**Metody numeryczne**

Lista 8

Katarzyna Korsak 229707

Piątek 730

**Zadanie 1**

Zadaniem 1 było zapoznanie się z programem uruchamiającym animację przedstawiającą ruch po okręgu brązowego koła.

**Zadanie 2**

**2.1. Rozwiązanie zadania**

Rozwiązaniem zadania 2 są 2 klasy, których kod widoczny jest poniżej. W klasie *WahadloAnimacja* metoda *display()* odpowiada za wyświetlenie okna modalnego z polami do wpisania wartości początkowych i suwaków na 3 pierwszych parametrów. Można je przesuwać m.in. za pomocą klawiatury. Metoda *getValues()* jest identyczna jak w zadaniu z poprzedniej listy, dodano tylko statyczne *ArrayList* przechowujące wartości kąta theta i czasu, potrzebne i wykorzystywane w kolejnej z metod*, drawAnimation()*, która na wzór zadania 1 rysuje animację, w pętli wywołując jednak kolejne wartości *ArrayList* *– thetaList*. Metoda *setautoreverse* ustawiona została na *true*, by wahadło mogło poruszać się tam i z powrotem. W klasie *WahadloAnimacjaFX* ,względem zadania z poprzedniej listy, został dodany przycisk *Animation* i metoda *updateParam()*, która w głównym oknie nadpisuje wartości TextFieldów wartościami wpisanymi w oknie modalnym.

**Klasa WahadloAnimacja**

**import** javafx.animation.PathTransition;  
**import** javafx.beans.value.ChangeListener;  
**import** javafx.beans.value.ObservableValue;  
**import** javafx.geometry.Insets;  
**import** javafx.geometry.Pos;  
**import** javafx.scene.Group;  
**import** javafx.scene.Scene;  
**import** javafx.scene.chart.XYChart;  
**import** javafx.scene.control.Button;  
**import** javafx.scene.control.Label;  
**import** javafx.scene.control.Slider;  
**import** javafx.scene.control.TextField;  
**import** javafx.scene.layout.HBox;  
**import** javafx.scene.layout.VBox;  
**import** javafx.scene.paint.Color;  
**import** javafx.scene.shape.Circle;  
**import** javafx.scene.shape.LineTo;  
**import** javafx.scene.shape.MoveTo;  
**import** javafx.scene.shape.Path;  
**import** javafx.stage.Modality;  
**import** javafx.stage.Stage;  
**import** javafx.util.Duration;  
  
**import** java.util.ArrayList;  
  
**public class** WahadloAnimacja {  
  
 **private double A**;  
 **private double f**;  
 **private double c**;  
 **private static double** *theta0*;  
 **private double t0**;  
 **private double h**;  
 **private double tend**;  
 **private double omega0**;  
  
 **public static** ArrayList<Double> *thetaList*;  
 **public static** ArrayList<Double> *timeList*;  
  
 **public** WahadloAnimacja() {  
 **c** = 0.5;  
 **f** = 1;  
 **A** = 0.5;  
 *theta0* = 5;  
 **omega0** = 0;  
 **t0** = 0;  
 **tend** = 100;  
 **h** = 0.01;  
 }  
  
 **public void** display() {  
 Stage window = **new** Stage();  
 window.setTitle(**"Parameters"**);  
 window.initModality(Modality.***APPLICATION\_MODAL***); *//okresla czy okno ktore zostalo otwarte blokuje ekran glowny ekranu* Label label\_c = **new** Label(**"c: "**);  
 Label label\_a = **new** Label(**"A [radiany]: "**);  
 Label label\_f = **new** Label(**"f [Hz]: "**);  
 Label label\_theta = **new** Label(**"theta [stopnie]: "**);  
 Label label\_t0 = **new** Label(**"t0 [s]: "**);  
 Label label\_tend = **new** Label(**"tend [s]: "**);  
 Label label\_omega = **new** Label(**"omega [rad/s]: "**);  
 Label label\_h = **new** Label(**"h: "**);  
  
