

Лабораторная работа №7. Работа с картами

Задание: создать приложение, отображающее после запуска карты Google или какие-нибудь другие карты.

Можно использовать разные карты: если недоступен ключ для Google Maps (хотя можно нагуглить обход этого ограничения), можно воспользоваться Yandex Maps или OpenStreetMaps

Google Maps

Создайте новый проект, но на этот раз выберите Google Maps Activity вместо Empty Activity. Для того, чтобы пользоваться картами Google, необходимо получить персональный ключ для приложения. Переходим в файл res/values/google_maps_api.xml (открывается по умолчанию), и в комментариях видим инструкцию на английском языке о том, как получить ключ для приложения. Если коротко, то необходимо перейти по персональной ссылке из комментария, залогиниться в свой Google аккаунт, нажать кнопку Continue и затем Create API key, после чего перед вами появится окно API key created с ключом, начинающимся на "Alza". Копируем значение ключа из поля и вставляем в файл res/values/google_maps_api.xml вместо фразы YOUR_KEY_HERE.

Теперь переходим в файл MapsActivity.java, и в самом последнем методе onMapReady добавляем следующее в начало, после строки mMap = googleMap;:

```
    LocationManager locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
```

Затем строчкой ниже начинаем писать LocationListener locationListener = new Loc и дальше выбираем первую подсказку. Появится 4 метода, в первом методе onLocationChanged пишем следующее:

```
Latlng latlng = new Latlng(location.getLatitude(), location.getLongitude());
mMap.clear();
mMap.moveCamera(CameraUpdateFactory.newLatlng(latlng));
mMap.animateCamera(CameraUpdateFactory.zoomTo(15));
mMap.addMarker(new MarkerOptions().position(latlng).title("I am here"));
```

После этих четырёх методов (после символа }; - не забудьте поставить точку с запятой) пишем следующее:

```
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0,
locationListener);
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, locationListener);
```

Здесь мы запрашиваем обновление местоположения по вышкам сотовой связи и по GPS. Код просит добавить проверку разрешений для пользователя, добавляем эту проверку (выделяем строку с ошибкой, слева видим красную лампочку с восклицательным знаком внутри, нажимаем на неё и выбираем Add permission check) и одновременно в файл манифеста добавляем соответствующие разрешения. Разрешение **ACCESS_FINE_LOCATION** уже есть, добавим недостающее:

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

Далее вернёмся в `MapsActivity.java` и в методе `onMapReady` закомментируем всё, что идёт в конце и относится к Сиднею (после `// Add a marker in Sydney and move the camera`), и после этого вставим две строчки:

```
mMap.setMyLocationEnabled(true);
mMap.getUiSettings().setZoomControlsEnabled(true);

mMap.setMapType(2);
```

Таким образом, на карте отобразится кнопка текущего местоположения и кнопки зума.

Прежде чем запускать приложение, необходимо создать новый эмулятор с поддержкой Google Play services (Google APIs), если у вас ещё нет такого эмулятора, так как на обычном эмуляторе карту вы не увидите, а увидите вместо неё надпись "Your_app_name won't run without Google Play services, which are not supported by your device.". Затем, возможно, после запуска приложения вы можете получить сообщение о том, что "Your_app_name won't run unless you update Google Play services", так как для работы приложения нужна версия 2 этих сервисов – нажмите кнопку Update в этом случае (кнопка будет работать только в том случае, если ваш эмулятор поддерживает функциональность Google Play). Лучше всего сконфигурировать эмулятор с одной из последних версий API (начиная с Lollipop) – тогда не надо будет обновлять Google Play services до последней версии. Если после запуска приложения появляется предложение установить Instant Run, надо согласиться.

Также помните о добавлении вашего ФИО в приложение – в этот раз это будет сделать сложнее, чем обычно, так как надо самостоятельно создать layout в файле `res -> layout -> activity_maps.xml`, обернуть в него fragment с картой и добавить элемент `TextView`.

