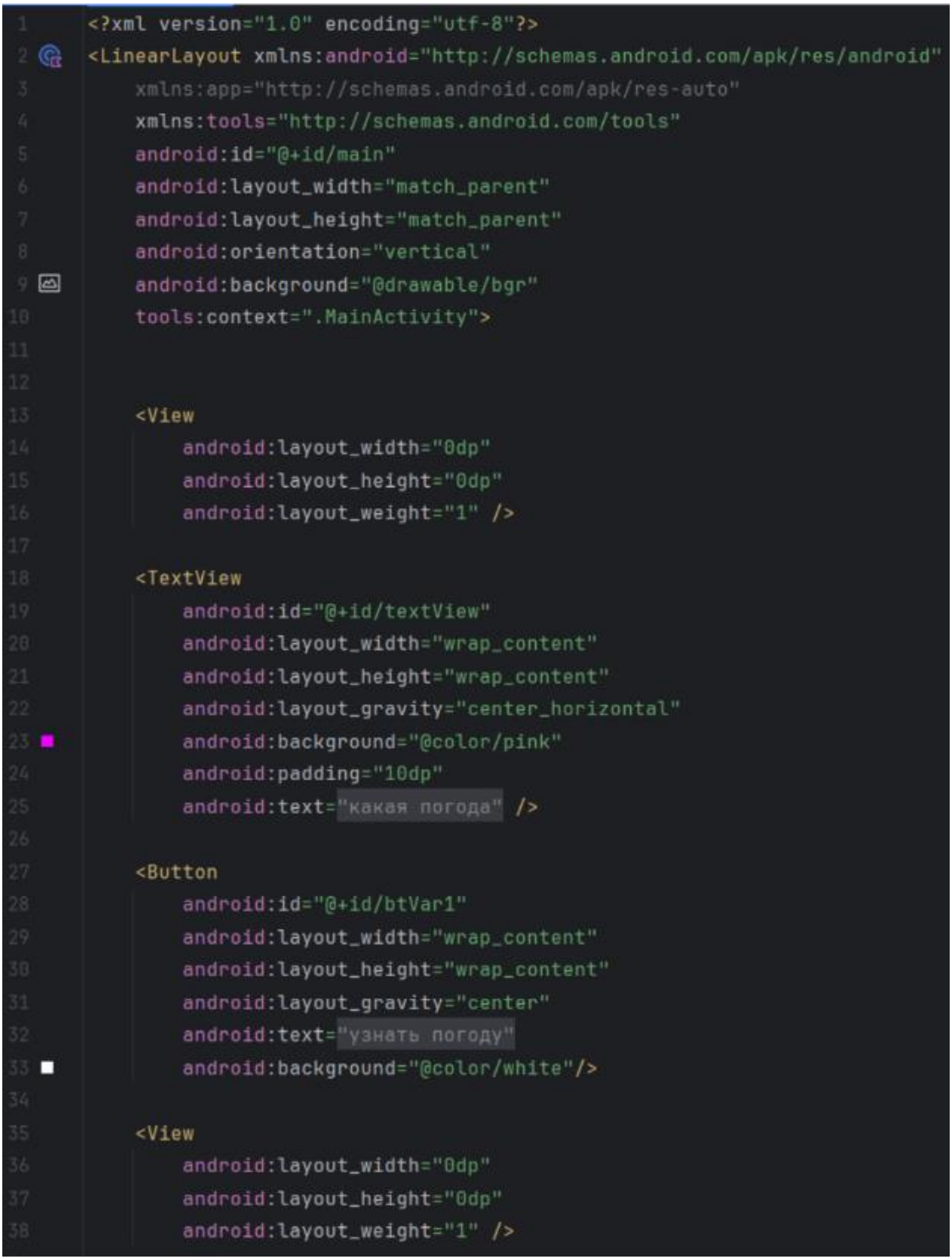
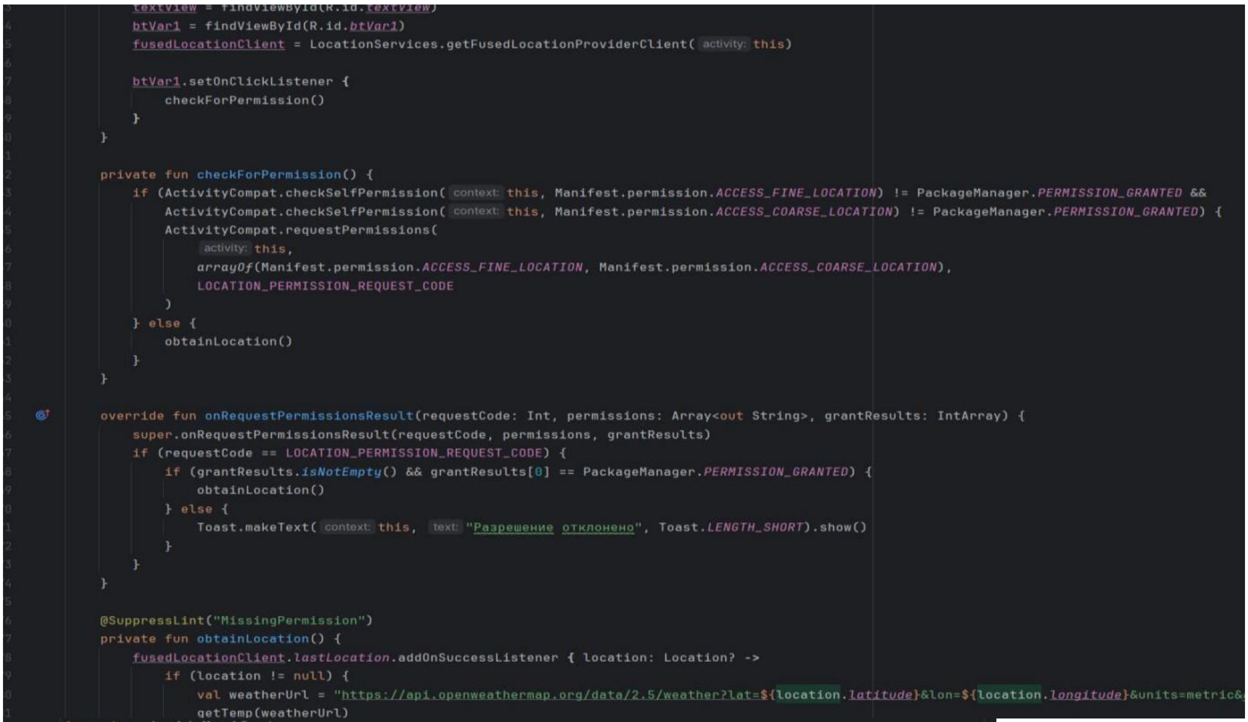
