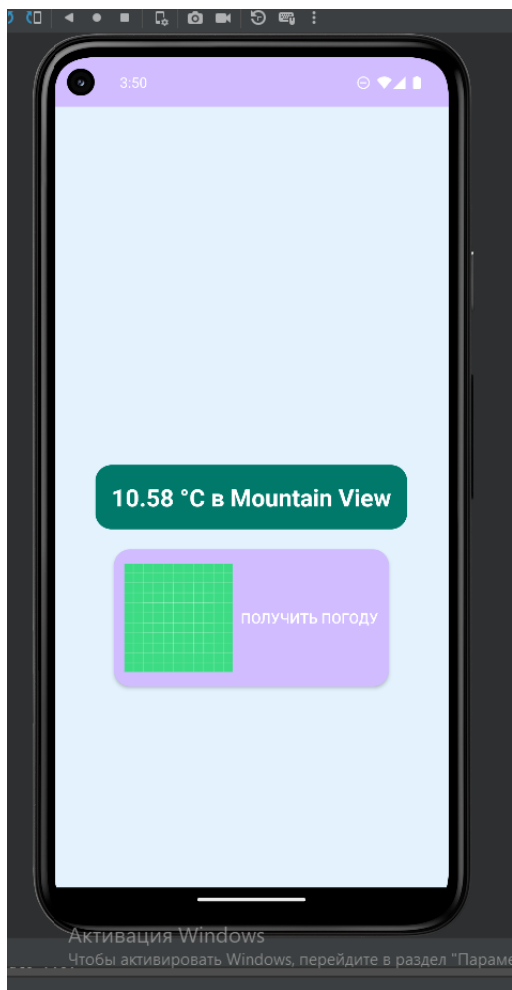


Практическая 11

Меин

```
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {  
    }  
  
    @SuppressWarnings("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( context: this, text: "Не удалось получить местоположение", Toast.LENGTH_SHORT).show()  
                }  
            }.addOnFailureListener { exception ->  
                Toast.makeText( context: this, text: "Location Permission not granted", Toast.LENGTH_SHORT).show()  
            }  
    }  
  
    @SuppressWarnings("SetTextI18n")  
    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: JSONObject = obj.getJSONObject( name: "main")  
            val temperature = main.getString( name: "temp")  
            val city = obj.getString( name: "name")  
            textView.text = "${temperature} °C в ${city}"  
        }, { textView.text = "Ошибка!" })  
  
        queue.add(stringReq)  
    }  
}
```

Мобилко



1. Регистрация и получение API-ключа: Зарегистрировавшись на сайте OpenWeatherMap, вы получили API-ключ, необходимый для выполнения запросов к их погодному сервису.

2. Разрешения в Android Manifest: В файл AndroidManifest.xml были добавлены необходимые разрешения для обеспечения доступа к интернету и местоположению:

- ACCESS_COARSE_LOCATION — для определения приблизительного местоположения.
- ACCESS_FINE_LOCATION — для получения более точных координат.
- INTERNET — для подключения к интернету.

3. Дизайн интерфейса: Создан макет в файле activity_main.xml, который включает в себя кнопку для получения данных о погоде и текстовое поле для отображения температуры и влажности.

4. Подключение библиотек: Добавлены зависимости для обработки местоположения и выполнения сетевых запросов:

- implementation(libs.play.services.location) — для работы с функциями определения местоположения.
- implementation(libs.volley) — для выполнения асинхронных HTTP-запросов.

5. Реализация MainActivity: В классе MainActivity настроены элементы интерфейса и добавлен обработчик нажатий на кнопку. При нажатии начинается процесс определения местоположения и выполняется запрос к API, если местоположение успешно получено.

6. Проверка разрешений: Разработаны методы для проверки и запроса разрешений на использование местоположения. При предоставлении разрешений запускается метод obtainLocation() для определения текущих координат.

7. Получение местоположения: Использование FusedLocationProviderClient для получения последнего известного местоположения устройства. На основе полученных координат формируется URL для запроса к API.

8. Выполнение запроса к API: Создан метод getTemp(), который использует библиотеку Volley для выполнения запроса к API. Полученный ответ обрабатывается для извлечения данных о температуре и влажности из JSON-объекта.

9. Отображение данных: Температура и влажность, полученные из API, выводятся в текстовом поле TextView.