Запускаем приложение, однако эмулятор не имеет данных о местоположении, но их можно ему передать! Для этого в панели инструментов справа от эмулятора нажмите на три точки внизу, после этого откроется окно, показанное на рисунке 30. Справа в окне видно, что можно ввести значения широты и долготы и передать их приложению с помощью кнопки SEND. Введите 55.769794 в поле `Longitude` и 37.655640 в поле `Latitude` (координаты Москвы) и нажмите кнопку SEND.

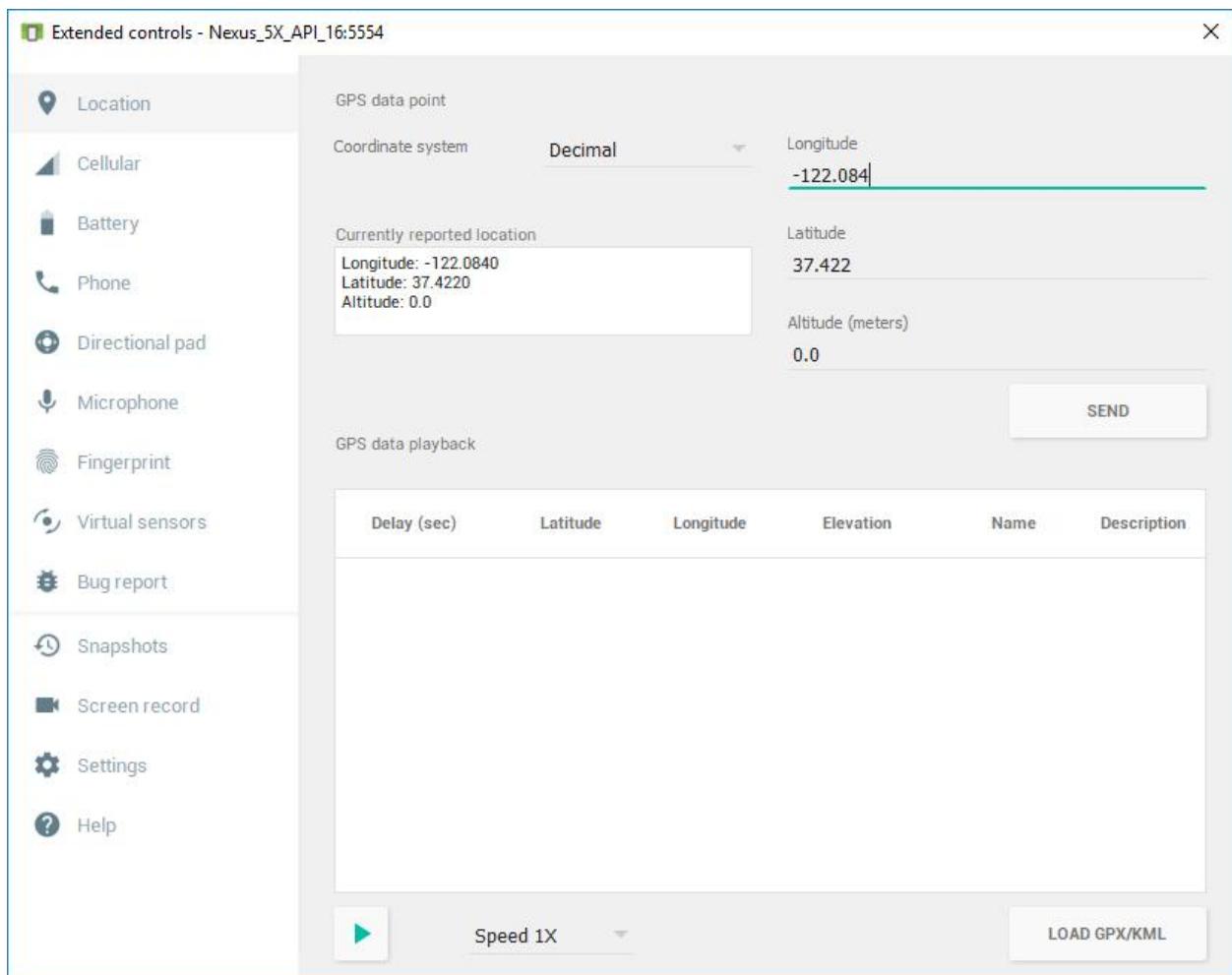


Рисунок 30 – Окно дополнительного управления эмулятором

Теперь эмулятор знает координаты и отображает своё местоположение по ним. Пример интерфейса работающего приложения показан на рисунках 31 и 32. Приложение можно запустить и на вашем реальном устройстве, на котором точно есть Google Play services.

