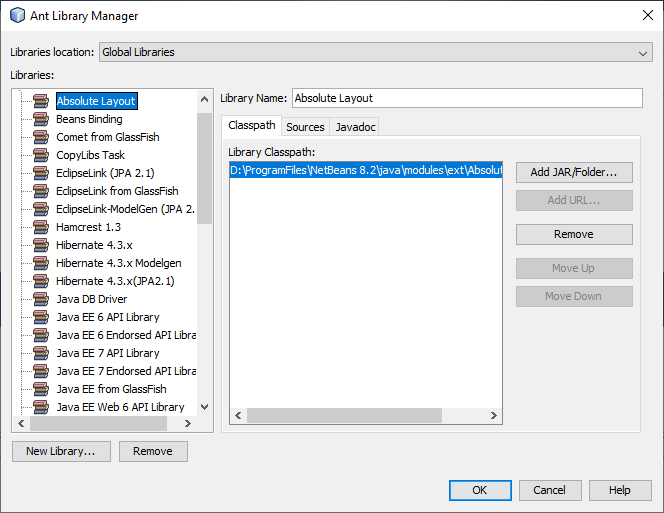
Για την 3η ΓΕ θα η ευκολότερη προσέγγιση για τη συμπερίληψη των βιβλιοθηκών που απαιτούνται στο project είναι να κάνετε ένα library με ένα δεδομένο όνομα. Εφόσον και οι καθηγητές σας έχουν το ίδιο library στον υπολογιστή τους θα τρέξει το project επιτυχώς χωρίς να χρειαστεί να φορτωθούν εξωτερικές βιβλιοθήκες.

Τρόπος δημιουργίας του library και συμπερίληψη σε αυτό των κατάλληλων βιβλιοθηκών.

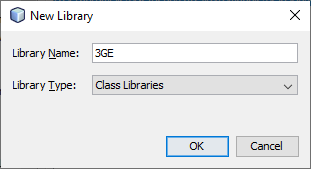
Οι βιβλιοθήκες (jar files) που θα χρειαστούμε για τη 3η ΓΕ είναι οι ακόλουθες:

1. <https://repo1.maven.org/maven2/org/jfree/jfreechart/1.0.19/jfreechart-1.0.19.jar>
2. <https://repo1.maven.org/maven2/org/jfree/jcommon/1.0.23/jcommon-1.0.23.jar>
3. <https://repo1.maven.org/maven2/com/squareup/okhttp3/okhttp/3.12.1/okhttp-3.12.1.jar>
4. <https://repo1.maven.org/maven2/com/squareup/okio/okio/2.4.3/okio-2.4.3.jar>
5. <https://repo1.maven.org/maven2/org/jetbrains/kotlin/kotlin-stdlib/1.3.61/kotlin-stdlib-1.3.61.jar>
6. <https://repo1.maven.org/maven2/com/google/code/gson/gson/2.8.5/gson-2.8.5.jar>

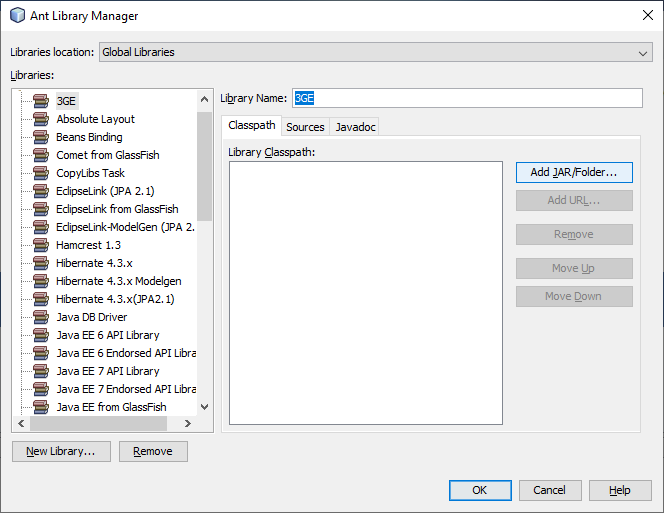
Αφού κατεβάσετε τα πιο πάνω αρχεία και τα τοποθετήσετε όλα σε ένα φάκελο, ξεκινήστε το NetBeans και πάτε στο μενού Tools 🡪 Libraries.



