell;  
import javafx.scene.control.ListView;  
import javafx.scene.control.TextField;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.input.MouseEvent;  
import javafx.stage.Stage;  
import javafx.util.Callback;  
  
  
import java.io.IOException;  
import java.net.URL;  
import java.util.ResourceBundle;  
  
  
public class Controller implements Initializable{  
  
 @FXML  
 private ListView listView;  
  
 @FXML  
 private ImageView omegaModel;  
  
 @FXML  
 private ImageView omegaLabel;  
  
 @FXML  
 private ImageView thetaLabel;  
  
 @FXML  
 private ImageView lambdaLabel;  
  
 @FXML  
 private ImageView lambdaModel;  
  
 @FXML  
 private ImageView k1Value;  
  
 @FXML  
 private ImageView k2Value;  
  
 @FXML  
 private ImageView k3Value;  
  
 @FXML  
 private ImageView k4Value;  
  
 @FXML  
 private Button viewGraphs;  
  
 @FXML Button viewGraph1;  
 @FXML Button viewGraph2;  
 @FXML Button viewGraph3;  
  
  
  
 private final ObservableList<String> currentList= FXCollections.observableArrayList();  
  
 public static int position;  
  
 public static int choiceModel;  
  
 String[] firstModel = {"third\_point\_model/omegaModel.png", "third\_point\_model/lambdaModel.png", "third\_point\_model/k1Value.png",  
 "third\_point\_model/k2Value.png", "third\_point\_model/k3Value.png","third\_point\_model/k4Value.png"};  
  
 String[] secondModel = {"multi\_third\_point\_model/omegaModel.png", "multi\_third\_point\_model/lambdaModel.png", "multi\_third\_point\_model/k1Value1.png",  
 "multi\_third\_point\_model/k2Value1.png", "multi\_third\_point\_model/k3Value1.png"};  
  
 @Override  
 public void initialize(URL url, ResourceBundle rb) {  
  
 Image labelOmega = new Image(Main.class.getResourceAsStream("third\_point\_model/labelomega.png"));  
 Image labelLambda = new Image(Main.class.getResourceAsStream("third\_point\_model/lambaLabel.png"));  
 Image labelTheta = new Image(Main.class.getResourceAsStream("third\_point\_model/thetaLabel.png"));  
  
 omegaLabel.setImage(labelOmega);  
 thetaLabel.setImage(labelTheta);  
 lambdaLabel.setImage(labelLambda);  
  
 setImages(1, firstModel);  
  
 listView.getSelectionModel().selectedItemProperty().addListener(  
 new ChangeListener<String>() {  
 @Override  
 public void changed(ObservableValue<? extends String> ov, String t, String t1) {  
 switch (t1)  
 {  
 case "ТРЬОХЧАСТОТНА ТРИГОНОМЕТРИЧНА КВАТЕРНІОННА МОДЕЛЬ \nОБЕРТАННЯ ТВЕРДОГО ТІЛА":  
 choiceModel = 1;  
 setImages(choiceModel, firstModel);  
 break;  
 case "МУЛЬТИПЛІКАТИВНІ ТРЬОХЧАСТОТНІ МОДЕЛІ \nОБЕРТАННЯ ТВЕРДОГО ТІЛА":  
 choiceModel = 2;  
 setImages(choiceModel, secondModel);  
 break;  
 }  
 }  
 });  
  
 listView.setOnMousePressed(new EventHandler<MouseEvent>() {  
 @Override  
 public void handle(MouseEvent event) {  
 System.out.print("Item" + " " +listView.getSelectionModel().getSelectedItem() );  
 }  
 });  
 /\* For a simple ListView as ours, this method can be omitted  
 \* However I have included here to give you a hint for the next lessons  
 \*/  
 listView.setCellFactory(new Callback<ListView<String>, ListCell<String>>(){  
  