 TextField theta = **new** TextField(Double.*toString*(*theta0*));  
 theta.setEditable(**true**);  
 TextField omega = **new** TextField(Double.*toString*(**omega0**));  
 omega.setEditable(**true**);  
 TextField tstart = **new** TextField(Double.*toString*(**t0**));  
 tstart.setEditable(**true**);  
 TextField tstop = **new** TextField(Double.*toString*(**tend**));  
 tstop.setEditable(**true**);  
 TextField stepH = **new** TextField(Double.*toString*(**h**));  
 stepH.setEditable(**true**);  
  
  
 Slider sliderC = **new** Slider(0, 0.5, **c**);  
 sliderC.setShowTickLabels(**true**);  
 sliderC.setShowTickMarks(**true**);  
 sliderC.setMajorTickUnit(0.1);  
 sliderC.setPadding(**new** Insets(10));  
 sliderC.setMinorTickCount(3);  
 sliderC.setBlockIncrement(0.05);  
 sliderC.setSnapToTicks(**true**);  
  
 TextField txt\_c = **new** TextField(Double.*toString*(**c**));  
  
 sliderC.valueProperty().addListener(**new** ChangeListener<Number>() {  
 @Override  
 **public void** changed(ObservableValue<? **extends** Number> observable, Number oldValue, Number newValue) {  
  
 txt\_c.setText(Double.*toString*(newValue.doubleValue()));  
 }  
 });  
  
 Slider sliderA = **new** Slider(0, 2, **A**);  
 sliderA.setShowTickLabels(**true**);  
 sliderA.setShowTickMarks(**true**);  
 sliderA.setMajorTickUnit(0.5);  
 sliderA.setPadding(**new** Insets(10));  
 sliderA.setMinorTickCount(3);  
 sliderA.setBlockIncrement(0.1);  
 sliderA.setSnapToTicks(**true**);  
  
 TextField txt\_a = **new** TextField(Double.*toString*(**A**));  
 sliderA.valueProperty().addListener(**new** ChangeListener<Number>() {  
 @Override  
 **public void** changed(ObservableValue<? **extends** Number> observable, Number oldValue, Number newValue) {  
 txt\_a.setText(Double.*toString*(newValue.doubleValue()));  
 }  
 });  
  
 Slider sliderF = **new** Slider(0, 2, **f**);  
 sliderF.setShowTickLabels(**true**);  
 sliderF.setShowTickMarks(**true**);  
 sliderF.setMajorTickUnit(0.5);  
 sliderF.setPadding(**new** Insets(10));  
 sliderF.setMinorTickCount(3);  
 sliderF.setBlockIncrement(0.1);  
 sliderF.setSnapToTicks(**true**);  
  
 TextField txt\_f = **new** TextField(Double.*toString*(**f**));  
 sliderF.valueProperty().addListener(**new** ChangeListener<Number>() {  
 @Override  
 **public void** changed(ObservableValue<? **extends** Number> observable, Number oldValue, Number newValue) {  
 txt\_f.setText(Double.*toString*(newValue.doubleValue()));  
 }  
 });  
  
 *//* Button btnSet = **new** Button(**"Set"**);  
 btnSet.setPrefWidth(100);  
 btnSet.setOnAction(e -> {  
 **c** = Double.*parseDouble*(txt\_c.getText());  
 **A** = Double.*parseDouble*(txt\_a.getText());  
 **f** = Double.*parseDouble*(txt\_f.getText());  
 *theta0* = Double.*parseDouble*(theta.getText());  
 **omega0** = Double.*parseDouble*(omega.getText());  
 **t0** = Double.*parseDouble*(tstart.getText());  
 **tend** = Double.*parseDouble*(tstop.getText());  
 **h** = Double.*parseDouble*(stepH.getText());  
 window.close();  
 });  
 btnSet.setAlignment(Pos.***CENTER***);  
  