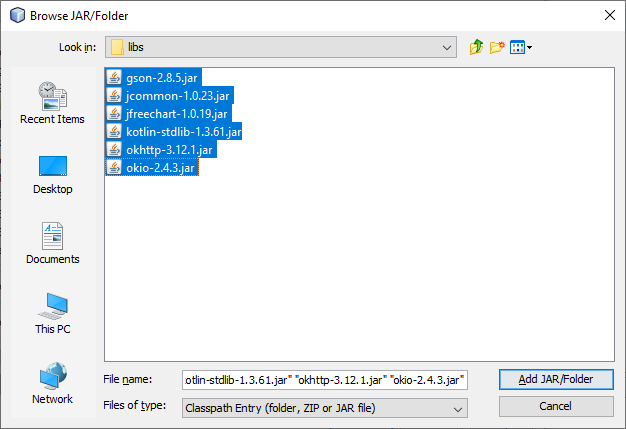
Από εκεί κάνετε κλικ κάτω αριστερά στο κουμπί “New Library” και δώστε στη βιβλιοθήκη το όνομα **3GE** με προσοχή ώστε να έχουμε όλοι το ίδιο όνομα για να μην αναζητηθεί η βιβλιοθήκη όταν ανοίξει το project σας το μέλος ΣΕΠ στον υπολογιστή του και αυτήν δεν μπορεί να βρεθεί. Δίνετε λοιπόν το όνομα αυτό όπως φαίνεται στην εικόνα.



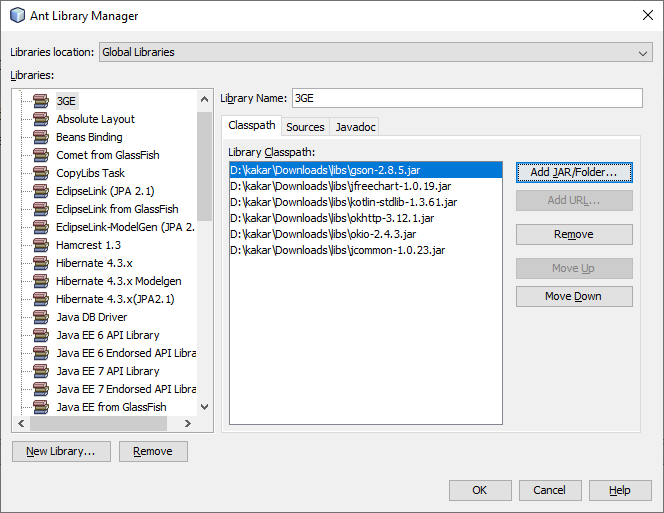
Αφού πατήσετε το ΟΚ στη συνέχεια θα εμφανισθεί το ξανά το πλαίσιο διαλόγου “Ant Library manager” για να προσθέσετε στη νέα σας βιβλιοθήκη τα jar αρχεία που κατεβάσατε. Όπως φαίνεται στην εικόνα η βιβλιοθήκη 3GE θα είναι επιλεγμένη αλλά κενή και θα μπορέσουμε τώρα πατώντας το κουμπί “Add Jar/Folder” να προσθέσουμε τα αρχεία που κατεβάσαμε.



Κάνετε λοιπόν κλικ στο κουμπί «Add Jar/Folder…» εμφανίζεται ένα πλαίσιο προσθήκης αρχείων. Σε αυτό μεταβαίνουμε στον φάκελο όπου κατεβάσαμε όλα τα Jar αρχεία και επιλέγουμε με shift στο πρώτο και Shift+click στο τελευταίο όλα τα JAR αρχεία και πατάμε το πλήκτρο «Add Jar/Folder», όπως φαίνεται στην εικόνα.



