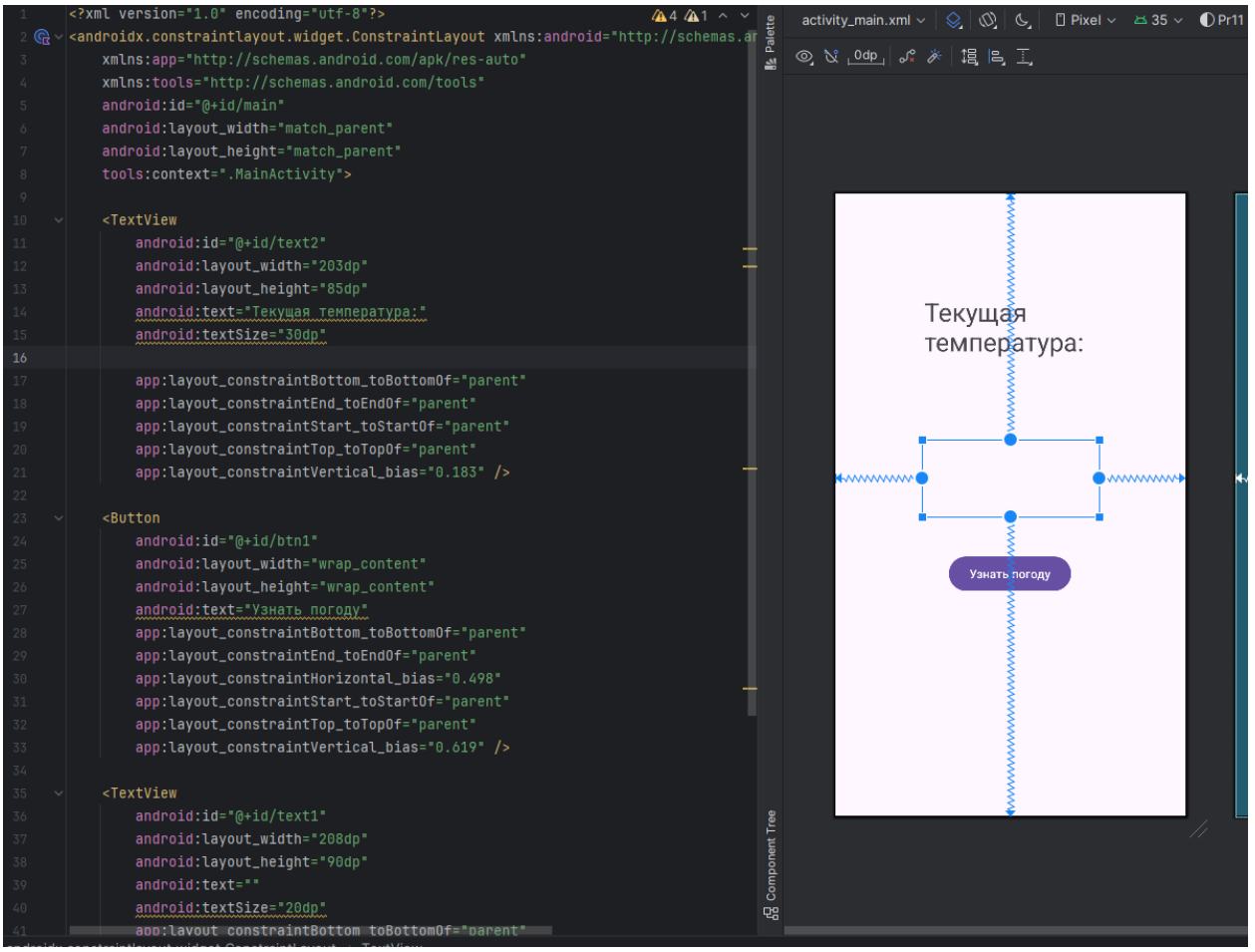
Практическая работа 11

1. Добавляем разрешение в манифест

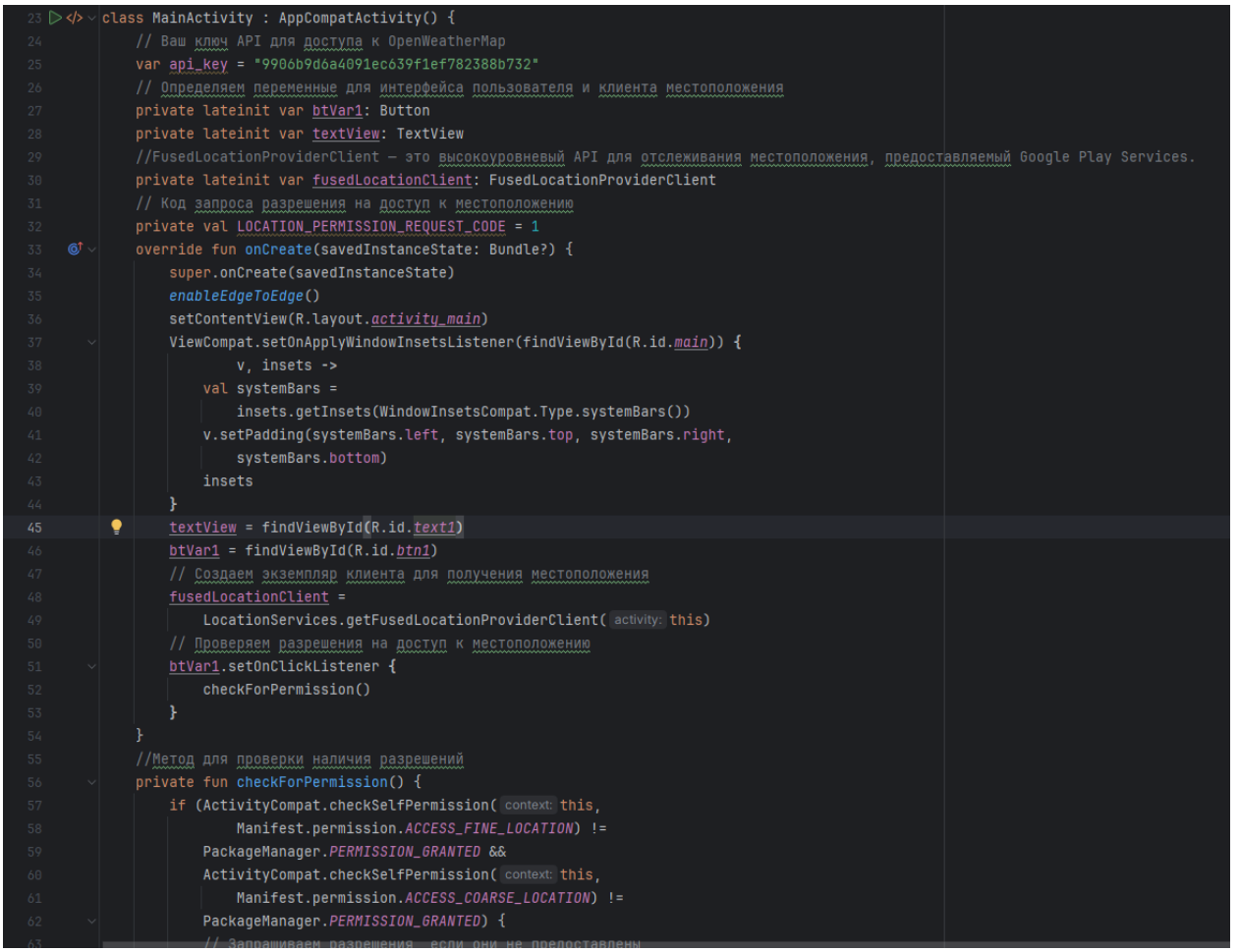
```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <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="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Pr11"
        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>
```

2. Activity_main




```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.ar
    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"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/text2"
        android:layout_width="203dp"
        android:layout_height="85dp"
        android:text="Текущая температура:"
        android:textSize="30dp"

        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.183" />

    <Button
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Узнать погоду"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.619" />

    <TextView
        android:id="@+id/text1"
        android:layout_width="208dp"
        android:layout_height="90dp"
        android:text=""
        android:textSize="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
```