 HBox hboxC = **new** HBox(label\_c, sliderC, txt\_c);  
 hboxC.setAlignment(Pos.***CENTER***);  
 HBox hboxA = **new** HBox(label\_a, sliderA, txt\_a);  
 hboxA.setAlignment(Pos.***CENTER***);  
 HBox hboxF = **new** HBox(label\_f, sliderF, txt\_f);  
 hboxF.setAlignment(Pos.***CENTER***);  
 HBox hboxtheta = **new** HBox(label\_theta, theta);  
 hboxtheta.setAlignment(Pos.***CENTER***);  
 HBox hboxomega = **new** HBox(label\_omega, omega);  
 hboxomega.setAlignment(Pos.***CENTER***);  
 HBox HBt0 = **new** HBox(label\_t0, tstart);  
 HBt0.setAlignment(Pos.***CENTER***);  
 HBox HBtend = **new** HBox(label\_tend, tstop);  
 HBtend.setAlignment(Pos.***CENTER***);  
 HBox HBh = **new** HBox(label\_h, stepH);  
 HBh.setAlignment(Pos.***CENTER***);  
  
  
 VBox vbox = **new** VBox(20, hboxC, hboxA, hboxF, hboxtheta, hboxomega, HBt0, HBtend, HBh, btnSet);  
  
 Scene scene = **new** Scene(vbox, 400, 650);  
 window.setScene(scene);  
  
 window.showAndWait(); *//bo okno modalne* }  
  
 **public** WahadloAnimacja(**double** c, **double** a, **double** f) {  
 **this**.**c** = c;  
 **this**.**A** = a;  
 **this**.**f** = f;  
 }  
  
 **public double** getC() {  
 **return c**;  
 }  
  
 **public double** getA() {  
 **return A**;  
 }  
  
 **public double** getF() {  
 **return f**;  
 }  
  
 **public double** getTheta0() {  
 **return** *theta0*;  
 }  
  
 **public double** getT0() {  
 **return t0**;  
 }  
  
 **public double** getH() {  
 **return h**;  
 }  
  
 **public double** getTend() {  
 **return tend**;  
 }  
  
 **public double** getOmega0() {  
 **return omega0**;  
 }  
  
 **public** WahadloAnimacja(**double** A, **double** f, **double** c, **double** theta0, **double** t0, **double** h, **double** tend, **double** omega0) {  
 **this**.**A** = A;  
 **this**.**f** = f;  
 **this**.**c** = c;  
 **this**.*theta0* = theta0;  
 **this**.**t0** = t0;  
 **this**.**h** = h;  
 **this**.**tend** = tend;  
 **this**.**omega0** = omega0;  
 }  
  
 **public** ArrayList<XYChart.Series> getValues() {  
 XYChart.Series values = **new** XYChart.Series();  
 XYChart.Series values2 = **new** XYChart.Series();  
 XYChart.Series values3 = **new** XYChart.Series();  
  
 ArrayList<XYChart.Series> daneOut = **new** ArrayList();  
  
 *thetaList* = **new** ArrayList<>();  
 *timeList* = **new** ArrayList<>();  
  
 **double** t = **t0**;  
 **double** theta = *theta0*;  
 **double** omega = **omega0**;  
 **double** n = Math.*floor*((**tend** - **t0**) / **h**);  
  
 **for** (**int** i = 1; i < n; i++) {  
 **double** thetaOld = theta;  
  
 **double** tMid = t + **h** / 2;  
 **double** thetaMid = theta + **h** / 2 \* omega;  
 **double** omegaMid = omega - Math.*sin*(thetaOld) \* **h** / 2 - **c** \* omega \* **h** / 2 + **A** \* Math.*sin*(t \* **f**) \* **h** / 2;  
  
 theta = theta + **h** \* omegaMid;  
  
 omega = omega - Math.*sin*(thetaMid) \* **h** - **c** \* omegaMid \* **h** + **A** \* Math.*sin*(tMid \* **f**) \* **h**;  
  
 t = t + **h**;  
  
 values.getData().add(**new** XYChart.Data(t, theta));  
  
 values2.getData().add(**new** XYChart.Data<>(t, omega));  
  
 values3.getData().add(**new** XYChart.Data<>(theta, omega));  
  
