

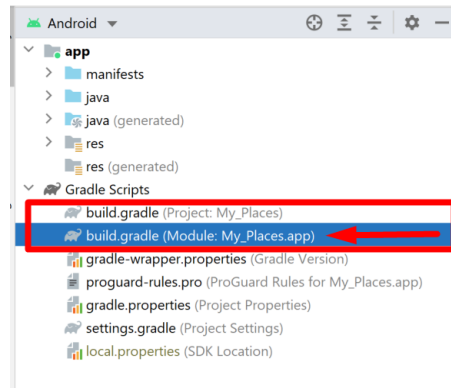
## MOSIS - Laboratorijska vežba 5

Na poslednjoj laboratorijskoj vežbi je omogućen unos omiljenih lokacija i njihovo pregledavanje. Najznačajniji prikaz omiljenih lokacija je onaj koji vezan za kontekst njihove stvarne lokacije na mapi. Za prikaz objekata na mapi, *Google* nudi programerima svoje rešenje za rad sa mapama u vidu posebnog *map* fragmenta koji je moguće dodati u sopstveni *Activity*. Ovim je značajno olakšan rad sa mapama i nije više potrebno biti GIS ekspert da bi se prikaz mape ugradio u aplikacije. U poslednje vreme, Google je promenio poslovnu politiku vezanu za korišćenje mapa u svojim aplikacijama i trenutno zahteva unos podataka o kreditnoj kartici kako bi bilo moguće koristiti API key za Google Maps. Zbog toga je neophodno potražiti alternativne načine prikaza mapa u Android projektima.

Jedna od mogućnosti je korišćenje Open Street Maps mapa. Open Street Maps su besplatne mape koje kreiraju volonteri. Njihovo korišćenje je slobodno, ali usled izostanka komercijalnih uslova kojima je podržan hosting mapa, potrebno je koristiti različite izvore dobijenih od strane Open Street Maps. U ovoj vežbi će akcenat biti na radu sa OSMDroid mapama, kao besplatnim modulom koji omogućuje korišćenje mapa u projektu, čime aplikacija dobija svoju originalnu svrhu. Kao i prethodne laboratorijske vežbe, i ova se oslanja na kod sa prethodnih laboratorijskih vežbi.

Ciljevi ove vežbe su:

1. Uključivanje osmdroid biblioteke u Android projekat.
  2. Priprema AndroidManifest.xml-a i dodavanje preferenci.
  3. Uključivanje osmdroid MapView komponente.
  4. Kreiranje *MapFragment* klase i prikaz mape sa trenutnom lokacijom korisnika.
  5. Dodavanje kontrola za unos i prikaz koordinata omiljene lokacije u *EditFragment*-u i podrška za izbor koordinata omiljene lokacije sa mape.
  6. Prilagođavanje *MapFragment*-a za različita stanja rada (prikaz lokacija na mapi, centriranje prikaza omiljene lokacije na mapi, izbor koordinata omiljene lokacije na mapi).
  7. Dodavanje markera omiljenih lokacija na mapi.
  8. Prikaz određene omiljene lokacije centrirano na mapi.
- 
1. Uključivanje osmdroid biblioteke u projekat.
    - a. Implementacija biblioteke osmdroid se nalazi na adresi:  
<https://github.com/osmdroid/osmdroid>
    - b. Biblioteku je potrebno uključiti u *build.gradle* fajl projekta. Po pravilu, android projekat ima dva *build.gradle* fajla, jedan je na nivou čitavog projekta, dok drugi odgovara aplikaciji.

