Divyajitsinh Mahida

Human Computer Interface

Final Project

MainFrame.Java

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this template

\*/

package projectfinal;

/\*\*

\*

\* @author Divyajitsinh

\*/

public class MainFrame extends javax.swing.JPanel {

/\*\*

\* Creates new form MainFrame

\*/

public MainFrame() {

initComponents();

waypointPlaceholder.add(ParkAdventures.waypointsPanel);

routePlaceholder.add(ParkAdventures.routesPanel);

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jScrollPane2 = new javax.swing.JScrollPane();

routePlaceholder = new javax.swing.JPanel();

jPanel1 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

waypointPlaceholder = new javax.swing.JPanel();

setPreferredSize(new java.awt.Dimension(667, 327));

routePlaceholder.setBorder(javax.swing.BorderFactory.createTitledBorder("Route"));

routePlaceholder.setLayout(new java.awt.BorderLayout());

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Waypoints"));

waypointPlaceholder.setLayout(new java.awt.BorderLayout());

jScrollPane1.setViewportView(waypointPlaceholder);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 493, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 294, Short.MAX\_VALUE)

.addContainerGap())

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(30, 30, 30)

.addComponent(routePlaceholder, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(routePlaceholder, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

getAccessibleContext().setAccessibleParent(this);

}// </editor-fold>//GEN-END:initComponents

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JPanel routePlaceholder;

private javax.swing.JPanel waypointPlaceholder;

// End of variables declaration//GEN-END:variables

}

ParkAdventures.Java

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template

\*/

package projectfinal;

import java.util.ArrayList;

import javax.swing.JFrame;

/\*\*

\*

\* @author Divyajitsinh

\*/

public class ParkAdventures {

public static ArrayList<Route> routes = new ArrayList<>();

public static ArrayList<Waypoint> waypoints = new ArrayList<>();

public static RoutesPanel routesPanel;

public static WaypointsPanel waypointsPanel;

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Route one, two, sh, lo;

Waypoint w;

one = new Route();

one.setName("Leisure Trail");

one.setBestTime("11:45 AM");

w = new Waypoint();

w.setName ("Deer Meadows");

w.setAvoid ("Do not chase the deer.");

w.setActivities ("Watch deer run and play in the meadows.");

w.setEastDistance (0.4);

w.setNorthDistance (4.9);

one.addWaypointAtEnd(w);

//save the waypoint in the repository

waypoints.add(w);

w = new Waypoint ();

w.setName("Redwood Run");

w.setAvoid("Do not climb trees");

w.setActivities("Walk through the shades of the Redwoods.");

w.setEastDistance(1.0);

w.setNorthDistance(3.5);

waypoints.add(w);

one.addWaypointAtEnd(w);

w = new Waypoint();

w.setName("Picnic Paradise");

w.setActivities("Enjoy your lunch.");

w.setAvoid("Please, do not litter and do not feed the wildlife.");

w.setEastDistance(0.6);

w.setNorthDistance(5.2);

waypoints.add(w);

one.addWaypointAtEnd(w);

routes.add(one); //save route in the repository

two = new Route();

two.setName("Fun Time Trail");

two.setBestTime("8:45 AM");

w = new Waypoint();

w.setName ("Floundering Fish Pond");

w.setAvoid ("Watch out for snakes.");

w.setActivities ("Feed fish only authorized food from feeder.");

w.setEastDistance (0.7);

w.setNorthDistance (2.9);

two.addWaypointAtEnd(w);

waypoints.add(w);

w = new Waypoint ();

w.setName("Camp Koala");

w.setAvoid("Do not climb trees");

w.setActivities("Koalas are not allowed human food.");

w.setEastDistance(1.9);

w.setNorthDistance(3.7);

waypoints.add(w);

two.addWaypointAtEnd(w);

w = new Waypoint();

w.setName("Nature's Picnic");

w.setActivities("Enjoy your lunch.");

w.setAvoid("Please, do not litter and do not feed the wildlife.");

w.setEastDistance(2.6);

w.setNorthDistance(5.4);

two.addWaypointAtEnd(w);

waypoints.add(w);

routes.add(two);

sh = new Route(); //created a new route ... it is empty now

sh.setName("Quick Route");

sh.setBestTime("11:45 AM");

w = new Waypoint(); //make and customize a new waypoint

w.setName("Walking Point");

w.setAvoid("Going on the track");

w.setActivities("Tracking");

w.setEastDistance(0.4);

w.setNorthDistance(4.9);

sh.addWaypointAtEnd(w); //add this waypoint to the route

//save the waypoint in the repository

waypoints.add(w);

w = new Waypoint(); //make and customize a new waypoint

w.setName("Red Beach");

w.setAvoid("Do not throw rocks");

w.setActivities("Swim");

w.setEastDistance(0.6);

w.setNorthDistance(5.9);