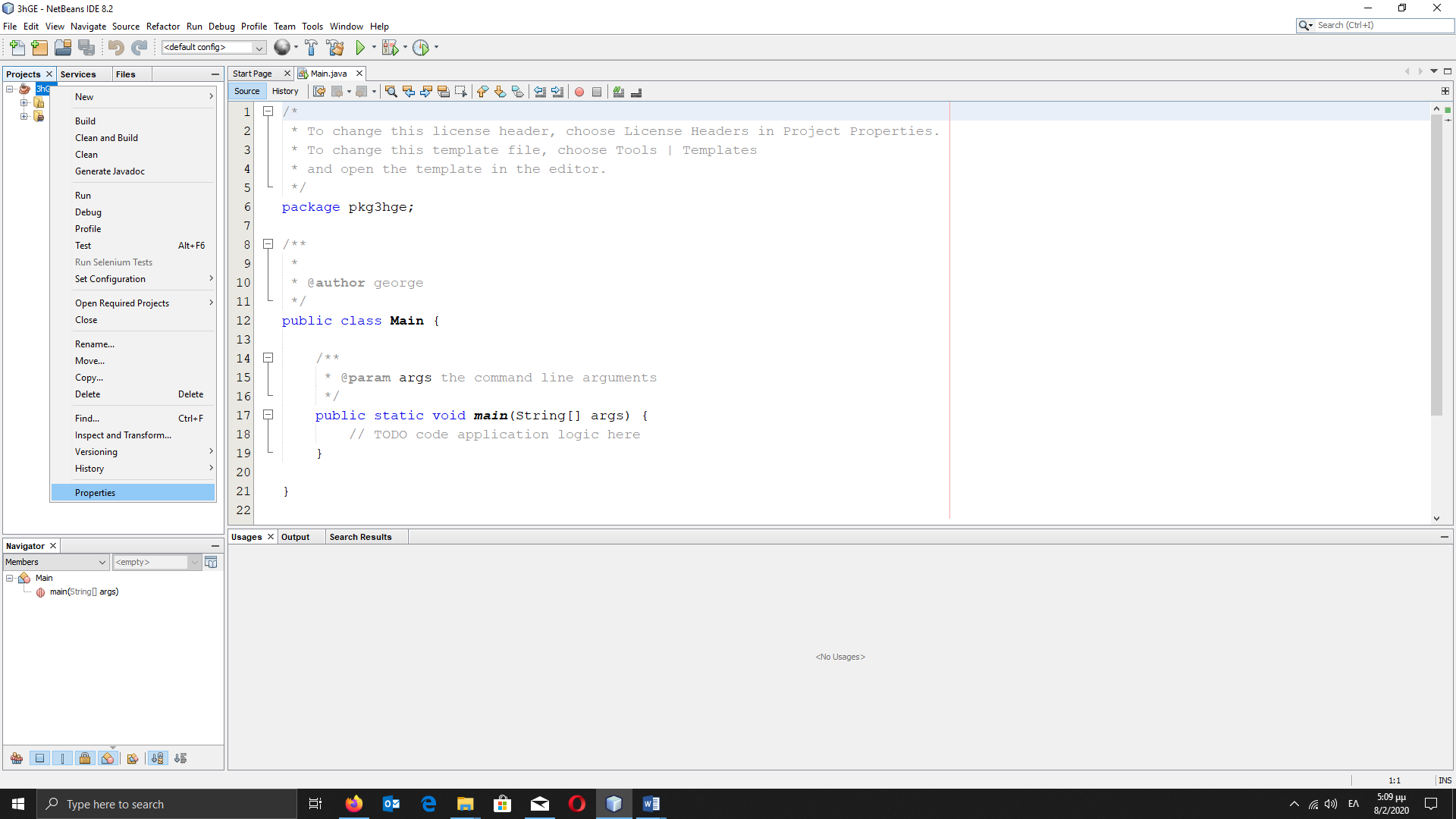
Αφού το κάνουμε αυτό θα εμφανισθεί ξανά ο library manager με τα αρχεία μας πλέον στην βιβλιοθήκη, όπως φαίνεται στην εικόνα.



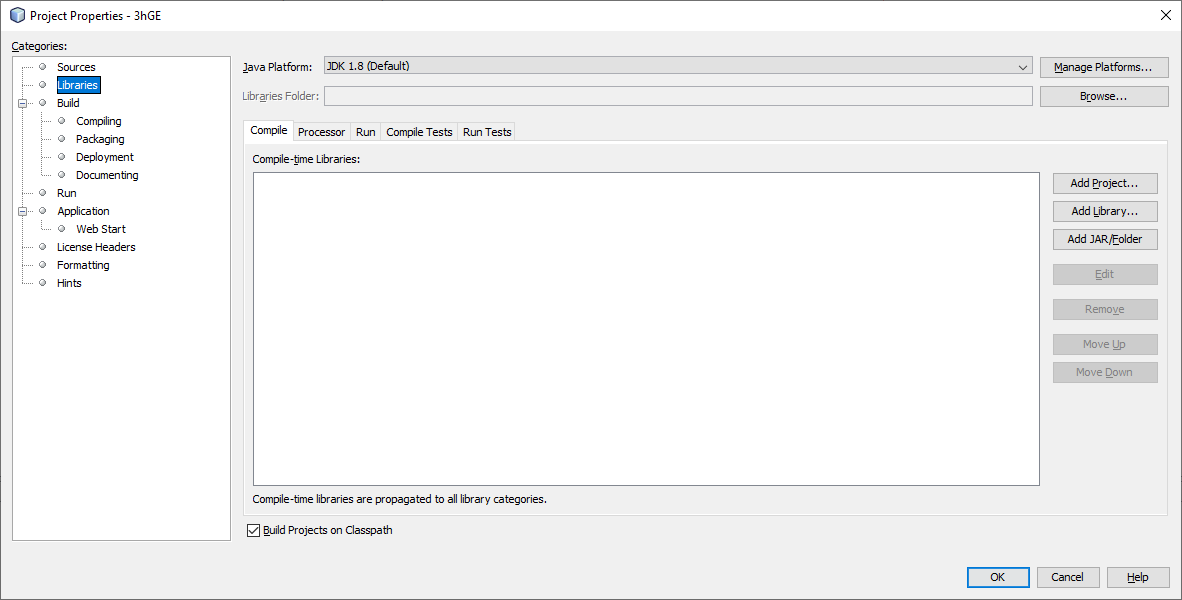
Πατάμε ΟΚ, για να κλείσουμε τον Library Manager.

