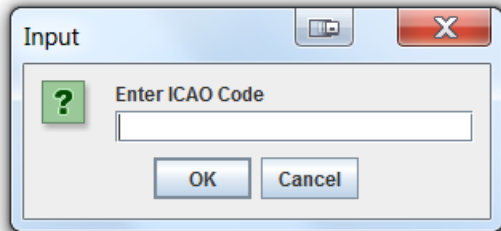


NOTE: Read the README.md file first

When you run the application a pop up window appears asking for the ICAO Code.

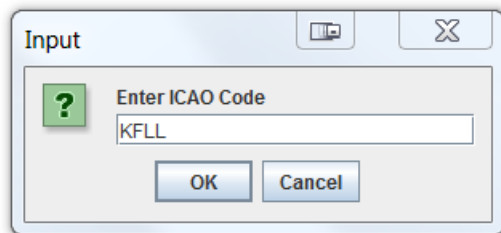
Below is a screenshot of the input window:



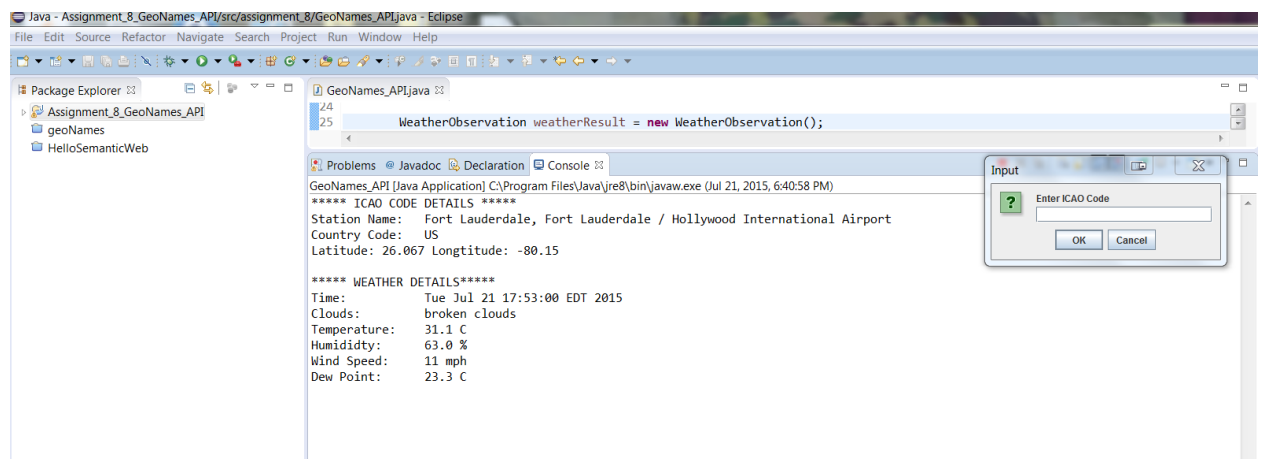
For this assignment I ran the ICAO codes for 5 US cities and 4 international cities. This was done to obtain the ICAO site information and the most recent weather available. The input and output screenshots are below.

1. Fort Lauderdale, US

INPUT:

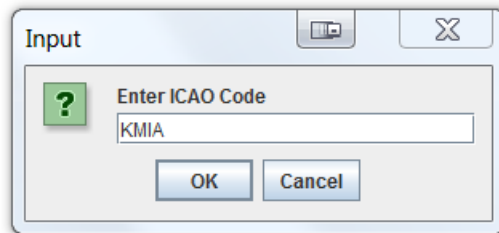


OUTPUT:



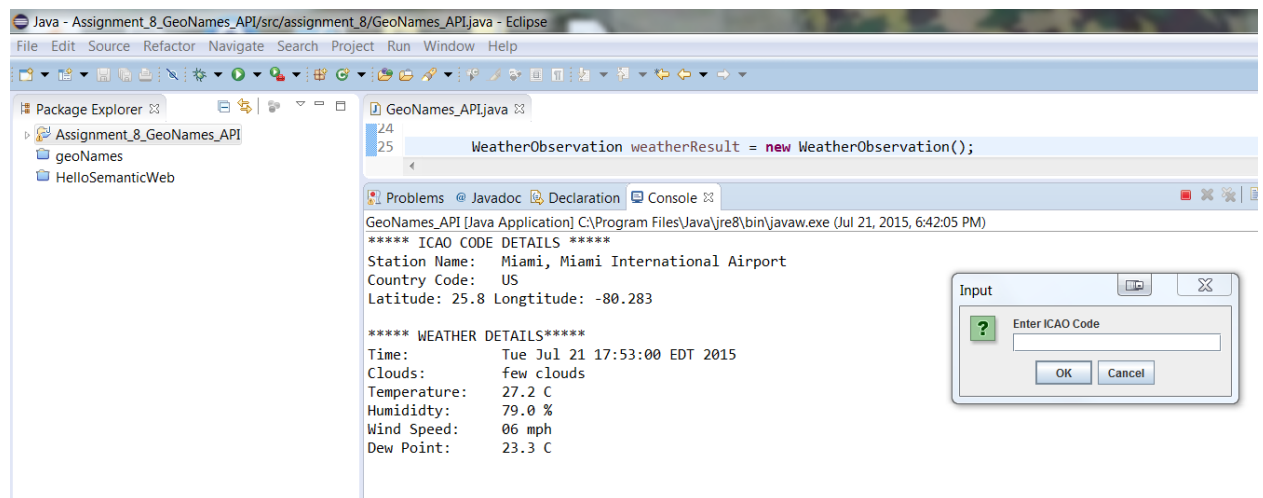
2. Miami, US

INPUT:



A Java Swing dialog box titled "Input". It contains a green square icon with a white question mark. To the right of the icon is the text "Enter ICAO Code". Below this text is a text input field containing the text "KMIA". At the bottom of the dialog are two buttons: "OK" and "Cancel".

OUTPUT:



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows the project structure: "Assignment_8_GeoNames_API" containing "geoNames" and "HelloSemanticWeb". The main editor shows the file "GeoNames_API.java" with the following code:

```
24  
25 WeatherObservation weatherResult = new WeatherObservation();
```

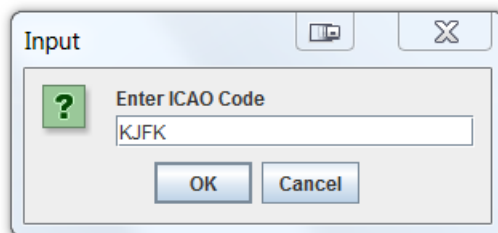
The Console window at the bottom displays the output of the application:

```
GeoNames_API [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (Jul 21, 2015, 6:42:05 PM)  
***** ICAO CODE DETAILS *****  
Station Name: Miami, Miami International Airport  
Country Code: US  
Latitude: 25.8 Longitude: -80.283  
  
***** WEATHER DETAILS*****  
Time: Tue Jul 21 17:53:00 EDT 2015  
Clouds: few clouds  
Temperature: 27.2 C  
Humidity: 79.0 %  
Wind Speed: 06 mph  
Dew Point: 23.3 C
```

Overlaid on the bottom right of the screenshot is a smaller version of the "Input" dialog box, identical to the one shown in the "INPUT:" section, but with the text input field empty.