//save the waypoint in the repository

waypoints.add(w);

w=new Waypoint();

w.setName("Peak Mountain");

w.setAvoid("leaning on the bars");

w.setActivities("Nature View");

w.setEastDistance(0.7);

w.setNorthDistance(6.8);

waypoints.add(w);

sh.addWaypointAtEnd(w); //add this waypoint to the route

routes.add(sh); //save the route in the repository

lo = new Route();

lo.setName("Long Route");

lo.setBestTime("9:00 AM");

w = new Waypoint(); //make and customize a new waypoint

w.setName("Fish Point");

w.setAvoid("Do not throw anything in the water beside fish food");

w.setActivities("Feed Fish");

w.setEastDistance(4.2);

w.setNorthDistance(3.5);

lo.addWaypointAtEnd(w); //add this waypoint to the route

w = new Waypoint(); //make and customize a new waypoint

w.setName("Sky Point");

w.setAvoid("Moving too much");

w.setActivities("Ziplining");

w.setEastDistance(5.5);

w.setNorthDistance(2.5);

lo.addWaypointAtEnd(w);

w = new Waypoint(); //make and customize a new waypoint

w.setName("Water riding");

w.setAvoid("Keeping your hand out while riding");

w.setActivities("Water rafting");

w.setEastDistance(0.10);

w.setNorthDistance(3.5);

lo.addWaypointAtEnd(w);

//save the waypoint in the repository

waypoints.add(w);

routes.add(lo); //also add lo to the repository

//print New Route

// System.out.println("Route " + one.getName() + ": Best time to start at " +

// one.getBestTime() + " of length " + one.computeTotalLength());

//

// for (int i = 0; i < one.getSizeSequence(); i++){

// Waypoint x = one.getWaypoint(i);

// System.out.println("\t" + x.getName() + ": at (" + x.getEastDistance()

// + ", " + x.getNorthDistance() + ")");

// System.out.println("\t\tThings to do: " + x.getActivities());

// System.out.println( "\t\tThings to avoid: " + x.getRules());

//

// }

// System.out.println("\n\n----------------------------Alternate Route for Leisure Trail------------\n\n");

//

// {

//

// one.removeWaypoint(b);

// one.removeWaypoint(c);

// one.addWaypointAtEnd(f);

// one.addWaypointAtEnd(d);

//

// //print New Route

// System.out.println("Route " + one.getName() + ": Best time to start at " +

// one.getBestTime() + " of length " + one.computeTotalLength());

//

// for (int i = 0; i < one.getSizeSequence(); i++){

// Waypoint x = one.getWaypoint(i);

// System.out.println("\t" + x.getName() + ": at (" + x.getEastDistance()

// + ", " + x.getNorthDistance() + ")");

// System.out.println("\t\tThings to do: " + x.getActivities());

// System.out.println( "\t\tThings to avoid: " + x.getRules());

//

//

//

// }

//

//

// a.setEastDistance (0.4);

// System.out.println("\n\n---------------------------------------------\n\n");

//

//

// System.out.println("Route " + two.getName() + ": Best time to start is at "

// + two.getBestTime() + " of length " + two.computeTotalLength());

//

// for (int i = 0; i < two.getSizeSequence(); i++){

// Waypoint x = two.getWaypoint(i);

// System.out.println("\t" + x.getName() + ": at (" + x.getEastDistance()

// + ", " + x.getNorthDistance() + ")");

// System.out.println("\t\tThings to do: " + x.getActivities());

// System.out.println( "\t\tThings to avoid: " + x.getRules());

//

// }

//

// System.out.println("\n\n----------------------------Alternate Route for Fun Time Trail------------\n\n");

//

// two.removeWaypoint(d);

// two.removeWaypoint(e);

// two.removeWaypoint(f);

//

// two.addWaypointAtEnd(c);

// two.addWaypointAtEnd(f);

// two.addWaypointAtEnd(d);

//

// System.out.println("Route " + two.getName() + ": Best time to start is at "

// + two.getBestTime() + " of length " + two.computeTotalLength());

//

// for (int i = 0; i < two.getSizeSequence(); i++){

// Waypoint x = two.getWaypoint(i);

// System.out.println("\t" + x.getName() + ": at (" + x.getEastDistance()

// + ", " + x.getNorthDistance() + ")");

// System.out.println("\t\tThings to do: " + x.getActivities());

// System.out.println( "\t\tThings to avoid: " + x.getRules());

//

// }

//

// }

// routes.add(one);

// routes.add(two);

routesPanel = new RoutesPanel();

waypointsPanel = new WaypointsPanel();

MainFrame main = new MainFrame();

JFrame r = new JFrame("Funtime Park");

//f.setLocation (1000, 190);

r.setSize(1230, 450);

r.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

r.add(main);

r.setVisible(true);