Στο project μας τώρα αφού το ανοίξουμε, κάνουμε τα εξής.

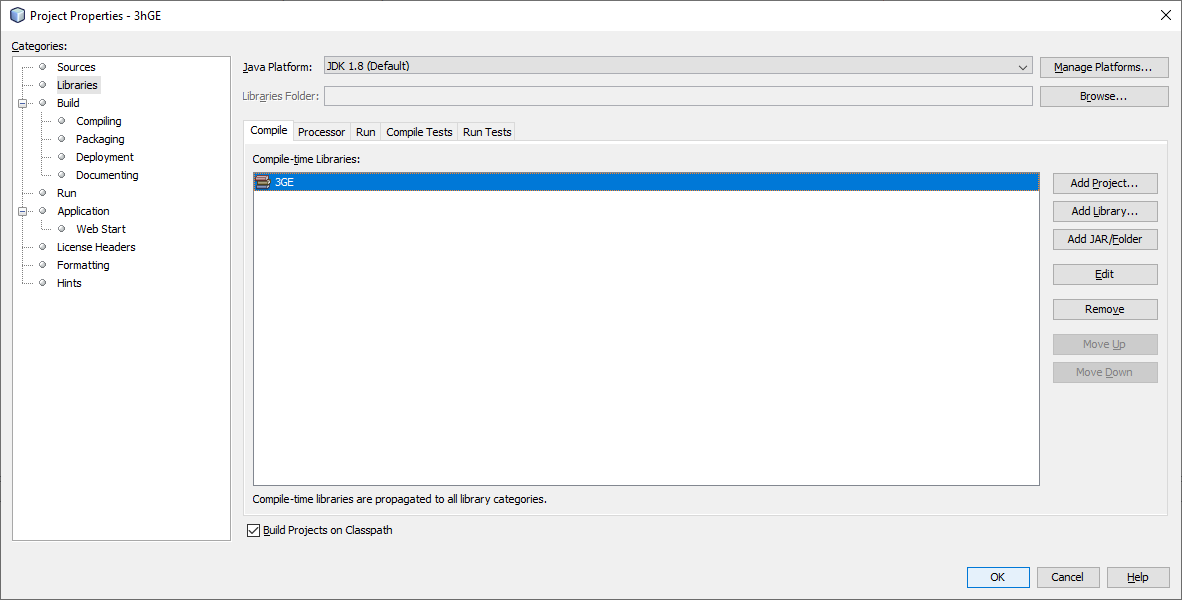
Αρχικά κάνουμε δεξί κλικ στο project και επιλέγουμε «Properties» από το αναδυόμενο μενού όπως φαίνεται στην εικόνα.