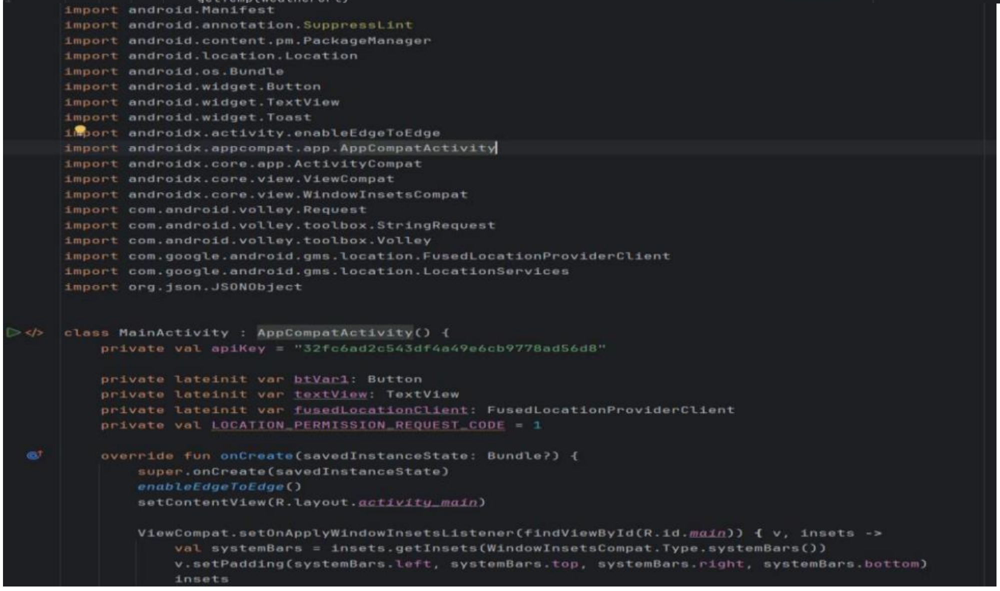


