The Gog We Trust

Second Assignment Report

<Mohammad Ghaderi>

<401222104>

Introduction:

In this assignment we are going to program a weather app that it has only one input and it's a name for the city that you want to know it's weather information.

- Well I think this assignment has three objectives:
    1. First getting to know what git is(at least understand it basically)
    2. In this assignment we used API to make a URL to send a request to the weather site to get the information in the String format then extract the data we need(Getting know what API is.).
    3. Learn how to write a report.

- The approach which I have used(without considering GUI) was only write some simple functions for extracting things we need (for example: extract the temperature from the data that site gave to us.)

Design and Implementation:

This program is written by Java programming language.

The whole idea is that when the user input the city name we will make URL with our API_Key witch is different for everybody.

After that user input the city name and URL is maked we will send a request to the weather site(https://www.weatherapi.com/)

And it will send back a String witch has all the data of weather and some other information about the city in itself.

We will extract information which is interested to us (Like temperature) by "JsonObject".

And for this extracting information I have written 9 functions(Regardless of Main function and GUI.)

Functions which has been used are:

1. currentLocation function will extract the location that user entered.

```
54      public static String currentLocation(String weatherJson)
55      {
56          JSONObject json = new JSONObject(weatherJson);
57          String regin;
58          regin = json.getJSONObject( key: "location").getString( key: "region");
59          return regin;
60      }
```

2. localTime function will extract and return the date and time of the current location that user entered:

```
62      public static String localTime(String weatherJson)
63      {
64          JSONObject json = new JSONObject(weatherJson);
65          String local_time;
66          local_time = json.getJSONObject( key: "location").getString( key: "localtime");
67          return local_time;
68      }
```

3. getTemperature function will extract and return the temperature (in celsius) from the "Json String" that weather site gave to us :

```
70      public static double getTemperature(String weatherJson){
71          JSONObject json = new JSONObject(weatherJson);
72          double answer = 0.0;
73          answer = json.getJSONObject( key: "current").getDouble( key: "temp_c");
74          return answer;
75      }
```

4. getFeelsLikeTem function will extract and return the temperature that feels like in the current location:

```
77      public static double getFeelsLikeTem (String weatherJson)
78      {
79          JSONObject json = new JSONObject(weatherJson);
80          double feelsLikeTem = 0.0;
81          feelsLikeTem = json.getJSONObject( key: "current").getDouble( key: "feelslike_c");
82          return feelsLikeTem;
83      }
```

5. getHumidity function will extract and return the humidity in the current location which user entered:

```
85      public static int getHumidity(String weatherJson){
86          JSONObject json = new JSONObject(weatherJson);
87          int answer = 0;
88          answer = json.getJSONObject( key: "current").getInt( key: "humidity");
89          return answer;
90      }
```

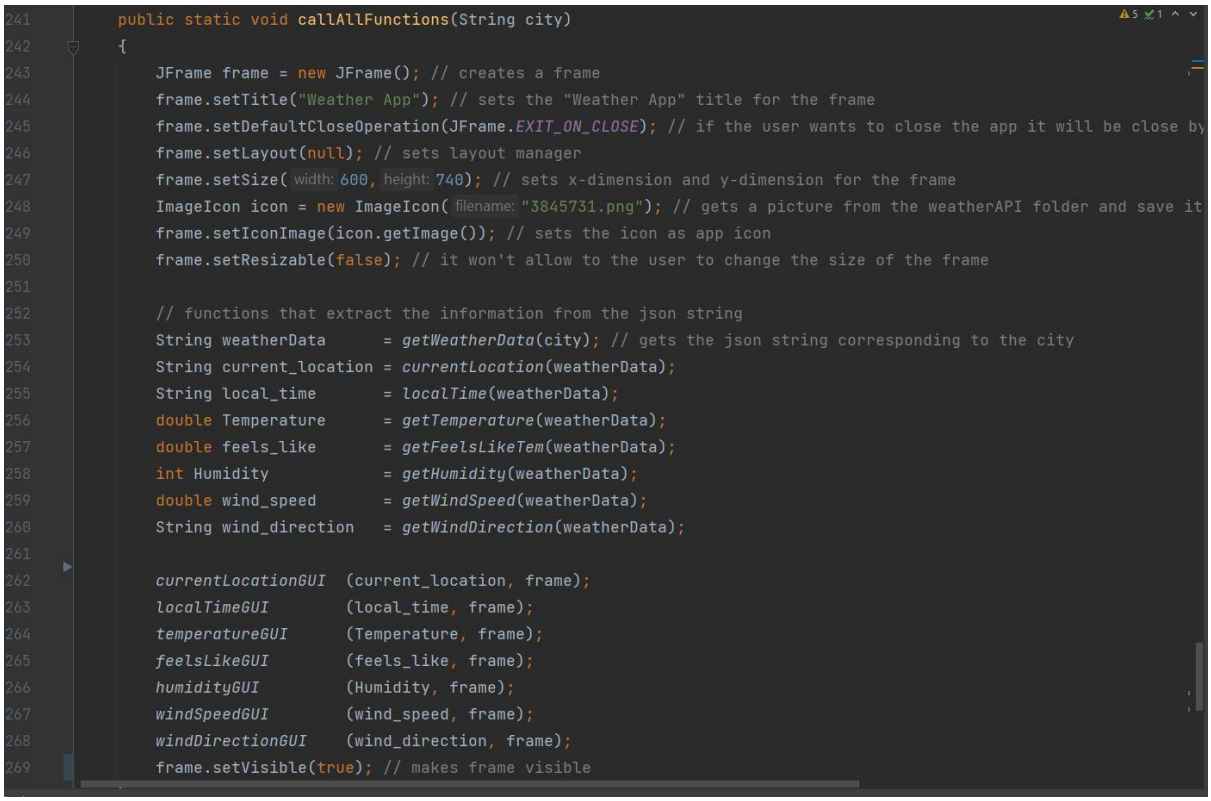
6. getWindSpeed function will extract and return wind's speed in the current location which user entered:

```
92      public static double getWindSpeed(String weatherJson)
93      {
94          JSONObject json = new JSONObject(weatherJson);
95          double wind_speed = 0.0;
96          wind_speed = json.getJSONObject( key: "current").getDouble( key: "wind_kph");
97          return wind_speed;
98      }
```