}

}

Route.Java

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package projectfinal;

import java.util.ArrayList;

/\*\*

\*

\* @author Divyajitsinh

\*/

public class Route {

private String name;

private String bestTime;

private ArrayList<Waypoint> sequence = new ArrayList<>();

@Override

public String toString() {

String tmp = name;

tmp = tmp + " " + getBestTime();

double d = computeTotalLength();

d = Math.round(d \* 100) /100.0; //Keep 2 digits after decimal.

tmp = tmp + " " + d + " mi";

return tmp;

}

public int getSizeSequence() {

return sequence.size();

}

public Waypoint getWaypoint(int idx) {

return sequence.get(idx);

}

public int getIndex(Waypoint w){

return sequence.indexOf(w);

}

public void addWaypoint (int idx, Waypoint additional) {

sequence.add(idx, additional);

}

public void addWaypointAtEnd (Waypoint additional){

sequence.add(additional);

}

public void removeWaypoint(int idx){

sequence.remove(idx);

}

public void removeWaypoint(Waypoint whichOne) {

sequence.remove(whichOne);

}

public boolean hasWaypoint(Waypoint w){

if(sequence.contains (w))

return true;

else

return false;

}

public double computeTotalLength(){

if (sequence.size() < 2)

return 0;

double tmpLength = 0;

for (int i = 1; i < sequence.size(); i++){

//compute distance between two points

Waypoint prev, next;

double dist;

prev = sequence.get(i-1);

next = sequence.get (i);

dist = Math.sqrt((prev.getEastDistance() - next.getEastDistance()) \*

(prev.getEastDistance() - next.getEastDistance()) +

(prev.getNorthDistance() - next.getNorthDistance()) \*

(prev.getNorthDistance() - next.getNorthDistance()));

tmpLength = tmpLength + dist; //add distance between points

}

return tmpLength;

}

public String getName(){

return name;

}

public void setName (String name){

this.name = name;

}

public String getBestTime() {

return bestTime;

}

public void setBestTime(String bestTime) {

this.bestTime = bestTime;

}

}

RoutesPanel.Java

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this template

\*/

package projectfinal;

import javax.swing.DefaultListModel;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Divyajitsinh

\*/

public class RoutesPanel extends javax.swing.JPanel {

/\*\*

\* Creates new form RoutePanel

\*/

public RoutesPanel() {

initComponents();

refreshRoutesList();

}

public void refreshRoutesList(){

//add all the waypoints from the repository to my list

DefaultListModel model;

model = (DefaultListModel) routesList.getModel(); //this is the data that the jList shows

//first remove any old information from that list

model.clear();

// //add all waypoints from the repository

// for(int i = 0; i < ProjectFinal.routes.size(); i++){

// Waypoint w;

// w = ProjectFinal.waypoint.get(i); //the the i-th waypoint

// model.addElement(w); //add it to the model of the list

// }

//

//add routes from the repository

for(int i = 0; i < ParkAdventures.routes.size(); i++) {

Route r;

r = ParkAdventures.routes.get(i); //the the i-th waypoint

model.addElement(r); //add it to the model of the list

}

}

//the w is the waypoint selected in the WaypointsPanel

public void updateAddWaypointButton(Waypoint w){

if(w == null){ //no waypoint selected in the other panel

addWaypointButton.setText("Add Waypoint");

addWaypointButton.setEnabled(false);

}

else{

String name = w.getName();

addWaypointButton.setText("Add " + name); //change the name displayed on the button

//check to see if the add button should be enabled

if(routesList.getSelectedValue() == null) //no route selected

addWaypointButton.setEnabled(false);

else

addWaypointButton.setEnabled(true);

}

}

public void refreshWaypointList(){

DefaultListModel model;

model = (DefaultListModel) waypointsList.getModel();

model.clear();

}

public void updateWaypointList(Route r){

//populate the waypoints jlist

DefaultListModel wModel = (DefaultListModel) waypointsList.getModel();

wModel.clear(); //remove any existing elements from the list. The correct ones will be added in the for loop

for(int i = 0; i < r.getSizeSequence(); i++){

Waypoint w = r.getWaypoint(i);

wModel.addElement(w); //add the i-th waypoint to the list

}

}

public void updateDeleteButton () {

String name, time;

Route r = this.routesList.getSelectedValue();

name = nameField.getText();

time = timeField.getText();

if(name.equals("") || time.equals("")){

//modifyButton.setEnabled(false); //Disable the button when fields are blank

deleteRouteButton.setEnabled(false);

return;

}

if(r == null){

// modifyButton.setEnabled(false);

deleteRouteButton.setEnabled(false);

return;

}

//if program gets here, everything is good.