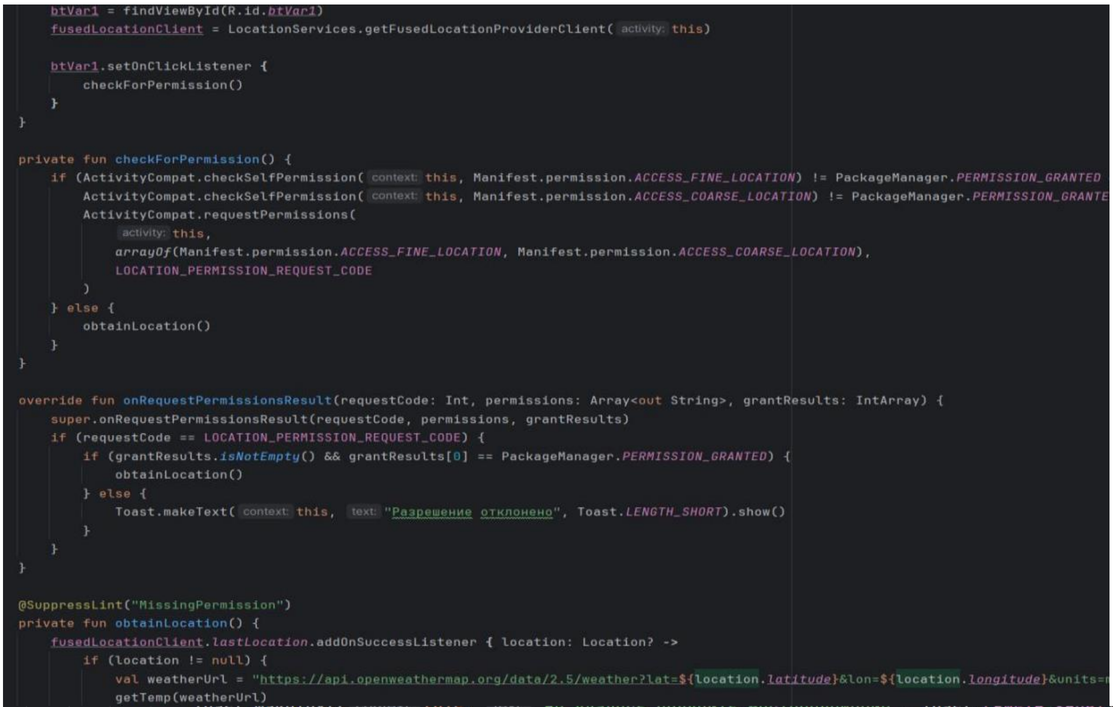
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/bgr"
    tools:context=".MainActivity">