7. getWindDirection function will extract and return wind's direction in the current location which user entered:

```
100     public static String getWindDirection(String weatherJson)
101     {
102         JSONObject json = new JSONObject(weatherJson);
103         String wind_dir;
104         wind_dir = json.getJSONObject( key: "current").getString( key: "wind_dir");
105         return wind_dir;
106     }
```

8. callAllFunctions function will call all the functions we have written even those functions that are for GUI.



```
public static void callAllFunctions(String city)
{
    JFrame frame = new JFrame(); // creates a frame
    frame.setTitle("Weather App"); // sets the "Weather App" title for the frame
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // if the user wants to close the app it will be close by
    frame.setLayout(null); // sets layout manager
    frame.setSize( width: 600, height: 740); // sets x-dimension and y-dimension for the frame
    ImageIcon icon = new ImageIcon( filename: "3845731.png"); // gets a picture from the weatherAPI folder and save it
    frame.setIconImage(icon.getImage()); // sets the icon as app icon
    frame.setResizable(false); // it won't allow to the user to change the size of the frame

    // functions that extract the information from the json string
    String weatherData      = getWeatherData(city); // gets the json string corresponding to the city
    String current_location = currentLocation(weatherData);
    String local_time       = localTime(weatherData);
    double Temperature      = getTemperature(weatherData);
    double feels_like       = getFeelsLikeTem(weatherData);
    int Humidity            = getHumidity(weatherData);
    double wind_speed       = getWindSpeed(weatherData);
    String wind_direction   = getWindDirection(weatherData);

    currentLocationGUI  (current_location, frame);
    localTimeGUI        (local_time, frame);
    temperatureGUI      (Temperature, frame);
    feelsLikeGUI        (feels_like, frame);
    humidityGUI         (Humidity, frame);
    windSpeedGUI        (wind_speed, frame);
    windDirectionGUI    (wind_direction, frame);
    frame.setVisible(true); // makes frame visible
```

9. invaliInput function will return and make a graphical panel and display it to the user when the city name that user has entered in not find and invalid!

```
public static void invalidInput()
{
    Border border = BorderFactory.createLineBorder(Color.cyan, thickness: 5);

    JLabel label = new JLabel(); // creates a label
    label.setText("The location that you have entered is INVALID!"); // sets a label inside the frame.
    label.setHorizontalAlignment(JLabel.CENTER); // sets the label in the center of the frame
    label.setForeground(Color.BLACK);//sets a color for the text label
    label.setFont(new Font( name: "MV Boil", Font.PLAIN,  size: 20));// sets a font for the text label
    label.setBackground(Color.PINK);// sets background color
    label.setOpaque(true); // display background color
    label.setBorder(border);

    JFrame frame = new JFrame(); //creates a frame
    frame.setTitle("Weather App"); //sets title for the frame
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // if we press the 'x' icon on the top right the frame will b
    frame.setSize( width: 600, height: 500); // sets the x-dimension and y-dimension for the frame
    frame.setResizable(false); // the size of the frame will not be changeable
    frame.setVisible(true); // makes frame be visible
    frame.add(label); // add the label to the frame
    ImageIcon image = new ImageIcon( filename: "3845731.png"); // creates an Image Icon
    frame.setIconImage(image.getImage()); // sets the Image Icon
}
```

An overview on other functions:

1. getWeatherData function:
   What this function dose is that make an URL and the API-key that you get

from your account on weather site the city name .
After that this function will send a request to the weather site and then get a "json string" as a response.
It’s the same "json string" that we were talking about in a few pages back.
If the city name was incorrect then this function will return a null string.

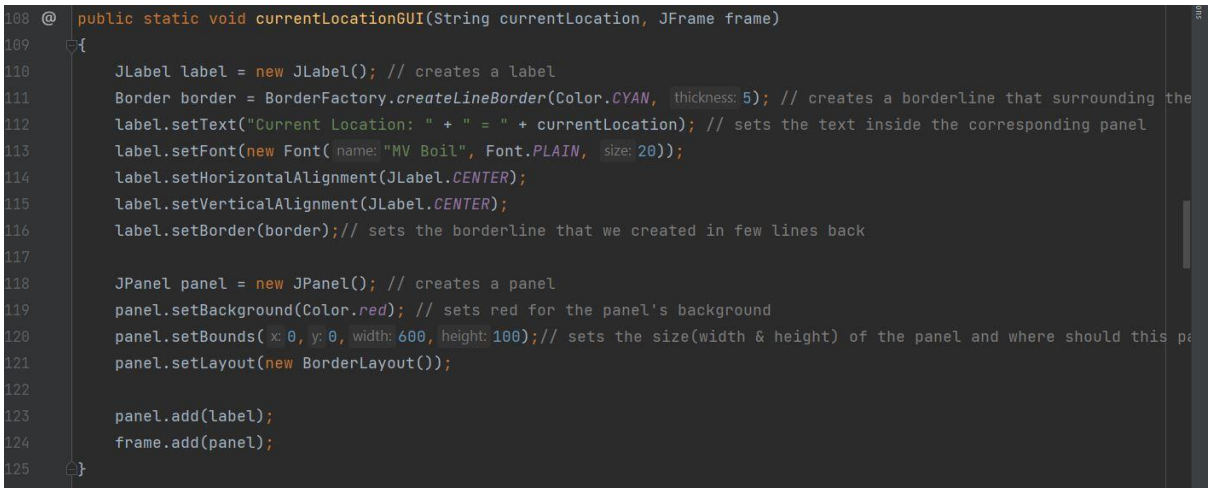
```
public static String getWeatherData(String city) {
    try {
        URL url = new URL( spec: "http://api.weatherapi.com/v1/current.json?key=" + apiKey + "&q=" + city);
        HttpURLConnection connection = (HttpURLConnection) url.openConnection();
        connection.setRequestMethod("GET");
        BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));
        StringBuilder stringBuilder = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            stringBuilder.append(line);
        }
        reader.close();
        return stringBuilder.toString();
    } catch (Exception e) {
//          e.printStackTrace();
        return null;
    }
}
```

2. GUI functions:

We have 7 functions that help us to show our information in graphical frame.

Here I will explain the first one of these functions because all of these 7 functions are the same:

currentLocationGUI function:



```
public static void currentLocationGUI(String currentLocation, JFrame frame)
{
    JLabel label = new JLabel(); // creates a label
    Border border = BorderFactory.createLineBorder(Color.CYAN, thickness: 5); // creates a borderline that surrounding the
    label.setText("Current Location: " + " = " + currentLocation); // sets the text inside the corresponding panel
    label.setFont(new Font( name: "MV Boil", Font.PLAIN, size: 20));
    label.setHorizontalAlignment(JLabel.CENTER);
    label.setVerticalAlignment(JLabel.CENTER);
    label.setBorder(border);// sets the borderline that we created in few lines back

    JPanel panel = new JPanel(); // creates a panel
    panel.setBackground(Color.red); // sets red for the panel's background
    panel.setBounds( x: 0, y: 0, width: 600, height: 100);// sets the size(width & height) of the panel and where should this pa
    panel.setLayout(new BorderLayout());

    panel.add(label);
    frame.add(panel);
}
```

1. What JLabel do?
   JLabel helps us to make a label here. With labels we can set texts,pictures and borderlines in the graphical panel we are going to display to the user.
2. What Border do?
   With border here we can make a borderline that we can choose it's color and thickness.

3. What JPanel do?
   Well with JPanel we can make a graphical panel and show label in it.