Лабораторная работа №11. Иванчук Вячеслав Сергеевич

MainACtivity.kt

```
class MainActivity : AppCompatActivity() {
    var api_key = "3ffed4367e72088a72d07abd46fc8eeb"
    private lateinit var btVar1: Button
    private lateinit var textView: TextView
    private lateinit var fusedLocationClient:
FusedLocationProviderClient
    private val LOCATION_PERMISSION_REQUEST_CODE = 1
    @SuppressLint("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
//
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.i
d.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.textF)
        btVar1 = findViewById(R.id.btVar1)
        fusedLocationClient =

LocationServices.getFusedLocationProviderClient(this)
        btVar1.setOnClickListener {
            checkForPermission()
        }
    }
    //Метод для проверки наличия разрешений
```

```
    private fun checkForPermission() {
        if (ActivityCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
!=
            PackageManager.PERMISSION_GRANTED &&
            ActivityCompat.checkSelfPermission(this,

Manifest.permission.ACCESS_COARSE_LOCATION) !=
            PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(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)) {
                // Получаем местоположение
                obtainLocation()
            } else {
                Toast.makeText(this, "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&appid=${api_key}"
                    getTemp(weatherUrl)
                } else {
                    Toast.makeText(this, "Не удалось получить местоположение", Toast.LENGTH_SHORT).show()
                }
            }
            .addOnFailureListener {
                Toast.makeText(this, "Location Permission not granted",
                    Toast.LENGTH_SHORT).show()
            }
    }
    private fun getTemp(url: String) {
        val queue = Volley.newRequestQueue(this)
        val stringReq = StringRequest(
            Request.Method.GET, url, { response ->
                val obj = JSONObject(response)
                val main: JSONObject =
obj.getJSONObject("main")
                val temperature = main.getString("temp")
                println(temperature)
                val city = obj.getString("name")
```

```
                println(city)
                textView.text = "${temperature} Градусов
по цельсию в ${city}"
                System.out.println(obj.toString())
            },
            { textView.text = "Ошибка!" })
        queue.add(stringReq)
    }
}
```