For: Melissa Bakke

Assignment: Exercise 24-7 LinkedList Animation

|  |
| --- |
| **Screenshot(s)** |
|  |
|  |

|  |
| --- |
| **Code** |
| *import java.util.LinkedList;*  *import javafx.application.Application;*  *import javafx.geometry.Pos;*  *import javafx.scene.Scene;*  *import javafx.scene.control.Button;*  *import javafx.scene.control.Label;*  *import javafx.scene.control.TextField;*  *import javafx.scene.layout.BorderPane;*  *import javafx.scene.layout.HBox;*  *import javafx.scene.layout.Pane;*  *import javafx.scene.paint.Color;*  *import javafx.scene.shape.Line;*  *import javafx.scene.shape.Rectangle;*  *import javafx.scene.text.Text;*  *import javafx.stage.Stage;*  */\*\**  *\* Class: LinkedListAnimation*  *\* Developer: Melissa Bakke*  *\* Date: 03/28/2017*  *\* Purpose: Program to animate search, insertion, and deletion in a linked list.*  *\*/*  *public class LinkedListAnimation extends Application {*  *private LinkedList<Integer> list = new LinkedList<>();*  *private LinkedListView view = new LinkedListView();*  *private Button btSearch = new Button("Search");*  *private Button btInsert = new Button("Insert");*  *private Button btDelete = new Button("Delete");*  *private TextField tfNumber = new TextField();*  *private TextField tfIndex = new TextField();*    *@Override*  *public void start(Stage primaryStage){*  *HBox hBox = new HBox(5);*  *hBox.getChildren().addAll(new Label ("Enter a value: "),tfNumber, new Label("Enter an index: "), tfIndex,btSearch,btInsert,btDelete);*  *hBox.setAlignment(Pos.CENTER);*    *BorderPane borderPane = new BorderPane();*  *borderPane.setCenter(view);*  *borderPane.setBottom(hBox);*  *Label lblStatus = new Label();*  *borderPane.setTop(lblStatus);*  *borderPane.setAlignment(lblStatus,Pos.CENTER);*    *Scene scene = new Scene(borderPane,500,250);*  *primaryStage.setTitle("Exercise 24\_07: Linked Animation");*  *primaryStage.setScene(scene);*  *primaryStage.show();*  *view.repaint();*  *tfNumber.setPrefColumnCount(2);*  *tfIndex.setPrefColumnCount(2);*  *btSearch.setOnAction(e -> {*  *lblStatus.setText("");*  *if (!list.contains(Integer.parseInt(tfNumber.getText()))) {*  *lblStatus.setText("This Key is not in the list!");*  *}*  *else{*  *lblStatus.setText("This Key is in the list!");*  *}*  *view.repaint();*  *});*    *btInsert.setOnAction(e -> {*  *lblStatus.setText("");*  *if (tfIndex.getText().trim().length() > 0)*  *list.add(Integer.parseInt(tfIndex.getText()),*  *Integer.parseInt(tfNumber.getText()));*  *else*  *list.add(Integer.parseInt(tfNumber.getText()));*  *view.repaint();*  *});*    *btDelete.setOnAction(e -> {*  *lblStatus.setText("");*  *if (!list.contains(Integer.parseInt(tfNumber.getText()))) {*  *lblStatus.setText("This Key is not in the list!");*  *}*  *else {*  *lblStatus.setText("Key is deleted from the list");*  *list.remove(new Integer(Integer.parseInt(tfNumber.getText())));*  *view.repaint();*  *}*  *});*  *}*  *public static void main(String[] args){*  *launch(args);*  *}*  *public class LinkedListView extends Pane{*  *private int startingX = 20;*  *private int startingY = 20;*  *private int boxWidth = 50;*  *private int boxHeight = 20;*  *private int arrowLineLength = 30;*  *private int hGap = 80;*  *protected void repaint() {*  *getChildren().clear();*  *if (list.size() == 0) {*  *getChildren().add(new Text(startingX, startingY, "head: doesn't exist yet"));*  *getChildren().add(new Text(startingX, startingY + 15, "tail: doesn't exist yet"));*  *}*  *else {*  *getChildren().add(new Text(startingX, startingY, "head"));*  *int x = startingX + 30;*  *int y = startingY + 20;*  *drawArrowLine(startingX + 5, startingY, x, y);*  *for (int i = 0; i < list.size(); i++) {*  *Rectangle rectangle = new Rectangle(x, y, boxWidth, boxHeight);*  *rectangle.setFill(Color.WHITE);*  *rectangle.setStroke(Color.BLACK);*  *getChildren().add(rectangle);*  *getChildren().add(new Line(x + arrowLineLength, y, x + arrowLineLength, y + boxHeight));*  *if (i < list.size() - 1) {*  *drawArrowLine(x + 40, y + boxHeight / 2, x + hGap, y + boxHeight / 2);*  *}*  *getChildren().add(new Text(x + 10, y + 15, list.get(i) + ""));*  *x = x + hGap;*  *}*  *getChildren().add(new Text(x, startingY, "tail"));*  *drawArrowLine(x, startingY, x - hGap, y);*  *}*  *}*  *public void drawArrowLine(double x1, double y1, double x2, double y2) {*  *getChildren().add(new Line(x1, y1, x2, y2));*  *// Find slope of this line*  *double slope = ((((double) y1) - (double) y2)) / (((double) x1) - (((double) x2)));*  *double arctan = Math.atan(slope);*  *// This will flip the arrow 45 off of a perpendicular line at pt x2*  *double set45 = 1.57 / 2;*  *// Arrows should always point towards i, not i + 1*  *if (x1 < x2) {*  *// Add 90 degreed to arrow lines*  *set45 = -1.57 \* 1.5;*  *}*  *// Set length of arrows*  *int arrlen = 15;*  *// Draw arrows on line*  *getChildren().add(new Line(x2, y2, (x2 + (Math.cos(arctan + set45) \* arrlen)), ((y2)) + (Math.sin(arctan + set45) \* arrlen)));*  *getChildren().add(new Line(x2, y2, (x2 + (Math.cos(arctan - set45) \* arrlen)), ((y2)) + (Math.sin(arctan - set45) \* arrlen)));*  *}// End drawArrowLine*  *} // End linkedListView*  *} // End LinkedListAnimation* |
|  |