 *thetaList*.add(theta);  
 *timeList*.add(t);  
 }  
 daneOut.add(values);  
 daneOut.add(values2);  
 daneOut.add(values3);  
 **return** daneOut;  
  
 }  
  
 **public void** drawAnimation(){  
  
 Stage stage=**new** Stage();  
 **double** xc=250f;  
 **double** yc=250f;  
 **double** r=150f;  
  
 Circle circle = **new** Circle();  
  
 circle.setCenterX(xc+r\*Math.*sin*(*theta0*));  
 circle.setCenterY(yc+r\*Math.*cos*(*theta0*));  
  
 circle.setRadius(25.0);  
  
 circle.setFill(Color.***ROSYBROWN***);  
  
 circle.setStrokeWidth(20);  
  
 Path path = **new** Path();  
 path.getElements().add(**new** MoveTo(xc+r\*Math.*sin*(*theta0*),yc+r\*Math.*cos*(*theta0*)));  
  
  
 **for**(**int** i=0;i<*thetaList*.size();i++){  
 **double** x=xc+r\*Math.*sin*(*thetaList*.get(i));  
 **double** y=yc+r\*Math.*cos*(*thetaList*.get(i));  
  
 path.getElements().add(**new** LineTo(x,y));  
 }  
  
 PathTransition pathTransition = **new** PathTransition();  
  
 pathTransition.setDuration(Duration.*seconds*(**tend**));  
  
 pathTransition.setNode(circle);  
  
 pathTransition.setPath(path);  
  
 pathTransition.setOrientation(PathTransition.OrientationType.***ORTHOGONAL\_TO\_TANGENT***);  
  
 pathTransition.setCycleCount(1);  
  
 pathTransition.setAutoReverse(**true**);  
  
 pathTransition.play();  
  
 Group root = **new** Group(circle);  
  
 Scene scene = **new** Scene(root,500,500);  
  
 stage.setTitle(**"Animation"**);  
  
 stage.setScene(scene);  
  
 stage.show();  
  