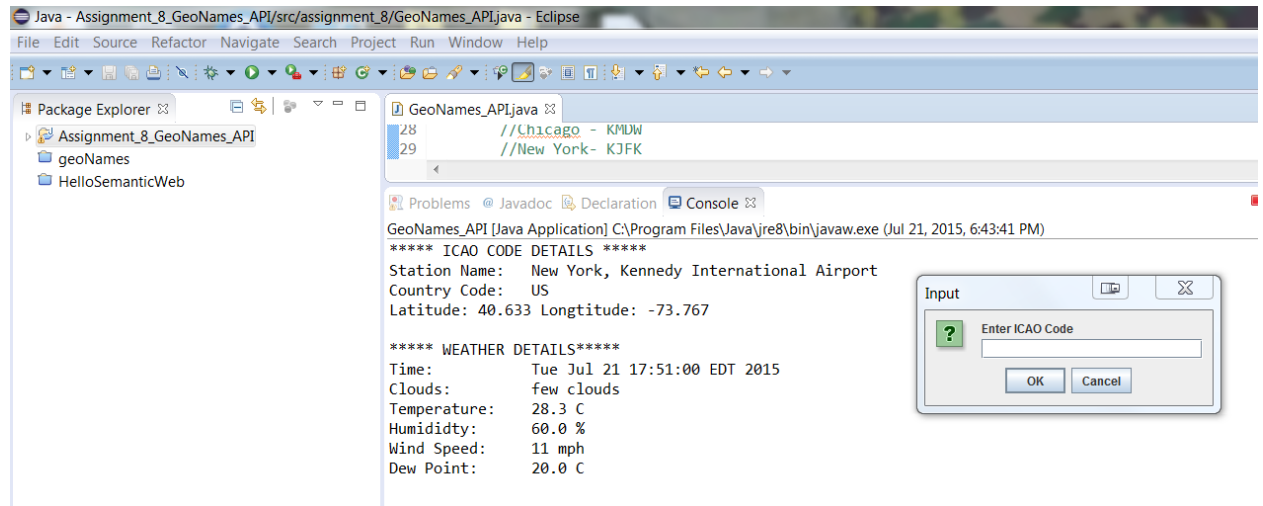
3. New York, US

INPUT:



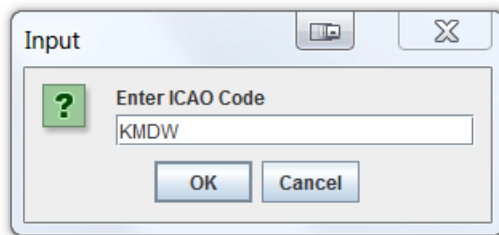
A Java Swing dialog box titled "Input". It contains a green square icon with a white question mark. To the right of the icon is the text "Enter ICAO Code". Below this text is a text input field containing the text "KJFK". At the bottom of the dialog are two buttons: "OK" and "Cancel".

OUTPUT:

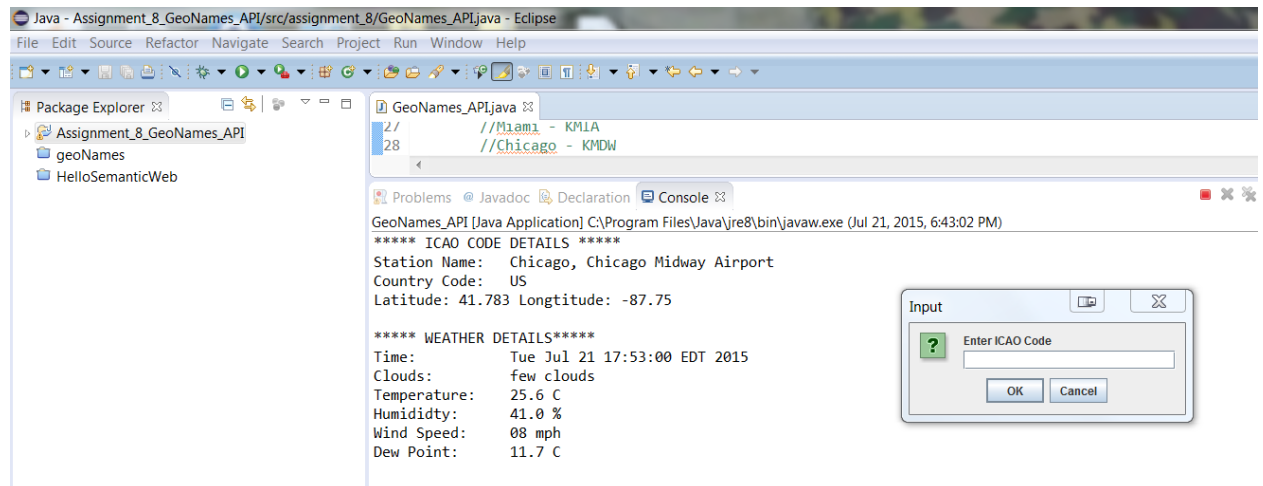


4. Chicago, US

INPUT:

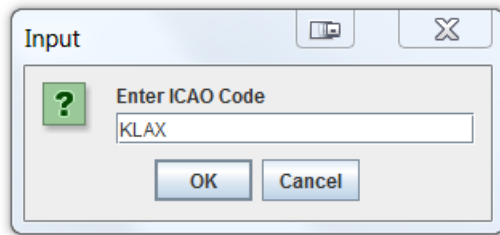


OUTPUT:



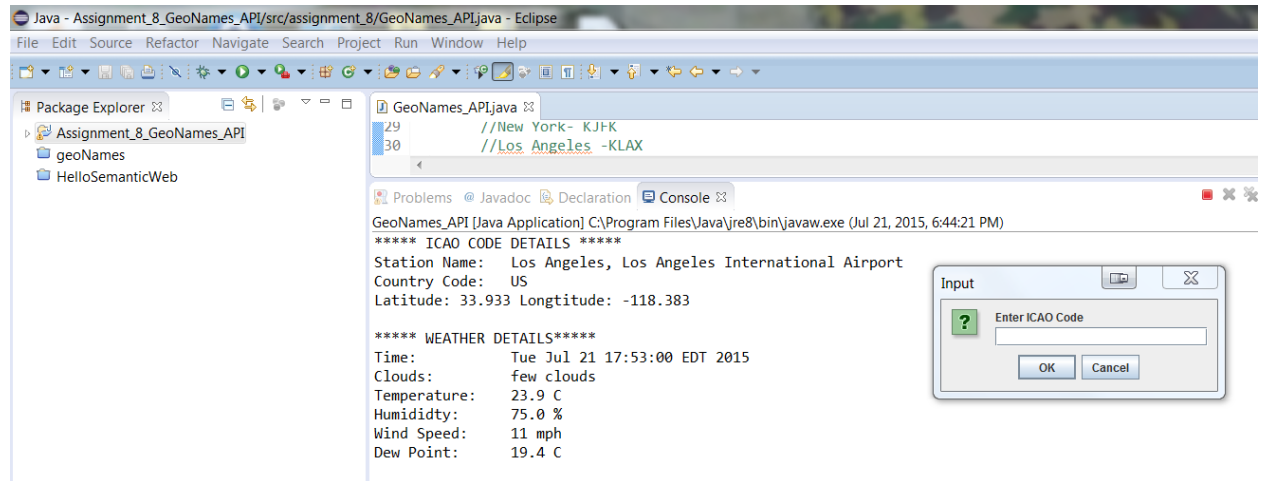
5. Los Angeles, US

INPUT:



A Java Swing dialog box titled "Input" with a green question mark icon. It contains a text field labeled "Enter ICAO Code" with the text "KLAX" entered. Below the text field are "OK" and "Cancel" buttons.

OUTPUT:



The screenshot shows the Eclipse IDE with the "GeoNames_API.java" file open. The code contains two comments: `//New York- KJ+K` and `//Los Angeles -KLAX`. The console output shows the results of the API call for Los Angeles:

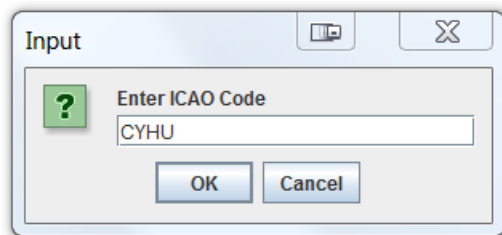
```
GeoNames_API [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (Jul 21, 2015, 6:44:21 PM)
***** ICAO CODE DETAILS *****
Station Name:  Los Angeles, Los Angeles International Airport
Country Code:  US
Latitude: 33.933 Longitude: -118.383

***** WEATHER DETAILS*****
Time:          Tue Jul 21 17:53:00 EDT 2015
Clouds:        few clouds
Temperature:   23.9 C
Humidity:      75.0 %
Wind Speed:    11 mph
Dew Point:     19.4 C
```

An inset image shows the "Input" dialog box with the "Enter ICAO Code" field empty, representing the state before the user enters "KLAX".