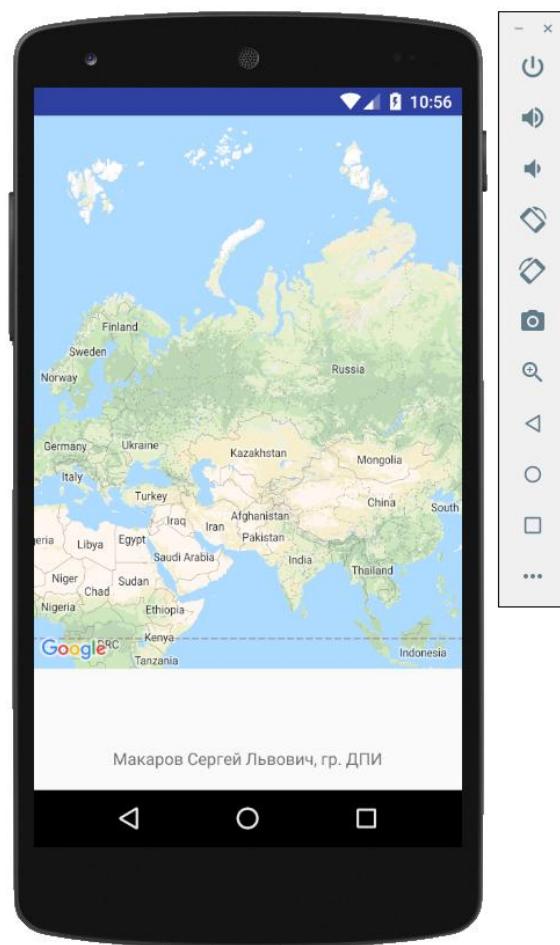


Рисунок 31 – Пример выполнения лабораторной работы №7 на эмуляторе

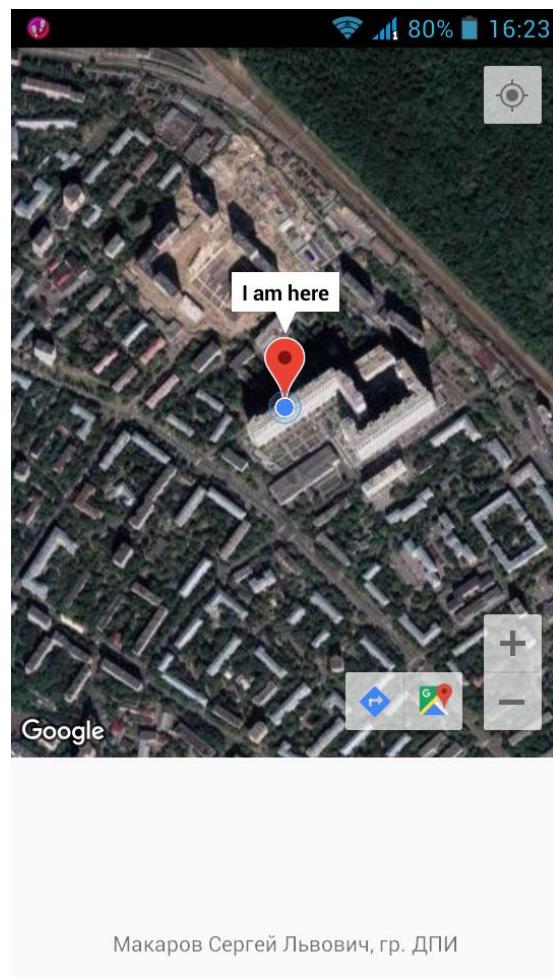


Рисунок 31 – Пример выполнения лабораторной работы №7 на смартфоне

Yandex Maps

Действуем согласно инструкции (она на Kotlin): https://yandex.ru/maps-api/docs/mapkit/android/generated/getting_started.html

Но: в файле settings.gradle.kts не надо ничего добавлять, там уже есть mavenCentral(). В файле build.gradle.kts(Module: app) добавим в dependencies (добавлять надо в скобках и двойных кавычках, а не как в инструкции, она у них устарела):

```
implementation("com.yandex.android:maps.mobile:4.5.0-lite")
```

Синхронизируем проект.

В методе onCreate **перед** setContentView добавим:

```
MapKitFactory.setApiKey("YOUR_API_KEY")
```

где ключ надо взять в кабинете разработчика Yandex.

Далее открываем activity_main.xml и добавляем в код следующее, удалив оттуда textView Hello world:

```
<com.yandex.mapkit.mapview.MapView
    android:id="@+id/mapview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Вернёмся в MainActivity и напишем там:

после объявления класса:

```
var mapView : MapView? = null
var point: Point? = null
val placemarkTapListener = MapObjectTapListener { _, point ->
    Toast.makeText(
        this@MainActivity,
        "Tapped the point (${point.latitude}, ${point.longitude})",
        Toast.LENGTH_SHORT
    ).show()
    true
}
```

в методе onCreate:

```
MapKitFactory.initialize(this)
mapView = findViewById<MapView>(R.id.mapview)
val map = mapView?.mapWindow?.map

val imageProvider = ImageProvider.fromResource(this, R.drawable.placemark_icon)
val placemark = map!!.mapObjects!!.addPlacemark().apply {
    geometry = Point(55.751225, 37.62954)
    setIcon(imageProvider)
}

point = Point(55.751225, 37.62954)
map.move(
    CameraPosition(
        point!!,
        /* zoom = */ 17.0f,
        /* azimuth = */ 150.0f,
        /* tilt = */ 30.0f
    )
)

placemark.addTapListener(placemarkTapListener)
```

и после него добавим ещё 2 метода, согласно гайду:

```
override fun onStart() {  
    super.onStart()  
    MapKitFactory.getInstance().onStart()  
    mapView!!.onStart()  
}  
  