 }  
  
}

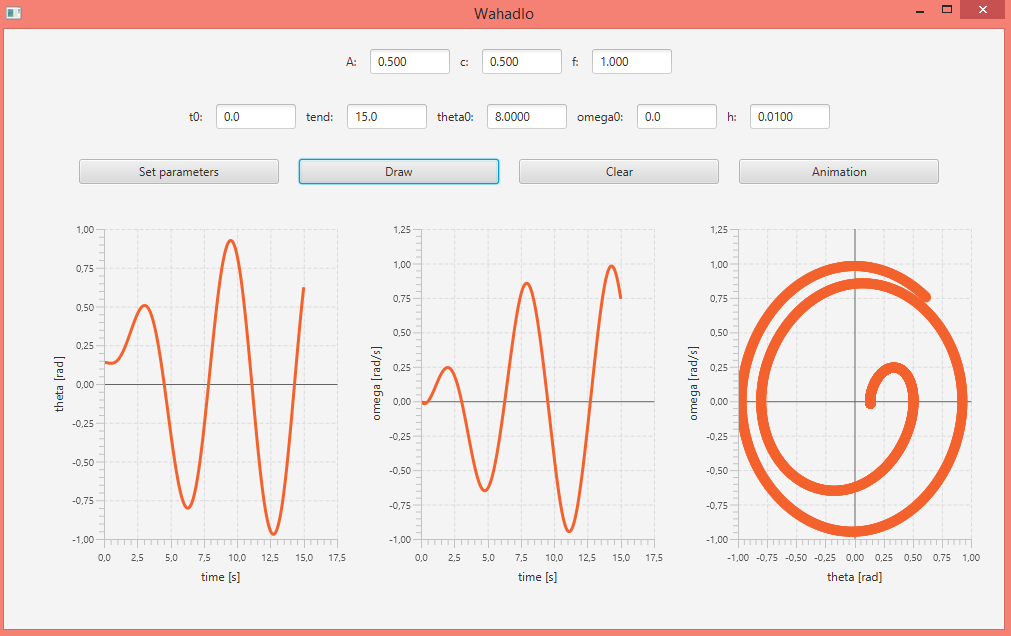
**Klasa WahadloAnimacjaFX**

**import** javafx.application.Application;  
**import** javafx.geometry.Insets;  
**import** javafx.geometry.Pos;  
**import** javafx.scene.Scene;  
**import** javafx.scene.chart.LineChart;  
**import** javafx.scene.chart.NumberAxis;  
**import** javafx.scene.chart.ScatterChart;  
**import** javafx.scene.chart.XYChart;  
**import** javafx.scene.control.Button;  
**import** javafx.scene.control.Label;  
**import** javafx.scene.control.TextField;  
**import** javafx.scene.layout.HBox;  
**import** javafx.scene.layout.VBox;  
**import** javafx.stage.Stage;  
**import** java.util.ArrayList;  
  
  
**public class** WahadloAnimacjaFX **extends** Application {  
  
  
 **private** TextField **t0Field**;  
 **private** TextField **tendField**;  
 **private** TextField **hField**;  
 **private** TextField **aField**;  
 **private** TextField **cField**;  
 **private** TextField **fField**;  
 **private** TextField **omega0Field**;  
 **private** TextField **theta0Field**;  
  
 Button **btnDraw**;  
 Button **btnClear**;  
 Button **btnAnimation**;  
 Button **btnSetParam**;  
  
  
 WahadloAnimacja **wahadlo**;  
  
 LineChart<Number, Number> **figure**;  
  
 LineChart<Number, Number> **figure2**;  
  
 ScatterChart<Number, Number> **figure3**;  
  
 @Override  
 **public void** start(Stage primaryStage) **throws** Exception {  
  
 primaryStage.setTitle(**"Wahadlo"**);  
  
  
 **wahadlo** = **new** WahadloAnimacja();  
  
 VBox layout = **new** VBox();  
 layout.setSpacing(30);  
 layout.setPadding(**new** Insets(20, 20, 30, 30));  
  
 Label tend = **new** Label(**"tend: "**);  
 Label f = **new** Label(**"f: "**);  
 Label a = **new** Label(**"A: "**);  
 Label h = **new** Label(**"h: "**);  
 Label c = **new** Label(**"c: "**);  
 Label omega0 = **new** Label(**"omega0: "**);  
 Label theta0 = **new** Label(**"theta0: "**);  
 Label t0 = **new** Label(**"t0: "**);  
  
 **tendField** = **new** TextField();  
**tendField**.setPrefWidth(80);  
 **tendField**.setEditable(**false**);  
 **aField** = **new** TextField();  
**aField**.setPrefWidth(80);  
 **aField**.setEditable(**false**);  
 **fField** = **new** TextField();  
**fField**.setPrefWidth(80);  
 **fField**.setEditable(**false**);  
 **t0Field** = **new** TextField();

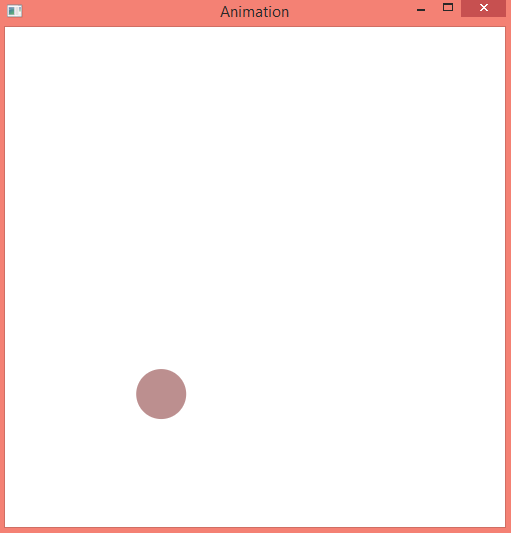
**t0Field**.setPrefWidth(80);  
 **t0Field**.setEditable(**false**);  
 **hField** = **new** TextField();  
**hField**.setPrefWidth(80);  
 **hField**.setEditable(**false**);  
 **cField** = **new** TextField();  
**cField**.setPrefWidth(80);  
 **cField**.setEditable(**false**);  
 **omega0Field** = **new** TextField();