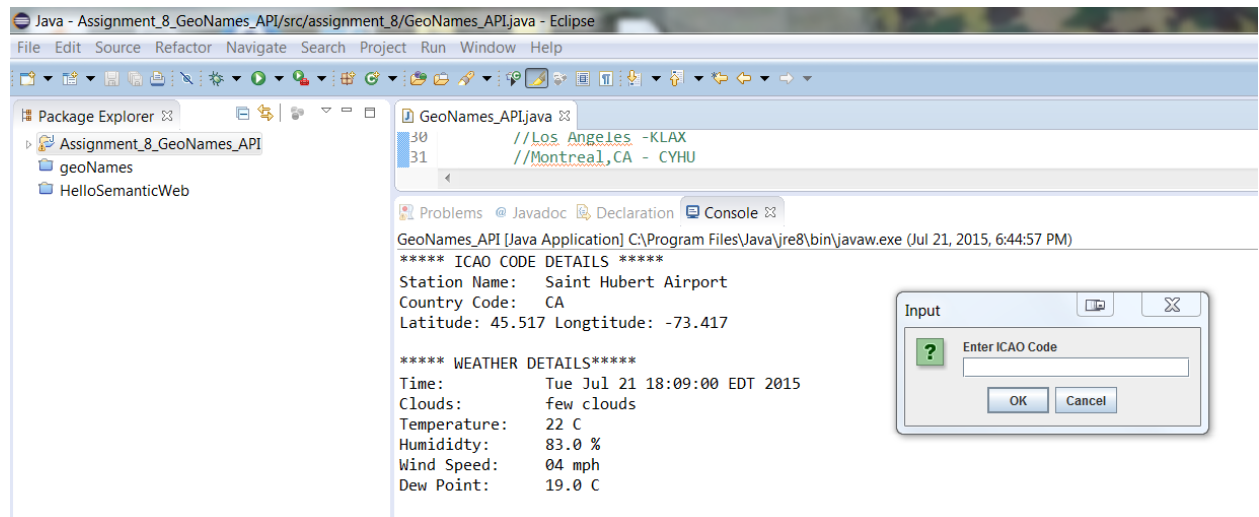
6. Montreal, Canada

INPUT:



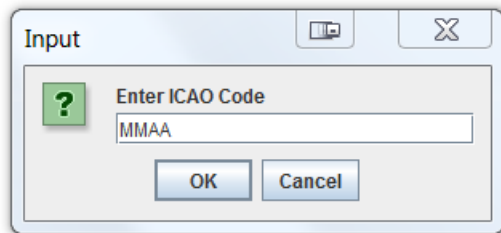
A Java Swing dialog box titled "Input" with a green question mark icon. It contains a text field labeled "Enter ICAO Code" with the text "CYHU" entered. Below the text field are "OK" and "Cancel" buttons.

OUTPUT:

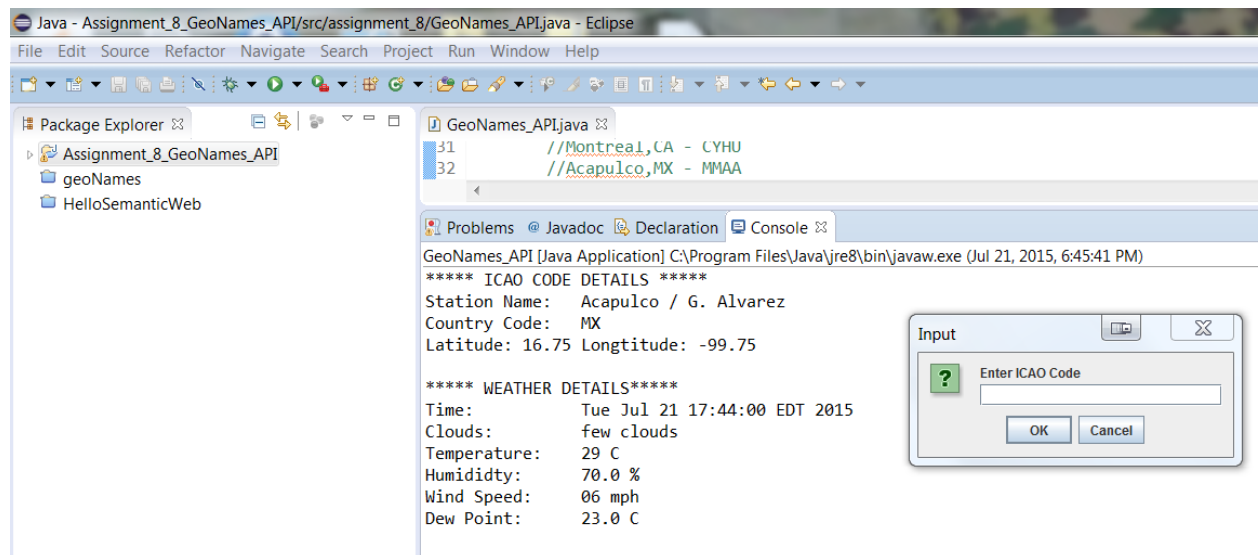


7. Acapulco, Mexico

INPUT:

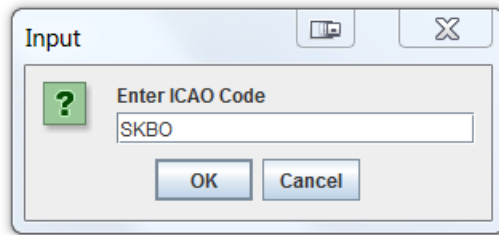


OUTPUT:



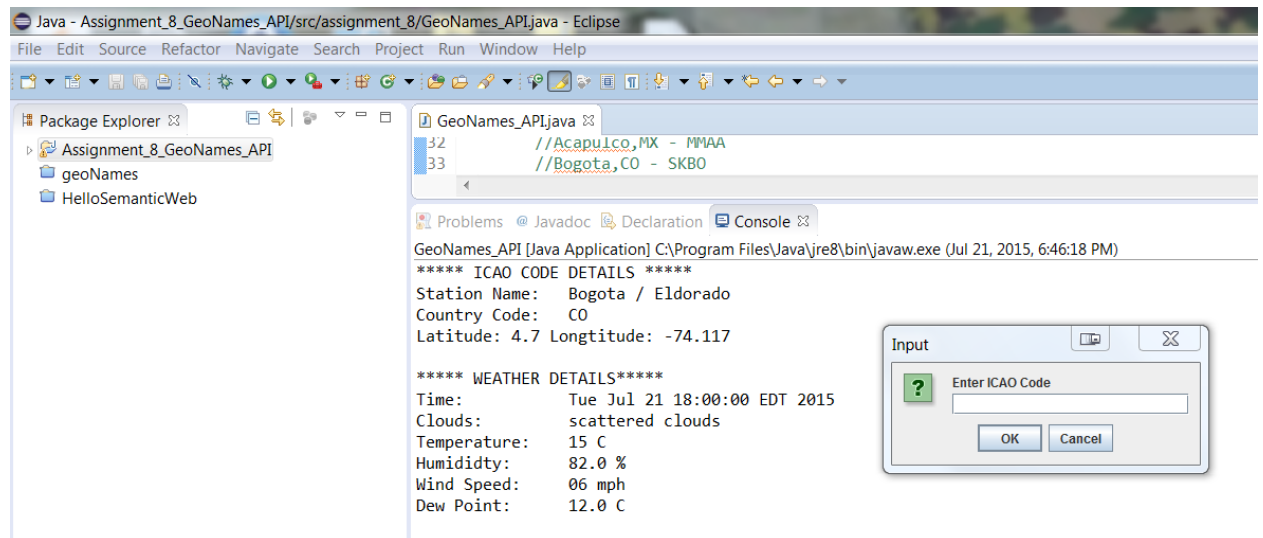
8. Bogota, Colombia

INPUT:



A dialog box titled "Input" with a green question mark icon. It contains a text field labeled "Enter ICAO Code" with the text "SKBO" entered. Below the text field are two buttons: "OK" and "Cancel".

OUTPUT:



The screenshot shows the Eclipse IDE with the file `GeoNames_API.java` open. The code contains two comments: `//Acapulco,MX - MMAA` and `//Bogota,CO - SKBO`. The console output shows the results for Bogota, Colombia:

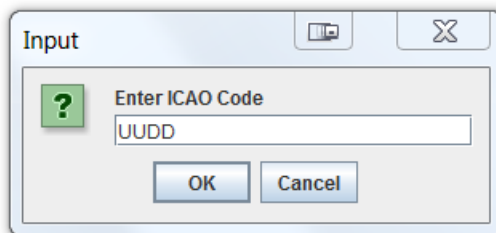
```
GeoNames_API [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (Jul 21, 2015, 6:46:18 PM)
**** ICAO CODE DETAILS ****
Station Name: Bogota / Eldorado
Country Code: CO
Latitude: 4.7 Longitude: -74.117

**** WEATHER DETAILS****
Time: Tue Jul 21 18:00:00 EDT 2015
Clouds: scattered clouds
Temperature: 15 C
Humidity: 82.0 %
Wind Speed: 06 mph
Dew Point: 12.0 C
```

Below the console output, there is a smaller version of the "Input" dialog box with the text "Enter ICAO Code" and "OK" and "Cancel" buttons.

