```java
package hyperDap.guiPres.application;

import java.util.Map;
import hyperDap.base.helpers.Tangenter;
import hyperDap.base.types.dataSet.ValueDataSet;
import hyperDap.generator.presInterface.PresGenerator;
import hyperDap.guiPres.fxEncapsulation.GUIMainForFX;
import hyperDap.guiPres.views.honoursMainView.HonoursMainController;
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;

/**
 * This is the core class of the {@link guiPresentation} module.
 *
 * @see javafx.application.Application
 * @see <a href=
 *
 "https://stackoverflow.com/questions/33881046/how-to-connect-fx-controller-with-main-
 app">
 *      this StackOverflow post<a/>
 * @see #start(Stage)
 *
 * @author soenk
 *
 */
public final class GUIMain extends Application implements GUIMainForFX {

  private static GUIMain instance;

  /**
   * Create a unique {@code GUIMain} instance if it does not already exist.
   *
   * @return The only allowed instance of {@code GUIMain}
   *
   * @see #GUIMain()
   */
  public synchronized static GUIMain newGUIMain() {
    if (instance == null) {
      Application.launch(GUIMain.class);
    }
    while (instance == null) {
      try {
        Thread.sleep(100);
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    }
    return instance;
  }

  /**
   * Gain access to an instance of {@code GUIMain}
   *
   * @return The only allowed instance of {@code GUIMain}
   *
   * @see #newGUIMain()
   * @see #GUIMain()
   */
  public synchronized static GUIMain getGUIMain() {
    if (instance == null) {
      return newGUIMain();
    }
    return instance;
  }

```

```java
    /**
     * Only one instance of {@code GUIMain} is allowed, which is managed by {@link #newGUIMain()} and
     * {@link #getGUIMain()}
     * <p>
     * This is required due to the forced encapsulation of JavaFX, see <a href=
     * "https://stackoverflow.com/questions/33881046/how-to-connect-fx-controller-with-main-app">this
     * StackOverflow post<a/>
     */
    public GUIMain() {
      if (instance == null) {
        instance = this;
      } else {
        throw new AssertionError(
            String.format("%s has already been instantiated! Only one allowed.", GUIMain.class));
      }
      System.out.println(String.format("%s has been instantiated.", GUIMain.class));
    }

    // *************************************************************************************************************************
    // real class begins here
    // *************************************************************************************************************************

    private HonoursMainController mainController;
    private Stage primaryStage;

    @Override
    public void init() {
      Tangenter.setPrecision(0.05);;
    }

    /**
     * {@inheritDoc}
     *
     * @see javafx.application.Application
     * @see <a href=
     * "https://stackoverflow.com/questions/33881046/how-to-connect-fx-controller-with-main-app">
     *      this StackOverflow post<a/>
     */
    @Override
    public void start(Stage primaryStage) throws Exception {
      this.primaryStage = primaryStage;
      Parent root;
      Scene scene;
      FXMLLoader loader;
      try {
        loader =
            new FXMLLoader(getClass().getResource("/hyperDap/guiPres/views/honoursMainView.fxml"));

        root = loader.load();
        this.mainController = loader.getController();
        this.mainController.giveGUIMain(this);

        System.out.println("fxml files have been loaded.");
      } catch (Exception e) {
        e.printStackTrace();

        Button button = new Button("Quit");
        root = new AnchorPane(
            new TextField(
                "An error has occurred loading UI files. \nThe application will now terminate."),
            button);

        button.setOnAction(new EventHandler<ActionEvent>() {
          @Override
          public void handle(ActionEvent ae) {
            terminate();
          }
        });
      }

      scene = new Scene(root);
      this.primaryStage.setScene(scene);
      this.primaryStage.show();
    }

    // fxEncapsulation
    // *****************************************************************************************************************************

    @Override
    public void execute(Map<String, Double> map) {
      System.out.println("Generating Data");
      for (String didi : map.keySet()) {
        System.out.println(String.format("%s: %s", didi, map.get(didi)));
      }
      System.out.println("Generation in progress...");
      ValueDataSet<? extends Number> set = PresGenerator.generate(map);
      System.out.println("Generation complete.");
      this.mainController.displayDataSet(set);
    }

    /**
     * Terminate the Application.
     * <p>
     * Will call {@link Stage#close()} and ask the Business Logic to release all resources.
     */
    @Override
    public void terminate() {
      System.out.println("Terminating Application");
      this.primaryStage.close();
    }

    // main ********************** main *********************** main *********************** main

    /**
     * Main for running and testing.
     *
     * @param args
     */
    public static void main(String[] args) {
      newGUIMain();
    }

  }
```