 @Override  
 public ListCell<String> call(ListView<String> p) {  
 final ListCell<String> cell=new ListCell<String>(){  
 @Override  
 public void updateItem(String item, boolean empty){  
 super.updateItem(item, empty);  
 if (item!=null){  
 setText(item);  
 }}};  
 return cell;  
 }});  
  
  
 listView.setItems(currentList);  
 currentList.add("ТРЬОХЧАСТОТНА ТРИГОНОМЕТРИЧНА КВАТЕРНІОННА МОДЕЛЬ \nОБЕРТАННЯ ТВЕРДОГО ТІЛА");  
 currentList.add("МУЛЬТИПЛІКАТИВНІ ТРЬОХЧАСТОТНІ МОДЕЛІ \nОБЕРТАННЯ ТВЕРДОГО ТІЛА");  
  
 viewGraphs.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
  
 position = 1;  
 createNewWindow("K1 = 0.015 K2 = 0.025 K3 = 0.005");  
 }  
 });  
  
 viewGraph1.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 position = 2;  
 createNewWindow("K1 = 0.015 K2 = 0.025 K3 = 0.03");  
 }  
 });  
  
 viewGraph2.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 position = 3;  
 createNewWindow("K1 = 0.005 K2 = 0.025 K3 = 0.025");  
 }  
 });  
  
 viewGraph3.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 position = 4;  
 createNewWindow("K1 = 0.048 K2 = 0.012 K3 = 0.012");  
 }  
 });  
 }  
  
 public void setImages(int position, String[] arr)  
 {  
 int size = arr.length;  
 if (position == 1) {  
 k4Value.setVisible(true);  
 viewGraph3.setVisible(true);  
 Image[] image = new Image[size];  
 for (int i = 0; i < size; i++) {  
 image[i] = new Image(Main.class.getResourceAsStream(arr[i]));  
 }  
 omegaModel.setImage(image[0]);  
 lambdaModel.setImage(image[1]);  
 k1Value.setImage(image[2]);  
 k2Value.setImage(image[3]);  
 k3Value.setImage(image[4]);  
 k4Value.setImage(image[5]);  
 }  
 else if (position == 2)  
 {  
 Image[] image = new Image[size];  
 for (int i = 0; i < size; i++) {  
 image[i] = new Image(Main.class.getResourceAsStream(arr[i]));  
 }  
 omegaModel.setImage(image[0]);  
 lambdaModel.setImage(image[1]);  
 k1Value.setImage(image[2]);  
 k2Value.setImage(image[3]);  
 k3Value.setImage(image[4]);  
 k4Value.setVisible(false);  
 viewGraph3.setVisible(false);  
 }  
 }  
 public void createNewWindow(String title)  
 {  
 Parent root;  
 try {  
 root = FXMLLoader.load(getClass().getResource("layout\_view\_pager.fxml"));  
 Stage stage = new Stage();  
 stage.setTitle(title);  
 stage.setScene(new Scene(root, 450, 450));  
 stage.show();  
 // Hide this current window (if this is what you want)  
 }  
 catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

package sample;  
  
import javafx.beans.value.ChangeListener;  
import javafx.beans.value.ObservableValue;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.fxml.Initializable;  
import javafx.scene.Node;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.ListCell;  
import javafx.scene.control.ListView;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.stage.Stage;  
import javafx.util.Callback;  
  
import java.io.IOException;  
import java.net.URL;  
import java.util.ResourceBundle;  
  
  
public class ControllerPager implements Initializable{  
  
 int size = 6, steps = 0;  
  
 @FXML  
 private Button leftButton;  
  
 @FXML  
 private Button rightButton;  
  
 @FXML ImageView viewImage;  
  
 String [] path = {"third\_point\_model/graphsK1/K1Pick1.png", "third\_point\_model/graphsK1/K1Pick2.png", "third\_point\_model/graphsK1/K1Pick3.png",  
 "third\_point\_model/graphsK1/K1Pick4.png", "third\_point\_model/graphsK1/K1Pick5.png", "third\_point\_model/graphsK1/K1Pick6.png"};  
  