    <View
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_weight="1" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:background="@color/pink"
        android:padding="10dp"
        android:text="какая погода" />

    <Button
        android:id="@+id/btVar1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="узнать погоду"
        android:background="@color/white"/>

    <View
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_weight="1" />
```



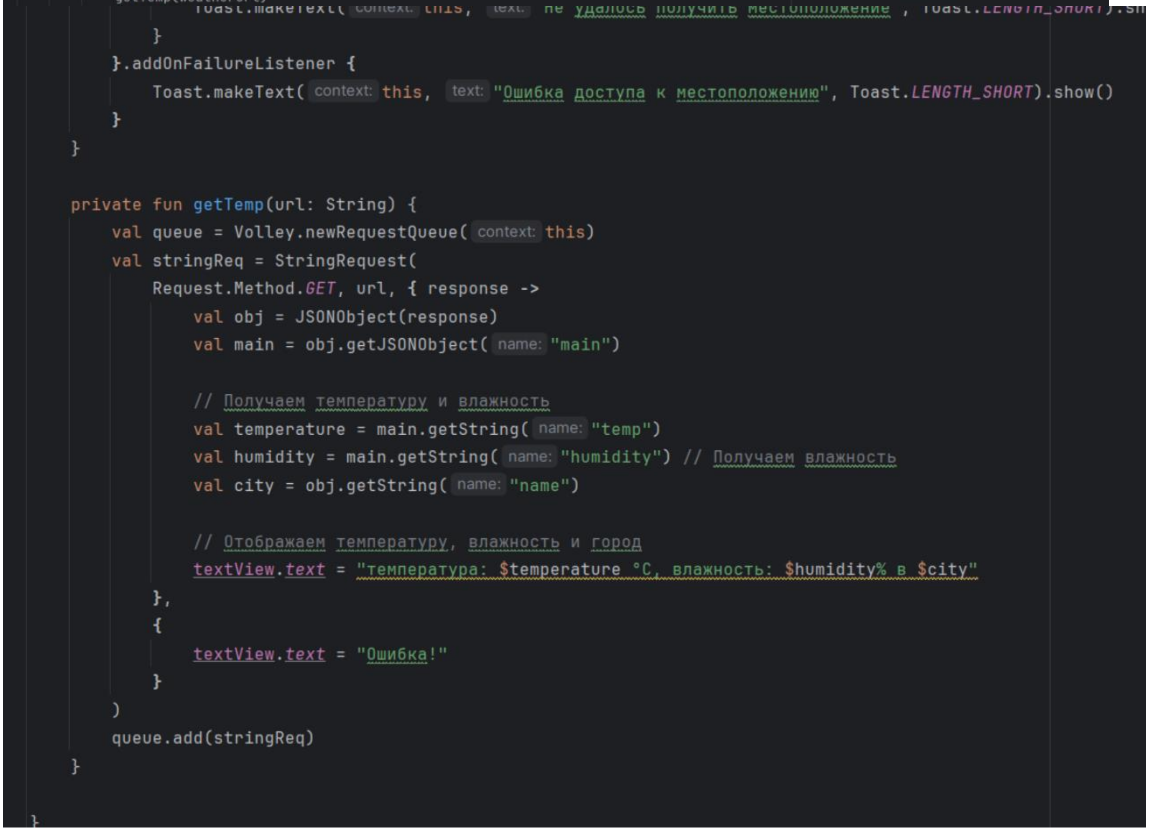
```
        textView = findViewById(R.id.textView)
        btVar1 = findViewById(R.id.btVar1)
        fusedLocationClient = LocationServices.getFusedLocationProviderClient( activity: this)

        btVar1.setOnClickListener {
            checkForPermission()
        }
    }

    private fun checkForPermission() {
        if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
            ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(
                 activity: this,
                arrayOf(Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.ACCESS_COARSE_LOCATION),
                LOCATION_PERMISSION_REQUEST_CODE
            )
        } else {
            obtainLocation()
        }
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
            if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                obtainLocation()
            } else {
                Toast.makeText( context: this,  text: "Разрешение отклонено", Toast.LENGTH_SHORT).show()
            }
        }
    }