**omega0Field**.setPrefWidth(80);  
 **omega0Field**.setEditable(**false**);  
 **theta0Field** = **new** TextField();  
**theta0Field**.setPrefWidth(80);  
 **theta0Field**.setEditable(**false**);  
  
 HBox hbox1 = **new** HBox(a, **aField**, c, **cField**, f, **fField**);  
  
 hbox1.setAlignment(Pos.***CENTER***);  
 hbox1.setSpacing(10);  
  
 layout.getChildren().add(hbox1);  
  
 HBox hbox2 = **new** HBox(t0, **t0Field**, tend, **tendField**, theta0, **theta0Field**, omega0, **omega0Field**, h, **hField**);  
  
 hbox2.setAlignment(Pos.***CENTER***);  
 hbox2.setSpacing(10);  
  
 layout.getChildren().add(hbox2);  
  
 **btnSetParam** = **new** Button(**"Set parameters"**);  
 **btnSetParam**.setPrefWidth(200);  
 **btnSetParam**.setOnAction(e -> {  
 **wahadlo**.display();  
 updateParam();  
 });  
  
  
 **btnDraw** = **new** Button(**"Draw"**);  
 **btnDraw**.setPrefWidth(200);  
 **btnDraw**.setOnAction(e -> {  
 **double** tstop = Double.*parseDouble*(**tendField**.getText());  
 **double** ff = Double.*parseDouble*(**fField**.getText());  
 **double** aa = Double.*parseDouble*(**aField**.getText());  
 **double** cc = Double.*parseDouble*(**cField**.getText());  
 **double** theta = Double.*parseDouble*(**theta0Field**.getText()) \* Math.***PI*** / 180;  
 **double** tstart = Double.*parseDouble*(**t0Field**.getText());  
 **double** hh = Double.*parseDouble*(**hField**.getText());  
 **double** omega = Double.*parseDouble*(**omega0Field**.getText());  
  
 WahadloAnimacja wahadlo = **new** WahadloAnimacja(aa, ff, cc, theta, tstart, hh, tstop, omega);  
  
 ArrayList<XYChart.Series> data = wahadlo.getValues();  
 ArrayList<ScatterChart.Series> data2 = wahadlo.getValues();  
  
 **figure**.setCreateSymbols(**false**);  
 **figure**.getData().clear();  
 **figure**.setLegendVisible(**false**);  
 **figure**.getData().add(data.get(0));  
  
 **figure2**.setCreateSymbols(**false**);  
 **figure2**.getData().clear();  
 **figure2**.setLegendVisible(**false**);  
 **figure2**.getData().add(data.get(1));  
  
 **figure3**.setLegendVisible(**false**);  
 **figure3**.getData().clear();  
 **figure3**.getData().add(data2.get(2));  
 });  
  
 **btnClear** = **new** Button(**"Clear"**);  
 **btnClear**.setPrefWidth(200);  
 **btnClear**.setOnAction(e -> {  
  
 **figure**.getData().clear();  
 **figure2**.getData().clear();  
 **figure3**.getData().clear();  
  
 });  
  
 **btnAnimation** = **new** Button(**"Animation"**);  
 **btnAnimation**.setPrefWidth(200);  
 **btnAnimation**.setOnAction(e -> {  
  