//modifyButton.setEnabled(true);

deleteRouteButton.setEnabled(true);

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jScrollPane2 = new javax.swing.JScrollPane();

routesList = new javax.swing.JList<>();

jLabel2 = new javax.swing.JLabel();

addRouteButton = new javax.swing.JButton();

routeNameText = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

deleteRouteButton = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

waypointsList = new javax.swing.JList<>();

addWaypointButton = new javax.swing.JButton();

removeWaypointButton = new javax.swing.JButton();

nameField = new javax.swing.JTextField();

timeField = new javax.swing.JTextField();

downWaypointButton = new javax.swing.JButton();

upWaypointButton = new javax.swing.JButton();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

routeTimeText = new javax.swing.JTextField();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

modifyButton = new javax.swing.JButton();

routesList.setModel(new DefaultListModel());

routesList.addListSelectionListener(new javax.swing.event.ListSelectionListener() {

public void valueChanged(javax.swing.event.ListSelectionEvent evt) {

routesListValueChanged(evt);

}

});

jScrollPane2.setViewportView(routesList);

jLabel2.setText("Route List");

jLabel2.setVerticalAlignment(javax.swing.SwingConstants.TOP);

addRouteButton.setText("Add Route");

addRouteButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addRouteButtonActionPerformed(evt);

}

});

routeNameText.addCaretListener(new javax.swing.event.CaretListener() {

public void caretUpdate(javax.swing.event.CaretEvent evt) {

routeNameTextCaretUpdate(evt);

}

});

routeNameText.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

routeNameTextActionPerformed(evt);

}

});

jLabel3.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N

jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel3.setText("Route Editor");

deleteRouteButton.setText("Delete Route");

deleteRouteButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

deleteRouteButtonActionPerformed(evt);

}

});

waypointsList.setModel(new DefaultListModel());

waypointsList.addListSelectionListener(new javax.swing.event.ListSelectionListener() {

public void valueChanged(javax.swing.event.ListSelectionEvent evt) {

waypointsListValueChanged(evt);

}

});

jScrollPane1.setViewportView(waypointsList);

addWaypointButton.setText("Add Waypoint");

addWaypointButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addWaypointButtonActionPerformed(evt);

}

});

removeWaypointButton.setText("Remove Waypoint");

removeWaypointButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

removeWaypointButtonActionPerformed(evt);

}

});

nameField.addCaretListener(new javax.swing.event.CaretListener() {

public void caretUpdate(javax.swing.event.CaretEvent evt) {

nameFieldCaretUpdate(evt);

}

});

timeField.addCaretListener(new javax.swing.event.CaretListener() {

public void caretUpdate(javax.swing.event.CaretEvent evt) {

timeFieldCaretUpdate(evt);

}

});

downWaypointButton.setText("Down");

downWaypointButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

downWaypointButtonActionPerformed(evt);

}

});

upWaypointButton.setText("Up");

upWaypointButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

upWaypointButtonActionPerformed(evt);

}

});

jLabel4.setText("Enter new route name");

jLabel5.setText("Enter best time to begin");

jLabel6.setText("Edit Route Name");

jLabel7.setText("Edit Begin Time");

jLabel8.setText("Route's Waypoint List");

modifyButton.setText("Modify");

modifyButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

modifyButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(51, 51, 51)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 243, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addComponent(addRouteButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(deleteRouteButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(routeNameText, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel4))

.addGroup(layout.createSequentialGroup()

.addComponent(routeTimeText, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel5)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(nameField, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel6))

.addGap(42, 42, 42)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel7)

.addGroup(layout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(timeField, javax.swing.GroupLayout.PREFERRED\_SIZE, 76, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(addWaypointButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 117, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(removeWaypointButton))

.addGap(45, 45, 45)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(upWaypointButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 61, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(downWaypointButton)))

.addComponent(jLabel8)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createSequentialGroup()

.addGap(109, 109, 109)

.addComponent(modifyButton))))

.addComponent(jLabel2)

.addComponent(jLabel3))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 21, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel6)

.addComponent(jLabel7))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(nameField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(timeField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(4, 4, 4)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(routeNameText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(routeTimeText, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel5))

.addGap(0, 0, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(modifyButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel8)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 130, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(addRouteButton)

.addComponent(deleteRouteButton)

.addComponent(addWaypointButton)

.addComponent(upWaypointButton))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(removeWaypointButton)

.addComponent(downWaypointButton)))

);

}// </editor-fold>//GEN-END:initComponents

private void routeNameTextCaretUpdate(javax.swing.event.CaretEvent evt) {//GEN-FIRST:event\_routeNameTextCaretUpdate

System.out.println("Caret event triggered");

String s;

s = this.routeNameText.getText(); //get what the user has typed in the textfield

this.jLabel2.setText("Route");

}//GEN-LAST:event\_routeNameTextCaretUpdate

private void addRouteButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_addRouteButtonActionPerformed

//System.out.println("Add button triggered");

// Get selected route:

String name, time;

name = routeNameText.getText();

time = routeTimeText.getText();

//update the route

Route newr = new Route();

newr.setName(name);

newr.setBestTime(time);

routesList.repaint();

// String routeName = routeNameText.getText();

// String routeTime = routeTimeText.getText();

//

// DefaultListModel model; //this is the data shown by the list

// model = (DefaultListModel) routeList.getModel(); //get the data for that list

//

//add the text in the textfiedl to the model

// String t = routeNameText.getText(); //this is what the user typed in

// if(!t.equals("")) //check if the text is empty

// model.addElement(t); //add the text at the end of the list // TODO add your handling code here:

//

//create a new route

// Route r = new Route();

// r.setName(routeName);

// r.setBestTime(routeTime);

//

//ensure name not in use already

for(int i = 0; i <ParkAdventures.routes.size(); i++){

Route g;

g = ParkAdventures.routes.get(i);

if(routeNameText.equals((g.getName()))){

JOptionPane.showMessageDialog(this, "A route with this name already exists", "Add new route",

JOptionPane.WARNING\_MESSAGE);

return;

}

}

//add it to the repository

ParkAdventures.routes.add(newr);

//clear the textfields

routeNameText.setText("");

routeTimeText.setText("");

//refresh interface to show new list

refreshRoutesList();

}//GEN-LAST:event\_addRouteButtonActionPerformed

private void routesListValueChanged(javax.swing.event.ListSelectionEvent evt) {//GEN-FIRST:event\_routesListValueChanged

Route r = routesList.getSelectedValue();

updateDeleteButton();

//update the add waypoint button

Waypoint w = ParkAdventures.waypointsPanel.getSelectedWaypoint();

updateAddWaypointButton(w);

if (r == null){

nameField.setText("");

timeField.setText("");

DefaultListModel wModel = (DefaultListModel) waypointsList.getModel();

wModel.clear();

return;

}

nameField.setText(r.getName());

timeField.setText(r.getBestTime());

//populate the waypoint list

updateWaypointList(r);

//refreshRoutesList();

//System.out.println("List Selection Changed");

//String whatsSelected = "Selected: ";

// DefaultListModel model; //this is the data shown by the list

// model = (DefaultListModel) routeList.getModel(); //get the data for that list

//

// int[] indices = routeList.getSelectedIndices(); //get the indices of all selected elements

// if(indices.length == 0){

// rSelectedLabel.setText("Nothing selected"); //nothing is selected

// return; //get out of this function ... basically ignore the for loop below...

// }

//

// //go thrugh all selected items

// for(int i = 0; i < indices.length; i++)

// whatsSelected = whatsSelected + " " + model.get(indices[i]); //add the i-th selected element to the string

//

// rSelectedLabel.setText(whatsSelected);// TODO add your handling code here:

}//GEN-LAST:event\_routesListValueChanged

private void deleteRouteButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_deleteRouteButtonActionPerformed

Route r = routesList.getSelectedValue();

ParkAdventures.routes.remove(r);

refreshRoutesList();

ParkAdventures.waypointsPanel.refreshWaypointList();

DefaultListModel model; //this is the data shown by the list

model = (DefaultListModel) routesList.getModel(); //get the data for that list

int[] indices = routesList.getSelectedIndices(); //get the indices of all selected elements

//go thrugh all selected items, starting at the end

for(int i = indices.length - 1; i >= 0 ; i--) //decrement i to move towards the beginning of the array

model.remove(indices[i]);// TODO add your handling code here:

}//GEN-LAST:event\_deleteRouteButtonActionPerformed

private void routeNameTextActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_routeNameTextActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_routeNameTextActionPerformed

private void addWaypointButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_addWaypointButtonActionPerformed

Waypoint w = ParkAdventures.waypointsPanel.getSelectedWaypoint();

Route r = routesList.getSelectedValue();//get the selected route

r.addWaypointAtEnd(w);

int idx = routesList.getAnchorSelectionIndex();

//ensure name not in use already

for(int i = 0; i <ParkAdventures.routes.size(); i++){

Route g;

g = ParkAdventures.routes.get(i);

if(routeNameText.equals((g.getName()))){

JOptionPane.showMessageDialog(this, "A route with this name already exists", "Add new route",

JOptionPane.WARNING\_MESSAGE);

return;

}

refreshRoutesList();

routesList.setSelectedIndex(idx);

DefaultListModel model; //this is the data shown by the list

model = (DefaultListModel) waypointsList.getModel(); //get the data for that list

//add the text in the textfiedl to the model

// String t = waypointText.getText(); //this is what the user typed in

// if(!t.equals("")) //check if the text is empty

// model.addElement(t); //add the text at the end of the list // TODO add your handling code here:

//

}//GEN-LAST:event\_addWaypointButtonActionPerformed

}

private void nameFieldCaretUpdate(javax.swing.event.CaretEvent evt) {//GEN-FIRST:event\_nameFieldCaretUpdate

// Check the name and if empty disable Modify button

//updateModifyButton();

}//GEN-LAST:event\_nameFieldCaretUpdate

private void timeFieldCaretUpdate(javax.swing.event.CaretEvent evt) {//GEN-FIRST:event\_timeFieldCaretUpdate

updateDeleteButton();

}//GEN-LAST:event\_timeFieldCaretUpdate

private void downWaypointButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_downWaypointButtonActionPerformed

// Get the selected route and waypoint

Route r = routesList.getSelectedValue();

Waypoint w;

int idx = waypointsList.getSelectedIndex();

w = r.getWaypoint(idx);

r.removeWaypoint (idx);

r.addWaypoint(idx + 1, w);

updateWaypointList(r);

refreshRoutesList();

}//GEN-LAST:event\_downWaypointButtonActionPerformed

private void waypointsListValueChanged(javax.swing.event.ListSelectionEvent evt) {//GEN-FIRST:event\_waypointsListValueChanged

Waypoint w = waypointsList.getSelectedValue();

int idx = waypointsList.getSelectedIndex();

int nElements = waypointsList.getModel().getSize();

if(w == null) {

upWaypointButton.setEnabled(false);

downWaypointButton.setEnabled(false);

removeWaypointButton.setEnabled(false);

}

else{

if(idx > 0)//is it the first element in the list

upWaypointButton.setEnabled(true);

else

upWaypointButton.setEnabled(false);

if(idx <nElements - 1) //is this the last element???

downWaypointButton.setEnabled(true);

else

downWaypointButton.setEnabled(false);

removeWaypointButton.setEnabled(true);

}

}//GEN-LAST:event\_waypointsListValueChanged

private void upWaypointButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_upWaypointButtonActionPerformed

//get the selected route and waypoint

Route r = routesList.getSelectedValue();//get selected route

Waypoint w;

int idx = waypointsList.getSelectedIndex();

w = r.getWaypoint(idx);

r.removeWaypoint(idx);//remove waypoint

r.addWaypoint(idx - 1, w); //add it one positionhigher

updateWaypointList(r); //update list to show new position

refreshRoutesList();

}//GEN-LAST:event\_upWaypointButtonActionPerformed

private void removeWaypointButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_removeWaypointButtonActionPerformed

Route r = routesList.getSelectedValue();

Waypoint s;

int idx = waypointsList.getSelectedIndex();

s = r.getWaypoint(idx);

r.removeWaypoint(idx);

updateWaypointList(r);

DefaultListModel model; //this is the data shown by the list

model = (DefaultListModel) waypointsList.getModel(); //get the data for that list

int[] indices = waypointsList.getSelectedIndices(); //get the indices of all selected elements

//go thrugh all selected items, starting at the end

for(int i = indices.length - 1; i >= 0 ; i--) //decrement i to move towards the beginning of the array

model.remove(indices[i]);// TODO add your handling code here:

refreshRoutesList();

}//GEN-LAST:event\_removeWaypointButtonActionPerformed

private void modifyButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_modifyButtonActionPerformed

Route r = routesList.getSelectedValue(); //get the selected route

String name, time;

name = nameField.getText(); //this is what the user entered

time = timeField.getText(); //this is what the user entered

//update the route

r.setName(name);

r.setBestTime(time);

routesList.repaint(); //force a repaint of the route to show the new name or time

}//GEN-LAST:event\_modifyButtonActionPerformed

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton addRouteButton;

private javax.swing.JButton addWaypointButton;

private javax.swing.JButton deleteRouteButton;

private javax.swing.JButton downWaypointButton;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JButton modifyButton;

private javax.swing.JTextField nameField;

private javax.swing.JButton removeWaypointButton;

private javax.swing.JTextField routeNameText;

private javax.swing.JTextField routeTimeText;

private javax.swing.JList<Route> routesList;

private javax.swing.JTextField timeField;

private javax.swing.JButton upWaypointButton;

private javax.swing.JList<Waypoint> waypointsList;

// End of variables declaration//GEN-END:variables

}

Waypoint.Java

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package projectfinal;

/\*\*

\*

\* @author Divyajitsinh

\*/

public class Waypoint {

private String name;

private String activities;

private String avoid;

private double eastDistance;

private double northDistance;

@Override

public String toString(){

String tmp = "\* ";

//check to see if waypoint is in use

for(int i = 0; i <ParkAdventures.routes.size(); i++){

Route r;

r = ParkAdventures.routes.get(i);

if (r.hasWaypoint(this)){ //is 'this' inside Route?

tmp = "";

break; //this waypoint is used so stop searching

}

}

tmp = tmp + name + ": (" + eastDistance + ", " + northDistance + ")";

tmp = tmp + " " + activities;

tmp = tmp + ", " + avoid;

return tmp;