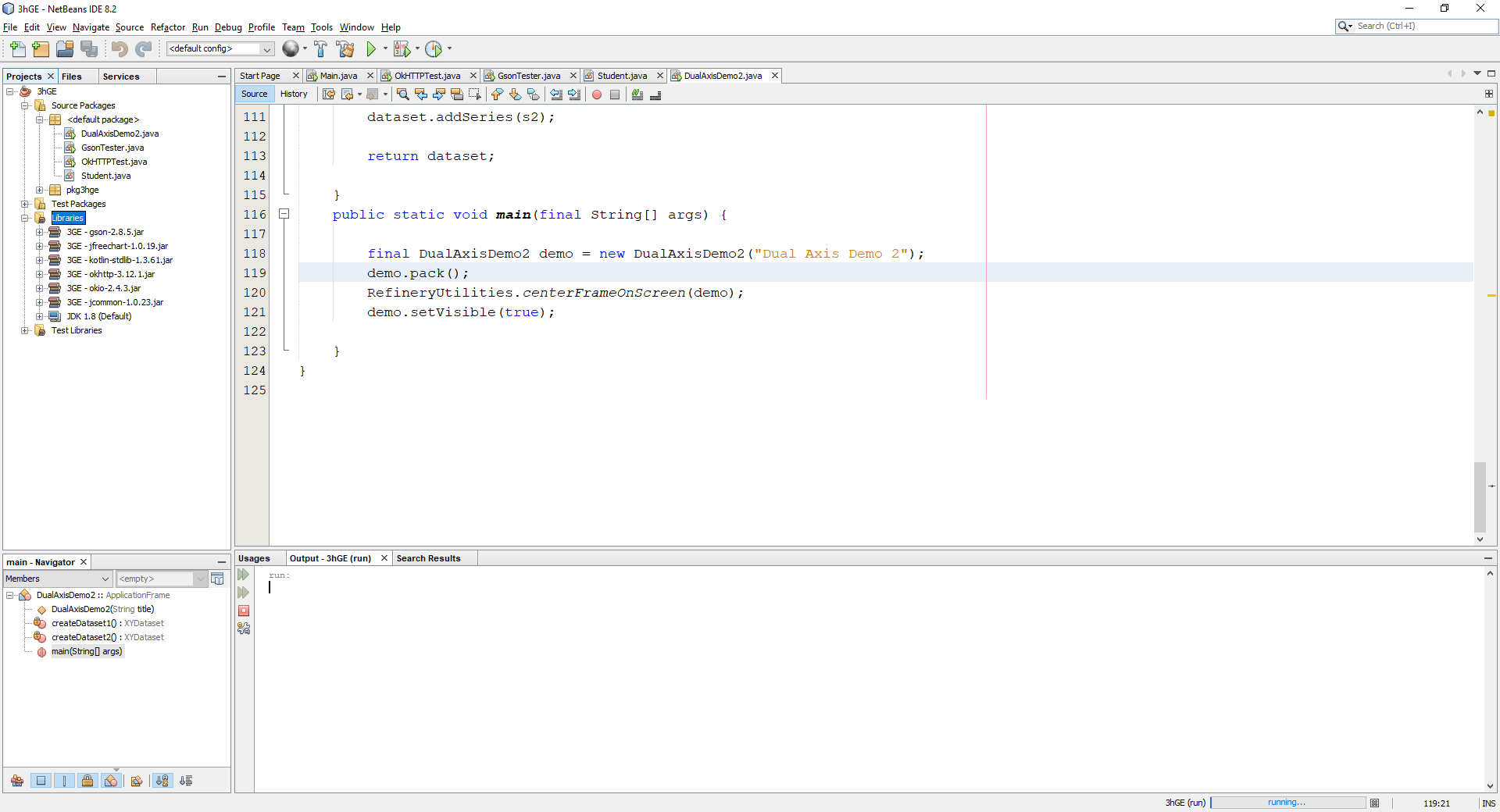
Στη συνέχεια από το πλαίσιο διαλόγου που θα εμφανισθεί επιλέγουμε «Libraries» και πατάμε το πλήκτρο «Add Library» στην δεξιά πλευρά του παραθύρου, όπως φαίνεται στην εικόνα.



Τέλος από τις διαθέσιμες βιβλιοθήκες επιλέγουμε την 3GE, η οποία πλέον εισάγεται στο project μας και πατάμε ΟΚ.



Για να επιβεβαιώσουμε πως εισάχθηκε η βιβλιοθήκη στο project μας, αρκεί να κάνουμε κλικ στον κόμβο libraries κάτω από το project ώστε να εμφανισθούν τα jar αρχεία που κατεβάσαμε και εισάγαμε με την βιβλιοθήκη.



Για να δοκιμάσετε τις βιβλιοθήκες που εισαγάγαμε μπορείτε να τρέξετε τρία δοκιμαστικά προγράμματα που τις χρησιμοποιούν και να βεβαιωθείτε πως εκτελούνται επιτυχώς:

# OkHttp

import java.io.IOException;

import okhttp3.OkHttpClient;

import okhttp3.Request;

import okhttp3.Response;

public class OkHTTPTest {

public static void main(String[] args) {

//Βάλετε το κλειδί σας εδώ. Αυτό δεν είναι έγκυρο

String key = "V-LthCRakZqR89s3zhi2";

String urlToCall

= "https://www.quandl.com/api/v3/datasets/WWDI/GRC\_NY\_GDP\_MKTP\_CN.json?api\_key="+key;

OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder().url(urlToCall).build();

try (Response response = client.newCall(request).execute()) {

if (response.isSuccessful() && response.body() != null) {

String responseString = response.body().string();

System.out.println(responseString);