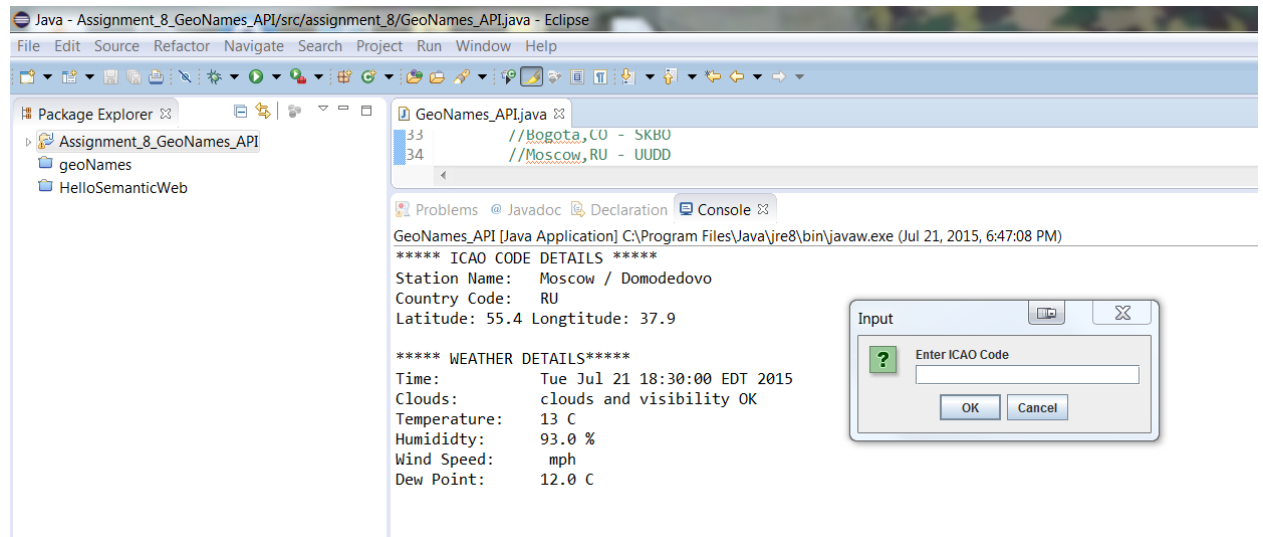
9. Moscow, Russia

INPUT:



A dialog box titled "Input" with a green question mark icon. It contains a text field labeled "Enter ICAO Code" with the text "UUDD" entered. Below the text field are two buttons: "OK" and "Cancel".

OUTPUT:



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure: Assignment_8_GeoNames_API, geoNames, and HelloSemanticWeb. The main editor shows the file GeoNames_API.java with two lines of code: `//Bogota,CO - SKBO` and `//Moscow,RU - UDD`. The Console window at the bottom shows the output of the Java application, which includes ICAO code details and weather details for Moscow. An 'Input' dialog box is also visible, prompting the user to 'Enter ICAO Code' with 'OK' and 'Cancel' buttons.

```
GeoNames_API [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (Jul 21, 2015, 6:47:08 PM)
***** ICAO CODE DETAILS *****
Station Name:  Moscow / Domodedovo
Country Code:  RU
Latitude: 55.4 Longitude: 37.9

***** WEATHER DETAILS*****
Time:           Tue Jul 21 18:30:00 EDT 2015
Clouds:         clouds and visibility OK
Temperature:    13 C
Humidity:       93.0 %
Wind Speed:     mph
Dew Point:      12.0 C
```

Input dialog box: Enter ICAO Code, OK, Cancel