 **wahadlo**.drawAnimation();  
 });  
  
 HBox hbox3 = **new** HBox(**btnSetParam**, **btnDraw**, **btnClear**, **btnAnimation**);  
 hbox3.setSpacing(20);  
 hbox3.setAlignment(Pos.***CENTER***);  
 layout.getChildren().add(hbox3);  
  
 NumberAxis x = **new** NumberAxis();  
 x.setLabel(**"time [s]"**);  
 NumberAxis y = **new** NumberAxis();  
 y.setLabel(**"theta [rad]"**);  
  
 **figure** = **new** LineChart<Number, Number>(x, y);  
  
  
 NumberAxis x2 = **new** NumberAxis();  
 x2.setLabel(**"time [s]"**);  
 NumberAxis y2 = **new** NumberAxis();  
 y2.setLabel(**"omega [rad/s]"**);  
  
 **figure2** = **new** LineChart<Number, Number>(x2, y2);  
  
 NumberAxis x3 = **new** NumberAxis();  
 x3.setLabel(**"theta [rad]"**);  
 NumberAxis y3 = **new** NumberAxis();  
 y3.setLabel(**"omega [rad/s]"**);  
  
 **figure3** = **new** ScatterChart<Number, Number>(x3, y3);  
  
 HBox hbox5 = **new** HBox(**figure**, **figure2**, **figure3**);  
 hbox5.setAlignment(Pos.***CENTER***);  
 layout.getChildren().add(hbox5);  
  
 Scene scene = **new** Scene(layout, 1000, 600);  
  
 primaryStage.setScene(scene);  
 primaryStage.show();  
  
 }  
  
 **public static void** main(String[] args) {  
 *launch*(args);  
 }  
  
 **private void** updateParam() { **cField**.setText(String.*format*(**"%.3f"**, **wahadlo**.getC()).replaceAll(**","**, **"."**));  
 **aField**.setText(String.*format*(**"%.3f"**, **wahadlo**.getA()).replaceAll(**","**, **"."**));  
 **fField**.setText(String.*format*(**"%.3f"**, **wahadlo**.getF()).replaceAll(**","**, **"."**));  
 **theta0Field**.setText(String.*format*(**"%.4f"**, **wahadlo**.getTheta0()).replaceAll(**","**, **"."**));  
 **omega0Field**.setText(Double.*toString*(**wahadlo**.getOmega0()));  
 **t0Field**.setText(Double.*toString*(**wahadlo**.getT0()));  
 **tendField**.setText(Double.*toString*(**wahadlo**.getTend()));  
 **hField**.setText(String.*format*(**"%.4f"**, **wahadlo**.getH()).replaceAll(**","**, **"."**));  
 }  
  
  
}

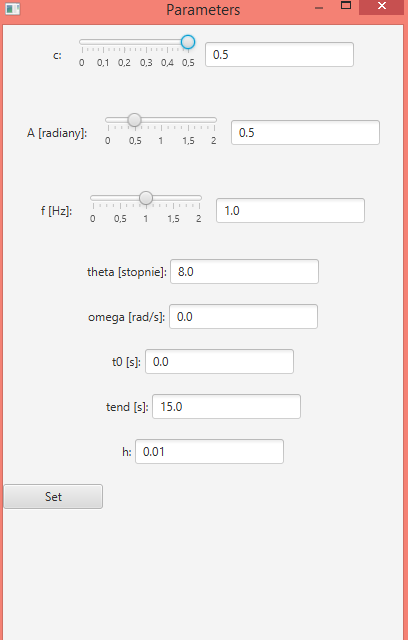
**2.2. Wyniki**



Rysunek 1. Wykresy położenia i prędkości od czasu oraz przestrzeni fazowej.



Rysunek 2. Animacja



Rysunek 3. Okno modalne, w którym można zmienić wartości wszystkich parametrów

**2.3 Wnioski**

Applet bardzo długo ładuje wartości wykresów dla czasów większych niż 20 przy zerowym stanie początkowym. Ciężko jest zmienić kolor wykresu Line Chart w Javie.