}

} catch (IOException e) {

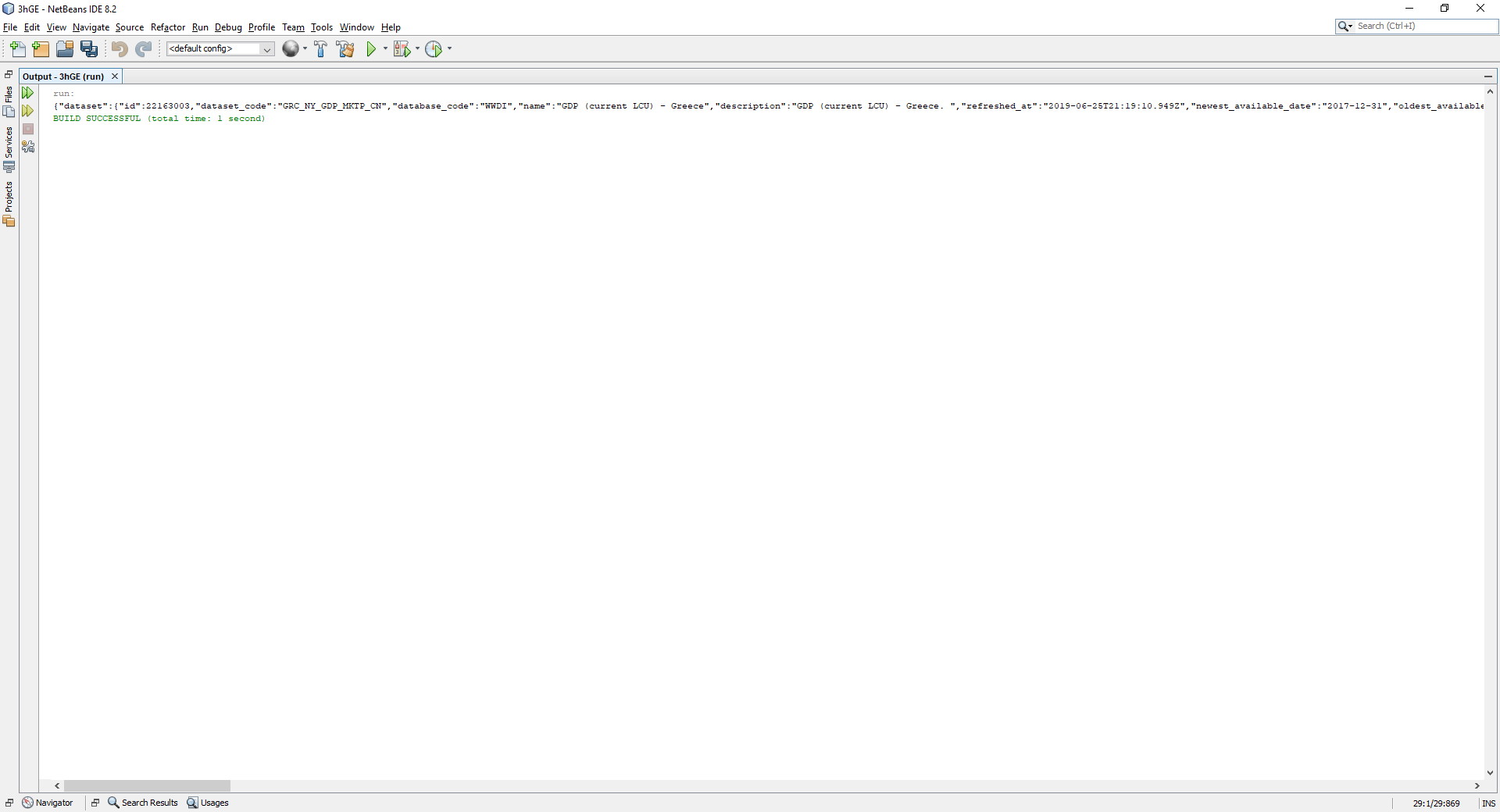
e.printStackTrace();

}

}

}

Θα πρέπει να δείτε το ακόλουθο output:



# GSon

Θα χρειαστούμε δύο αρχείο το GsonTester και την βοηθητική κλάση Student.

import com.google.gson.Gson;

import com.google.gson.GsonBuilder;

public class GsonTester {

public static void main(String[] args) {

String jsonString = "{\"name\":\"Mahesh\", \"age\":21}";

GsonBuilder builder = new GsonBuilder();

builder.setPrettyPrinting();

Gson gson = builder.create();

Student student = gson.fromJson(jsonString, Student.class);

System.out.println(student);

jsonString = gson.toJson(student);

System.out.println(jsonString);