}

public String getName(){

return name;

}

public void setName(String name) {

this.name = name;

}

public String getActivities() {

return activities;

}

public void setActivities(String activities) {

this.activities = activities;

}

public String getAvoid() {

return avoid;

}

public void setAvoid(String Rules) {

this.avoid = Rules;

}

public double getEastDistance() {

return eastDistance;

}

public void setEastDistance(double eastDistance) {

this.eastDistance = eastDistance;

}

public double getNorthDistance() {

return northDistance;

}

public void setNorthDistance(double northDistance) {

this.northDistance = northDistance;

}

}

WaypointsPanel.Java

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this template

\*/

package projectfinal;

import java.awt.Color;

import javax.swing.DefaultListModel;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Divyajitsinh

\*/

public class WaypointsPanel extends javax.swing.JPanel {

/\*\*

\* Creates new form waypointPanel

\*/

public WaypointsPanel() {

initComponents();

//add all waypoints from repository

refreshWaypointList();

}

//return waypoint selected in waypointList

public Waypoint getSelectedWaypoint() {

Waypoint w = waypointsList.getSelectedValue();

return w;

}

public void refreshWaypointList() {

//add all waypoints from repository to list

DefaultListModel model;

model = (DefaultListModel) waypointsList.getModel(); //data shown in jList

model.clear(); //REmove anything in the list

//add all waypoints from repository

for(int i = 0; i < ParkAdventures.waypoints.size(); i++){

Waypoint w;

w = ParkAdventures.waypoints.get(i);

model.addElement(w);

}

}

public void updateModifyButton(){

String name, east, north, act, rls;

Waypoint x = this.waypointsList.getSelectedValue();

name = nameField.getText(); //this is what the user entered

east = eastField.getText();//this is what the user entered

north = northField.getText();

act = activities.getText();

rls = rules.getText();

if(name.equals("") || north.equals("")){

modifyButton.setEnabled(true); //diasble the button if the name or time are empty

return;

}

if(x == null){

modifyButton.setEnabled(false); //diasble the button if nothing is selected

return;

}

//if my program gets here, everything is fine

modifyButton.setEnabled(true);

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jLabel2 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jLabel1 = new javax.swing.JLabel();

waypointLocationLabel = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

eastField = new javax.swing.JTextField();

northField = new javax.swing.JTextField();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

activities = new javax.swing.JTextField();

jLabel7 = new javax.swing.JLabel();

rules = new javax.swing.JTextField();

nameField = new javax.swing.JTextField();

jPanel2 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

waypointsList = new javax.swing.JList<>();

deleteButton = new javax.swing.JButton();

addButton = new javax.swing.JButton();

modifyButton = new javax.swing.JButton();

jLabel2.setText("jLabel2");

jTextField1.setText("jTextField1");

setBorder(javax.swing.BorderFactory.createTitledBorder("Waypoint Editor"));

jLabel1.setText("Waypoint Name");

waypointLocationLabel.setText("Location (mi) ");

jLabel3.setText("E");

jLabel4.setText("N");

eastField.setToolTipText("");

jLabel5.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N

jLabel6.setText("Activities to Do");

jLabel7.setText("Avoid");

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(), "All Waypoints"));

waypointsList.setModel(new DefaultListModel());

waypointsList.addListSelectionListener(new javax.swing.event.ListSelectionListener() {

public void valueChanged(javax.swing.event.ListSelectionEvent evt) {

waypointsListValueChanged(evt);

}

});

jScrollPane1.setViewportView(waypointsList);

deleteButton.setText("Delete");

deleteButton.setToolTipText("Delete Waypoint from list");

deleteButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

deleteButtonActionPerformed(evt);

}

});

addButton.setText("Add");

addButton.setToolTipText("Add Waypoint to list");

addButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addButtonActionPerformed(evt);

}

});

modifyButton.setText("Modify");

modifyButton.setToolTipText("Modify A Waypoint");

modifyButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

modifyButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addComponent(addButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 102, Short.MAX\_VALUE)

.addComponent(deleteButton)))

.addContainerGap())

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(81, 81, 81)

.addComponent(modifyButton)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 176, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(deleteButton)

.addComponent(addButton))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(modifyButton)

.addGap(0, 13, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);

this.setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 137, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel6)

.addComponent(jLabel7)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(rules, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(activities, javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel4)

.addGap(18, 18, 18)

.addComponent(northField, javax.swing.GroupLayout.DEFAULT\_SIZE, 217, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel3)

.addGap(18, 18, 18)

.addComponent(eastField))

.addComponent(jLabel1, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(waypointLocationLabel, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(nameField, javax.swing.GroupLayout.Alignment.LEADING)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(jLabel5))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGap(39, 39, 39)