    @SuppressLint("MissingPermission")
    private fun obtainLocation() {
        fusedLocationClient.lastLocation.addOnSuccessListener { location: Location? ->
            if (location != null) {
                val weatherUrl = "https://api.openweathermap.org/data/2.5/weather?lat=${location.latitude}&lon=${location.longitude}&units=metric&
                getTemp(weatherUrl)
```



```
import android.Manifest
import android.annotation.SuppressLint
import android.content.pm.PackageManager
import android.location.Location
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.android.volley.Request
import com.android.volley.toolbox.StringRequest
import com.android.volley.toolbox.Volley
import com.google.android.gms.location.FusedLocationProviderClient
import com.google.android.gms.location.LocationServices
import org.json.JSONObject


class MainActivity : AppCompatActivity() {
    private val apiKey = "32fc6ad2c543df4a49e6cb9778ad56d8"

    private lateinit var btVar1: Button
    private lateinit var textView: TextView
    private lateinit var fusedLocationClient: FusedLocationProviderClient
    private val LOCATION_PERMISSION_REQUEST_CODE = 1

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
```

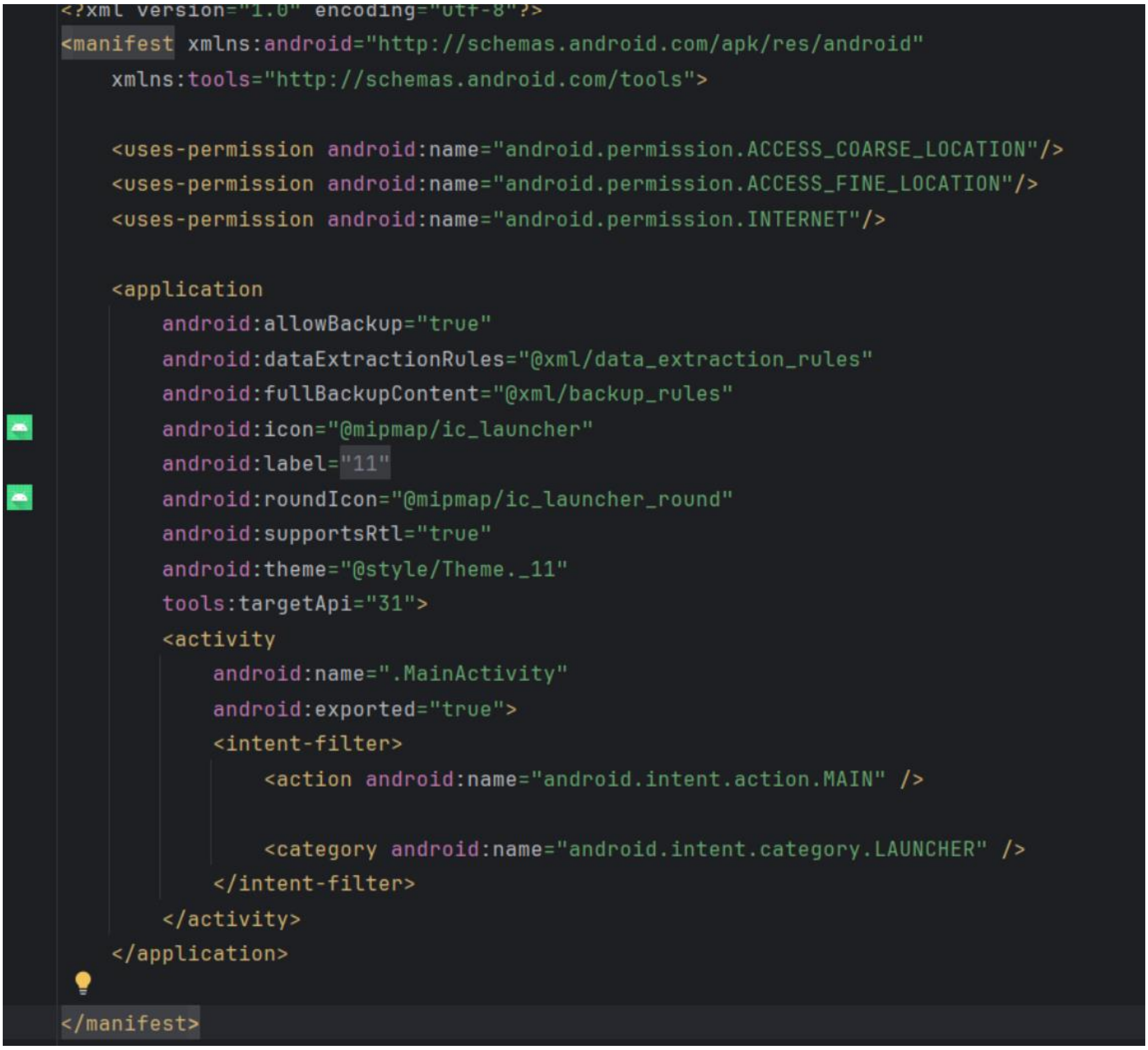