 String [] path1 = {"third\_point\_model/graphsK2/K2Pick1.png", "third\_point\_model/graphsK2/K2Pick2.png", "third\_point\_model/graphsK2/K2Pick3.png",  
 "third\_point\_model/graphsK2/K2Pick4.png", "third\_point\_model/graphsK2/K2Pick5.png", "third\_point\_model/graphsK2/K2Pick6.png"};  
  
 String [] path2 = {"third\_point\_model/graphsK3/K3Pick1.png", "third\_point\_model/graphsK3/K3Pick2.png", "third\_point\_model/graphsK3/K3Pick3.png",  
 "third\_point\_model/graphsK3/K3Pick4.png", "third\_point\_model/graphsK3/K3Pick5.png", "third\_point\_model/graphsK3/K3Pick6.png"};  
  
 String [] path3 = {"third\_point\_model/graphsK4/K4Pick1.png", "third\_point\_model/graphsK4/K4Pick2.png", "third\_point\_model/graphsK4/K4Pick3.png",  
 "third\_point\_model/graphsK4/K4Pick4.png", "third\_point\_model/graphsK4/K4Pick5.png", "third\_point\_model/graphsK4/K4Pick6.png"};  
  
 String [] path12 = {"multi\_third\_point\_model/graphsK1/K1Pick1.png", "multi\_third\_point\_model/graphsK1/K1Pick2.png", "multi\_third\_point\_model/graphsK1/K1Pick3.png",  
 "multi\_third\_point\_model/graphsK1/K1Pick4.png", "multi\_third\_point\_model/graphsK1/K1Pick5.png", "multi\_third\_point\_model/graphsK1/K1Pick6.png"};  
  
 String [] path22 = {"multi\_third\_point\_model/graphsK2/K2Pick1.png", "multi\_third\_point\_model/graphsK2/K2Pick2.png", "multi\_third\_point\_model/graphsK2/K2Pick3.png",  
 "multi\_third\_point\_model/graphsK2/K2Pick4.png", "multi\_third\_point\_model/graphsK2/K2Pick5.png", "multi\_third\_point\_model/graphsK2/K2Pick6.png"};  
  
 String [] path32 = {"multi\_third\_point\_model/graphsK3/K3Pick1.png", "multi\_third\_point\_model/graphsK3/K3Pick2.png", "multi\_third\_point\_model/graphsK3/K3Pick3.png",  
 "multi\_third\_point\_model/graphsK3/K3Pick4.png", "multi\_third\_point\_model/graphsK3/K3Pick5.png", "multi\_third\_point\_model/graphsK3/K3Pick6.png"};  
 @Override  
 public void initialize(URL url, ResourceBundle rb) {  
  
  
  
 int tapPosition = Controller.position;  
 int model = Controller.choiceModel;  
 String[] tempPath = new String[size];  
  
 if (model == 1) {  
 switch (tapPosition) {  
 case 1:  
 tempPath = path;  
 break;  
 case 2:  
 tempPath = path1;  
 break;  
 case 3:  
 tempPath = path2;  
 break;  
 case 4:  
 tempPath = path3;  
 break;  
  
 }  
 }  
 else if (model == 2)  
 {  
 switch (tapPosition) {  
 case 1:  
 tempPath = path12;  
 break;  
 case 2:  
 tempPath = path22;  
 break;  
 case 3:  
 tempPath = path32;  
 break;  
 }  
 }  
  
 Image[] image = new Image[6];  
  
 for (int i = 0; i < size; i++) {  
 image[i] = new Image(Main.class.getResourceAsStream(tempPath[i]));  
 }  
  
 viewImage.setImage(image[0]);  
  
 rightButton.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 if (steps < 5) {  
 steps++;  
 viewImage.setImage(image[steps]);  
 }  
  
 }  
 });  
  
 leftButton.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 if (steps > 0)  
 {  
 steps--;  
 viewImage.setImage(image[steps]);  
 }  
 }  
 });  
 }  
}