.addComponent(nameField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(27, 27, 27)

.addComponent(waypointLocationLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3)

.addComponent(eastField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(northField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(activities, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jLabel7)

.addGap(18, 18, 18)

.addComponent(rules, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

}// </editor-fold>//GEN-END:initComponents

private void waypointsListValueChanged(javax.swing.event.ListSelectionEvent evt) {//GEN-FIRST:event\_waypointsListValueChanged

Waypoint w = waypointsList.getSelectedValue();

ParkAdventures.routesPanel.updateAddWaypointButton(w);

updateModifyButton();

if(w == null) {//nothing is selected from list

nameField.setText("");

eastField.setText("");

northField.setText("");

activities.setText("");

rules.setText("");

DefaultListModel wModel = (DefaultListModel) waypointsList.getModel();

wModel.clear();

return;

}

nameField.setText(w.getName());

activities.setText(w.getActivities());

rules.setText(w.getAvoid());

}//GEN-LAST:event\_waypointsListValueChanged

private void deleteButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_deleteButtonActionPerformed

Waypoint w = waypointsList.getSelectedValue();

ParkAdventures.waypoints.remove(w);

ParkAdventures.waypointsPanel.refreshWaypointList();

}//GEN-LAST:event\_deleteButtonActionPerformed

private void addButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_addButtonActionPerformed

String name = nameField.getText(); //see what the user entered here

String es = eastField.getText(); //get the text entered by the user for East

String ns = northField.getText(); //same for North

String act = activities.getText();

String av = rules.getText();

//now see if it can be

double e;

double n;

try{

e = Double.parseDouble(es);

//at this point the conversion to double was successful - background goes back to white

eastField.setBackground(Color.white);

}

catch(Exception x){

System.out.println("Could not convert East textfield to a number");

eastField.setBackground(Color.RED); //make the textfield red

return;

}

try{

n = Double.parseDouble(ns);

//at this point the conversion to double was successful - background goes back to white

northField.setBackground(Color.white);

}

catch(Exception x){

System.out.println("Could not convert North textfield to a number");

northField.setBackground(Color.RED); //make the textfield red

return;

}

//create a new waypoint: new object of type Waypoint

Waypoint neww = new Waypoint();

neww.setName(name);

neww.setEastDistance(e);

neww.setNorthDistance(n);

neww.setActivities(act);

neww.setAvoid(av);

//before adding to the waypoint repository, make sure the name is not already in there

for(int i = 0; i <ParkAdventures.waypoints.size(); i++){

Waypoint w;

w = ParkAdventures.waypoints.get(i);

if(name.equals(w.getName())){

JOptionPane.showMessageDialog(this, "A waypoint with this name already exists", "Add waypoint",

JOptionPane.WARNING\_MESSAGE);

return; //do not continue this function

}

}

//add it to the repository

ParkAdventures.waypoints.add(neww);

//clear the textfields, so the waypoint that was just added does not show in there anymore

nameField.setText("");

eastField.setText("");

northField.setText("");

activities.setText("");

rules.setText("");

//refresh the interface to show the new repository

refreshWaypointList();

}//GEN-LAST:event\_addButtonActionPerformed

private void modifyButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_modifyButtonActionPerformed

Waypoint y = waypointsList.getSelectedValue();

String name, east, north, act, rls;

name = nameField.getText();

east = eastField.getText();

north = northField.getText();

act = activities.getText();

rls = rules.getText();

//now see if it can be

double es;

double ns;

try{

es = Double.parseDouble(east);

//at this point the conversion to double was successful - background goes back to white

eastField.setBackground(Color.white);

}

catch(Exception x){

System.out.println("Could not convert East textfield to a number");

eastField.setBackground(Color.RED); //make the textfield red

return;

}

try{

ns = Double.parseDouble(north);

//at this point the conversion to double was successful - background goes back to white

northField.setBackground(Color.white);

}

catch(Exception x){

System.out.println("Could not convert North textfield to a number");

northField.setBackground(Color.RED); //make the textfield red

return;

}

//update the waypoint

y.setName(name);

y.setEastDistance(es);

y.setNorthDistance(ns);

y.setActivities(act);

y.setAvoid(rls);

refreshWaypointList();

}//GEN-LAST:event\_modifyButtonActionPerformed

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JTextField activities;

private javax.swing.JButton addButton;

private javax.swing.JButton deleteButton;

private javax.swing.JTextField eastField;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JPanel jPanel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextField jTextField1;

private javax.swing.JButton modifyButton;

private javax.swing.JTextField nameField;

private javax.swing.JTextField northField;

private javax.swing.JTextField rules;

private javax.swing.JLabel waypointLocationLabel;

private javax.swing.JList<Waypoint> waypointsList;

// End of variables declaration//GEN-END:variables

}

Code Running SS

A screenshot of a computer

Description automatically generated