override fun onStop() {  
    mapView!!.onStop()  
    MapKitFactory.getInstance().onStop()  
    super.onStop()  
}
```

Метку `placemark_icon` можно взять прямо из гайда, скачав значок доллара на компьютер и затем перетащив его в папку `drawable`. Метка должна быть маленькой, если она слишком большая, на карте она будет не видна.

В итоге, карта будет работать, а тап по метке будет выводить тост, см. рисунок 32.

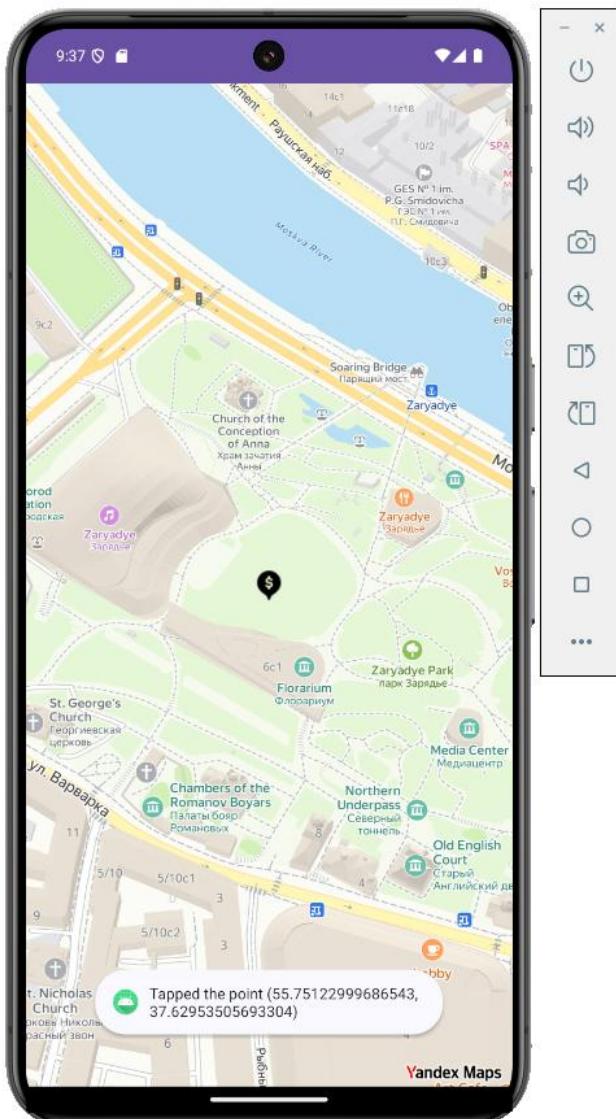


Рисунок 32 – Приложение с Яндекс-картой

OpenStreetMap

Действуем согласно инструкции: <https://github.com/osmdroid/osmdroid>

Добавляем в новый проект в gradle-файл уровня приложения Module: app (например, build.gradle.kts(Module: app)) в секцию dependencies зависимость:

```
implementation("org.osmdroid:osmdroid-android:6.1.20")
```

Синхронизируем проект. Затем переходим либо сюда

[https://github.com/osmdroid/osmdroid/wiki/How-to-use-the-osmdroid-library-\(Java\)](https://github.com/osmdroid/osmdroid/wiki/How-to-use-the-osmdroid-library-(Java)) (Java), либо сюда [https://github.com/osmdroid/osmdroid/wiki/How-to-use-the-osmdroid-library-\(Kotlin\)](https://github.com/osmdroid/osmdroid/wiki/How-to-use-the-osmdroid-library-(Kotlin)) (Kotlin).

Добавляем в файл манифеста разрешения:

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

Добавляем в код activity_main.xml карту:

```
<org.osmdroid.views.MapView
    android:id="@+id/map"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Добавляем в начало класса MainActivity переменные:

```
private val REQUEST_PERMISSIONS_REQUEST_CODE = 1
private lateinit var map : MapView
```

Добавляем в конец метода onCreate строки:

```
map = findViewById<MapView>(R.id.map)
map.setTileSource(TileSourceFactory.MAPNIK)
```

а в начало этого метода перед setContentView строку

```
getInstance().load(this, PreferenceManager.getDefaultSharedPreferences(this))
```

для которой затем в импортах проекта перед классом надо добавить следующий импорт (без этого карта работать не будет):

```
import org.osmdroid.config.Configuration.*
```

Добавляем в класс MainActivity методы onResume, onPause для управления картой в случае, если пользователь отвлечётся от приложения, и onRequestPermissionsResult для динамического разрешения, чтобы карта вообще показывалась:

```
override fun onResume() {
    super.onResume()
    map.onResume()
}

override fun onPause() {
    map.onPause()
    super.onPause()
}

override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    val permissionsToRequest = ArrayList<String>()
    var i = 0
    while (i < grantResults.size) {
        permissionsToRequest.add(permissions[i])
        i++
    }
    if (permissionsToRequest.size > 0) {
        ActivityCompat.requestPermissions(
            this,
            permissionsToRequest.toTypedArray(),
            REQUEST_PERMISSIONS_REQUEST_CODE)
    }
}
```

Подвинем карту в определённую локацию, например, такую же, как было в yandex maps. Для этого добавим в конец метода onCreate строки:

```
val mapController = map.controller
mapController.setZoom(17.0)
val startPoint = GeoPoint(55.751225, 37.62954);
mapController.setCenter(startPoint);
```

Добавим метку на карту в этой точке (у OSM своя метка, иконка не нужна), и по тапу на метку тоже будем выводить тост:

```
var point = GeoPoint(55.751225, 37.62954)
val items = ArrayList<OverlayItem>()
items.add(OverlayItem("I am here", "", point))

var overlay = ItemizedOverlayWithFocus<OverlayItem>(items, object:
    ItemizedIconOverlay.OnItemGestureListener<OverlayItem> {
    override fun onItemSingleTapUp(index:Int, item:OverlayItem):Boolean {
        Toast.makeText(
            this@MainActivity,
```

```

    "Tapped the point (${point.latitude}, ${point.longitude})",
    Toast.LENGTH_SHORT
).show()

return true
}
override fun onItemLongPress(index:Int, item:OverlayItem):Boolean {
    return false
}
}, this)
overlay.setFocusItemsOnTap(true);

map.overlays.add(overlay);

```

Добавим инструменты для зума карты:

```

val dm : DisplayMetrics = this.resources.displayMetrics
val scaleBarOverlay = ScaleBarOverlay(map)
scaleBarOverlay.setCentred(true)
scaleBarOverlay.setScaleBarOffset(dm.widthPixels / 2, 10)
map.overlays.add(scaleBarOverlay)

```

И... готово! см. рисунок 33.

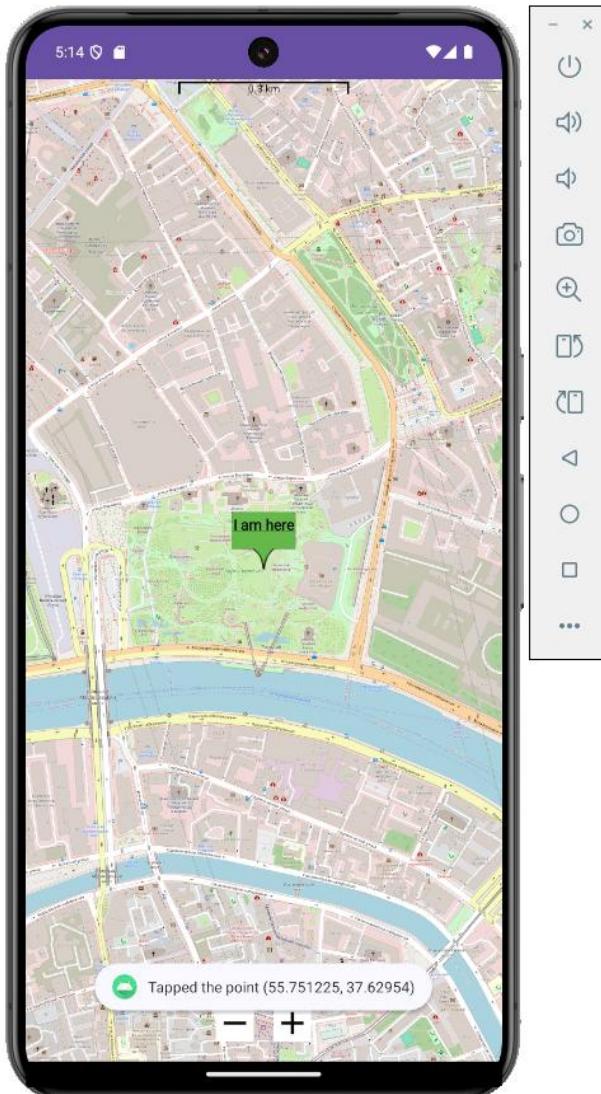


Рисунок 33 – Приложение с OpenStreetMap-картой