}

}

class Student {

private String name;

private int age;

public Student(){}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

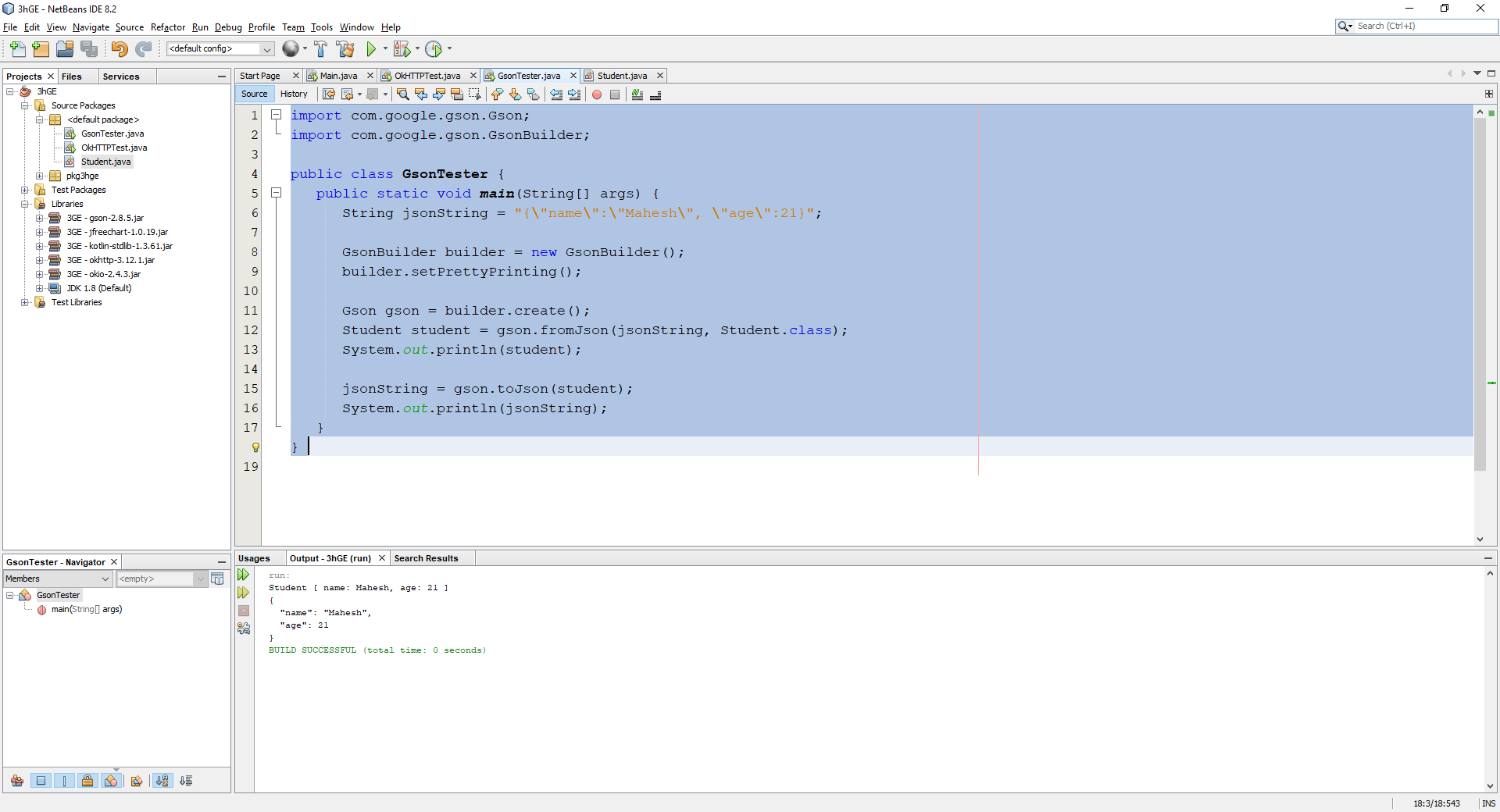
public String toString() {

return "Student [ name: "+name+", age: "+ age+ " ]";

}

}

Αφού τα κάνετε και τρέξετε το GsonTester θα πρέπει να δείτε το ακόλουθο output.