3. MainActivity



```
class MainActivity : AppCompatActivity() {
    // Ваш ключ API для доступа к OpenWeatherMap
    var api_key = "990bb9d6a4091ec639f1ef782388b732"
    // Определяем переменные для интерфейса пользователя и клиента местоположения
    private lateinit var btVar1: Button
    private lateinit var textView: TextView
    //FusedLocationProviderClient — это высокоуровневый API для отслеживания местоположения, предоставляемый Google Play Services.
    private lateinit var fusedLocationClient: FusedLocationProviderClient
    // Код запроса разрешения на доступ к местоположению
    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
        }
        textView = findViewById(R.id.text1)
        btVar1 = findViewById(R.id.btn1)
        // Создаем экземпляр клиента для получения местоположения
        fusedLocationClient =
            LocationServices.getFusedLocationProviderClient( activity: this)
        // Проверяем разрешения на доступ к местоположению
        btVar1.setOnClickListener {
            checkForPermission()
        }
    }
    //Метод для проверки наличия разрешений
    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) {
            // Запрашиваем разрешения, если они не предоставлены
            ActivityCompat.requestPermissions( activity: this,
                arrayOf(Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.ACCESS_COARSE_LOCATION),  LOCATION_PERMISSION_REQUEST_CODE)
        } else {
            // Если разрешения уже предоставлены, получаем местоположение
            obtainLocation()
        }
    }
    // Метод, который обрабатывает результат запроса разрешений
    override fun onRequestPermissionsResult(requestCode: Int, permissions:
    Array<out String>, grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions,
            grantResults)
        // Проверяем код запроса разрешений
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
            // Если разрешение было предоставлено
            if ((grantResults.isNotEmpty() && grantResults[0] ==
                        PackageManager.PERMISSION_GRANTED)) {
                // Получаем местоположение
                obtainLocation()
            } else {
                // Если разрешение было отклонено, показываем сообщение
                Toast.makeText( context: this,  text: "Разрешение отклонено",
                    Toast.LENGTH_SHORT).show()
            }
        }
    }
    @SuppressLint("MissingPermission")
    private fun obtainLocation() {
        // Получаем последнее известное местоположение
        fusedLocationClient.lastLocation
            .addOnSuccessListener { location: Location? ->
                // Проверяем, что местоположение не равно null
                if (location != null) {
                    // Формируем URL для запроса к API погоды с текущими координатами
                    val weatherUrl =
```

```
                    "https://api.openweathermap.org/data/2.5/weather?lat=${location.latitude}&lon=${location.longitude}&units=metric&appid=${api_key}"
                // Запрашиваем температуру по текущему местоположению
                getTemp(weatherUrl)
            } else {
                // Если не удалось получить местоположение, выводим сообщение
                Toast.makeText( context: this,  text: "Не удалось получить местоположение", Toast.LENGTH_SHORT).show()
            }
        }
    //это метод, который используется в Android для обработки неудачных результатов асинхронных операций
        .addOnFailureListener { exception ->
        // Если не удалось получить местоположение, показываем сообщение
        Toast.makeText( context: this,  text: "Location Permission not granted",
            Toast.LENGTH_SHORT).show()
    }
}
private fun getTemp(url: String) {
    // Создаем очередь для запросов
    //Volley - это HTTP-библиотека, которая используется для
    // кэширования и выполнения сетевого запроса в приложениях Android.
    val queue = Volley.newRequestQueue( context: this)
    // Выполняем запрос к API погоды
    val stringReq = StringRequest(
        Request.Method.GET, url, { response ->
            // Получаем JSON-объект из ответа
            val obj = JSONObject(response)
            // Получаем данные о температуре из объекта
            val main: JSONObject = obj.getJSONObject( name: "main")
            val temperature = main.getString( name: "temp")
            println(temperature)
            // Получаем название города из объекта
            val city = obj.getString( name: "name")
            println(city)
            // Устанавливаем текст в textView, показывая температуру и название города
                textView.text = "${temperature} Градусов по цельсию в ${city}"
            System.out.println(obj.toString())
        },
```

```
            // В случае ошибки показываем сообщение
            { textView.text = "Ошибка!" })
        // Добавляем запрос в очередь
        queue.add(stringReq)
    }
}
```

4. Результат