```
        btVar1 = findViewById(R.id.btVar1)
        fusedLocationClient = LocationServices.getFusedLocationProviderClient( activity: this)

        btVar1.setOnClickListener {
            checkForPermission()
        }
    }

    private fun checkForPermission() {
        if (ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
            ActivityCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTE
            ActivityCompat.requestPermissions(
                 activity: this,
                arrayOf(Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.ACCESS_COARSE_LOCATION),
                LOCATION_PERMISSION_REQUEST_CODE
            )
        } else {
            obtainLocation()
        }
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
            if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                obtainLocation()
            } else {
                Toast.makeText( context: this,  text: "Разрешение отклонено", Toast.LENGTH_SHORT).show()
            }
        }
    }

    @SuppressLint("MissingPermission")
    private fun obtainLocation() {
        fusedLocationClient.lastLocation.addOnSuccessListener { location: Location? ->
            if (location != null) {
                val weatherUrl = "https://api.openweathermap.org/data/2.5/weather?lat=${location.latitude}&lon=${location.longitude}&units=
                getTemp(weatherUrl)
```



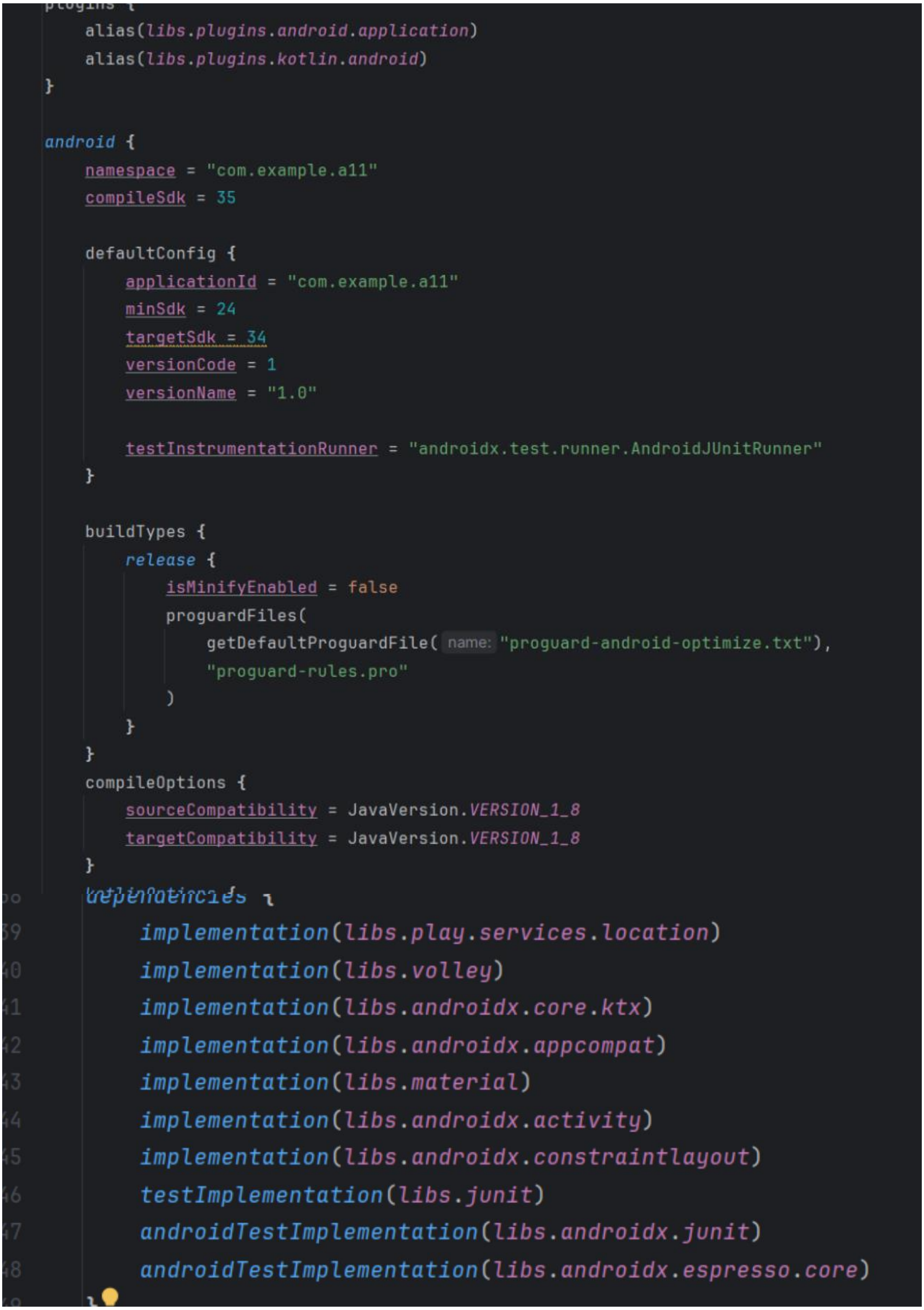
```
                Toast.makeText( context: this,  text: "Не удалось получить местоположение", Toast.LENGTH_SHORT).sh
            }
        }.addOnFailureListener {
            Toast.makeText( context: this,  text: "Ошибка доступа к местоположению", Toast.LENGTH_SHORT).show()
        }
    }

    private fun getTemp(url: String) {
        val queue = Volley.newRequestQueue( context: this)
        val stringReq = StringRequest(
            Request.Method.GET, url, { response ->
                val obj = JSONObject(response)
                val main = obj.getJSONObject( name: "main")

                // Получаем температуру и влажность
                val temperature = main.getString( name: "temp")
                val humidity = main.getString( name: "humidity") // Получаем влажность
                val city = obj.getString( name: "name")

                // Отображаем температуру, влажность и город
                textView.text = "температура: $temperature °C, влажность: $humidity% в $city"
            },
            {
                textView.text = "Ошибка!"
            }
        )
        queue.add(stringReq)
    }

}
```

код в AndroidManifest.xml



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="11"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme._11"
        tools:targetApi="31">
        <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>
```

build.gradle.kts



```
plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
}

android {
    namespace = "com.example.a11"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.a11"
        minSdk = 24
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile( name: "proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
    dependencies {
        implementation(libs.play.services.location)
        implementation(libs.volley)
        implementation(libs.androidx.core.ktx)
        implementation(libs.androidx.appcompat)
        implementation(libs.material)
        implementation(libs.androidx.activity)
        implementation(libs.androidx.constraintlayout)
        testImplementation(libs.junit)
        androidTestImplementation(libs.androidx.junit)
        androidTestImplementation(libs.androidx.espresso.core)
    }
```