# JFreeChart

import java.awt.Color;

import java.text.SimpleDateFormat;

import org.jfree.chart.ChartFactory;

import org.jfree.chart.ChartPanel;

import org.jfree.chart.JFreeChart;

import org.jfree.chart.axis.DateAxis;

import org.jfree.chart.axis.NumberAxis;

import org.jfree.chart.labels.StandardXYToolTipGenerator;

import org.jfree.chart.plot.XYPlot;

import org.jfree.chart.renderer.xy.StandardXYItemRenderer;

import org.jfree.chart.renderer.xy.XYItemRenderer;

import org.jfree.data.time.Month;

import org.jfree.data.time.TimeSeries;

import org.jfree.data.time.TimeSeriesCollection;

import org.jfree.data.xy.XYDataset;

import org.jfree.ui.ApplicationFrame;

import org.jfree.ui.RefineryUtilities;

public class DualAxisDemo2 extends ApplicationFrame {

public DualAxisDemo2(final String title) {

super(title);

final String chartTitle = "Dual Axis Demo 2";

final XYDataset dataset = createDataset1();

final JFreeChart chart = ChartFactory.createTimeSeriesChart(

chartTitle,

"Date",

"Price Per Unit",

dataset,

true,

true,

false

);

final XYPlot plot = chart.getXYPlot();

final NumberAxis axis2 = new NumberAxis("Secondary");

axis2.setAutoRangeIncludesZero(false);

plot.setRangeAxis(1, axis2);

plot.setDataset(1, createDataset2());

plot.mapDatasetToRangeAxis(1, 1);

final XYItemRenderer renderer = plot.getRenderer();

renderer.setToolTipGenerator(StandardXYToolTipGenerator.getTimeSeriesInstance());

if (renderer instanceof StandardXYItemRenderer) {

final StandardXYItemRenderer rr = (StandardXYItemRenderer) renderer;

rr.setShapesFilled(true);

}

final StandardXYItemRenderer renderer2 = new StandardXYItemRenderer();

renderer2.setSeriesPaint(0, Color.black);

renderer.setToolTipGenerator(StandardXYToolTipGenerator.getTimeSeriesInstance());

plot.setRenderer(1, renderer2);

final DateAxis axis = (DateAxis) plot.getDomainAxis();

axis.setDateFormatOverride(new SimpleDateFormat("MMM-yyyy"));

final ChartPanel chartPanel = new ChartPanel(chart);

chartPanel.setPreferredSize(new java.awt.Dimension(500, 270));

setContentPane(chartPanel);

}

private XYDataset createDataset1() {

final TimeSeries s1 = new TimeSeries("Random Data 1", Month.class);

s1.add(new Month(2, 2001), 181.8);

s1.add(new Month(3, 2001), 167.3);

s1.add(new Month(4, 2001), 153.8);

s1.add(new Month(5, 2001), 167.6);

s1.add(new Month(6, 2001), 158.8);

s1.add(new Month(7, 2001), 148.3);

s1.add(new Month(8, 2001), 153.9);

s1.add(new Month(9, 2001), 142.7);

s1.add(new Month(10, 2001), 123.2);

s1.add(new Month(11, 2001), 131.8);

s1.add(new Month(12, 2001), 139.6);

s1.add(new Month(1, 2002), 142.9);

s1.add(new Month(2, 2002), 138.7);

s1.add(new Month(3, 2002), 137.3);

s1.add(new Month(4, 2002), 143.9);

s1.add(new Month(5, 2002), 139.8);

s1.add(new Month(6, 2002), 137.0);

s1.add(new Month(7, 2002), 132.8);

final TimeSeriesCollection dataset = new TimeSeriesCollection();

dataset.addSeries(s1);

return dataset;

}

private XYDataset createDataset2() {

final TimeSeries s2 = new TimeSeries("Random Data 2", Month.class);

s2.add(new Month(2, 2001), 429.6);

s2.add(new Month(3, 2001), 323.2);

s2.add(new Month(4, 2001), 417.2);

s2.add(new Month(5, 2001), 624.1);

s2.add(new Month(6, 2001), 422.6);

s2.add(new Month(7, 2001), 619.2);

s2.add(new Month(8, 2001), 416.5);

s2.add(new Month(9, 2001), 512.7);

s2.add(new Month(10, 2001), 501.5);

s2.add(new Month(11, 2001), 306.1);

s2.add(new Month(12, 2001), 410.3);

s2.add(new Month(1, 2002), 511.7);

s2.add(new Month(2, 2002), 611.0);

s2.add(new Month(3, 2002), 709.6);

s2.add(new Month(4, 2002), 613.2);

s2.add(new Month(5, 2002), 711.6);

s2.add(new Month(6, 2002), 708.8);

s2.add(new Month(7, 2002), 501.6);

final TimeSeriesCollection dataset = new TimeSeriesCollection();

dataset.addSeries(s2);

return dataset;

}

public static void main(final String[] args) {

final DualAxisDemo2 demo = new DualAxisDemo2("Dual Axis Demo 2");

demo.pack();

RefineryUtilities.centerFrameOnScreen(demo);

demo.setVisible(true);

}

}

To output από την εκτέλεση αυτού του δοκιμαστικού προγράμματος φαίνεται στην εικόνα που ακολουθεί:

