Reverse Geocoding

Android Geocoder class

- Android Geocoder class is used **for Geocoding as well as Reverse Geocoding**.
- Geocoding refers to transforming street address or any address into latitude and longitude.
- Reverse Geocoding refers to transforming latitude and longitude into its corresponding street address.

Locale class

- A Locale object represents a specific geographical, political, or cultural region.
- An operation that requires a Locale to perform its task is called *locale-sensitive* and uses the Locale to tailor information for the user.
- For example, displaying a number is a locale-sensitive operation— the number should be formatted according to the customs and conventions of the user's native country, region, or culture.
- Once you've created a Locale you can query it for information about itself. Use getCountry to get the country (or region) code and getLanguage to get the language code.
- You can use getDisplayCountry to get the name of the country suitable for displaying to the user.
- **Locale getDefault() method**
  - This method returns default Locale set by the Java Virtual Machine. This is static method so it can be called without creating object of the class Locale.

Address class

- Address class helps in fetching the street address, locality, sub-locality, city, country, landmark etc. features of the location.

- The method **getFromLocation(double latitude, double longitude, int maxResults)** returns a List of Addresses for the current location.
- Within the Address object, the method **getAddressLine(int index)** returns a line of the address numbered by the given index or null if no address exists.
- We append that address to a StringBuilder which is eventually displayed in the TextView.
- **getMaxAddressLineIndex()** returns the largest index currently in use to specify an address line.

Create an android application that displays the current location of your device from longitude and latitude values(Reverse Geocoding).

AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.getlocationaddress">
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-
permission>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-
permission>
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.GetLocationAddress">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="155dp"
        android:text="TextView" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Check Location" />
</LinearLayout>
```

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import java.util.List;
import java.util.Locale;

public class MainActivity extends AppCompatActivity implements LocationListener {
    Button button;
    TextView textView;
    LocationManager locationManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView = findViewById(R.id.textView);
        button = findViewById(R.id.button);
        if (ContextCompat.checkSelfPermission(MainActivity.this, android.Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(MainActivity.this, new String[]{
                    android.Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.INTERNET}, 100);
        }
```

```
 button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                getLocation();
            }
        });
    }

    @SuppressLint("MissingPermission")
    private void getLocation() {
        try {
            locationManager = (LocationManager)
getApplicationContext().getSystemService(LOCATION_SERVICE);
            locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 5000, 5,
(LocationListener) this);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Override
    public void onLocationChanged(@NonNull Location location) {
        String myLocation = getAddress(location.getLatitude(), location.getLongitude());
        textView.setText("Latitude:" + location.getLatitude() + "\nLongitude:" +
location.getLongitude()+"\n\n"+myLocation);
    }
```

```
private String getAddress(double latitude, double longitude) {
    String address = "";
    Geocoder geocoder = new Geocoder(this, Locale.getDefault());
    try {
      List<Address> addresses = geocoder.getFromLocation(latitude, longitude, 1);
      if (address != null) {
        Address returnaddress = addresses.get(0); //return returnaddress.toString();
        StringBuilder stringBuilderAddress = new StringBuilder("");
        for (int i = 0; i <= returnaddress.getMaxAddressLineIndex(); i++) {
          stringBuilderAddress.append(returnaddress.getAddressLine(i)).append("\n");
        }
        address = stringBuilderAddress.toString();
        return address;
      } else {
        return "Address Not Found";
      }
    } catch (Exception e) {
      Toast.makeText(this, e.getMessage().toString(), Toast.LENGTH_LONG).show();
    }
    return address;
  }
}
```