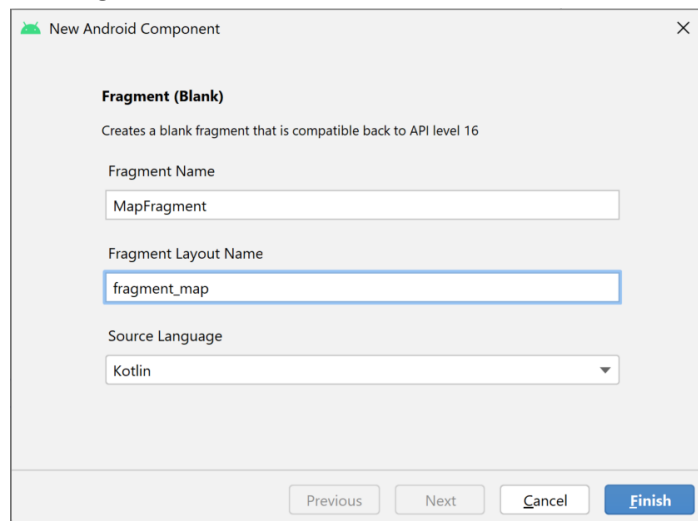


- c. Otvoriti *build.gradle* fajl koji odgovara modulu aplikacije (Module:app) i dodati: *implementation 'org.osmdroid:osmdroid-android:6.1.13'*.

```
dependencies {
    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.5.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
    implementation 'androidx.navigation:navigation-fragment-ktx:2.3.5'
    implementation 'androidx.navigation:navigation-ui-ktx:2.3.5'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'org.osmdroid:osmdroid-android:6.1.13'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

## 2. Dodavanje *MapFragment*-a.

- Potrebno je pokrenuti dodavanje novog *Fragment*-a (*Blank Fragment*) u projekat.
- Podesiti ime *Fragment*-a na *MapFragment* a zatim, nakon dodavanja, obrisati višak dodatog koda.



- c. Dodati potrebne dozvole za pristup aplikacije resursima telefona. Tipično je potrebno dodati dozvolu za pristup finoj lokaciji sa senzora za lokaciju kako bi mogla biti očitana trenutna lokacija, pristup internetu i očitavanju statusa mreže, što omogućuje preuzimanje mape preko online izvora i za pristup fajl sistemu kako bi mapa mogla biti učitana sa fajl sistema i kako bi delovi mape mogli biti keširani.

```
Fragment.kt x AndroidManifest.xml x nav_graph.xml x EditFragment.kt x ListFragment.kt x Vie
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="elfak.mosis.myplaces">
    <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" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Places"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.MyPlaces">
```

- d. Potrebno je u *layout* fragmenta pravilno dodati *MapView* na sledeći način:

```
ity_main.xml x fragment_map.xml x content_main.xml x ListFragment.kt x ListView.
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapFragment">
    <org.osmdroid.views.MapView android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tilesource="Mapnik"/>
</FrameLayout>
```

- e. Za inicijalizaciju mape je potrebno učitati standardne deljene preference aplikacije kako bi aplikacija mogla da kešira delove mape na samom uređaju. U novijim verzijama Androida je za to potrebno dodati novu biblioteku u Gradle.

```
dependencies {
    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.5.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
    implementation 'androidx.navigation:navigation-fragment-ktx:2.3.5'
    implementation 'androidx.navigation:navigation-ui-ktx:2.3.5'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'androidx.preference:preference-ktx:1.1.1'
    implementation 'org.osmdroid:osmdroid-android:6.1.13'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

- f. Dodati odgovarajući kod u *MapFragment.kt* koji inicijalizuje mapu.

## Android

```
vent_map.xml | MapFragment.kt | ListFragment.kt | EditFragment.kt | ViewFragment.kt | MainActivity.kt | fra
package elfak.mosis.mylplaces

import android.content.Context
import android.os.Bundle
import androidx.preference.PreferenceManager
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import org.osmdroid.config.Configuration
import org.osmdroid.views.MapView

class MapFragment : Fragment() {
    lateinit var map: MapView

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_map, container, attachToRoot: false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        var ctx: Context? = getActivity()?.getApplicationContext()
        Configuration.getInstance().load(ctx, PreferenceManager.getDefaultSharedPreferences((ctx)))
        map = requireView().findViewById<MapView>(R.id.map)
    }

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

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

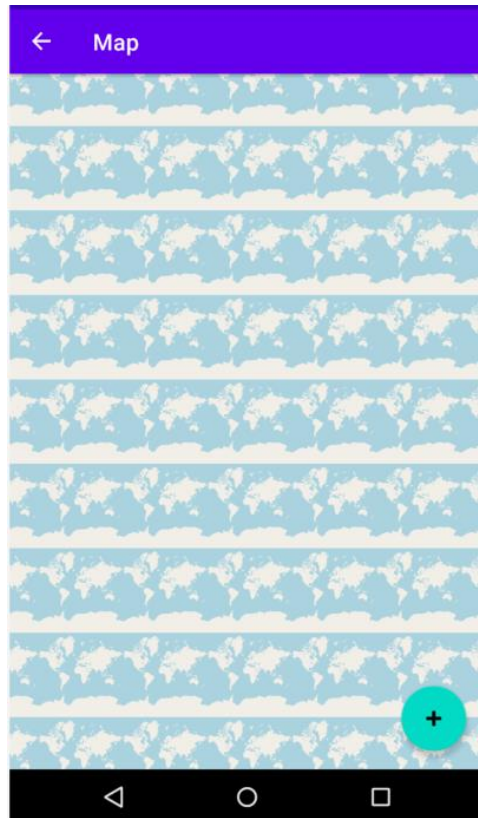
- g. U cilju testiranja dodatog *MapFragment*-a potrebno je dodati odgovarajuću navigaciju u *NavGraph*.

```
<action
    android:id="@+id/action_HomeFragment_to_EditFragment"
    app:destination="@id/EditFragment" />
<action
    android:id="@+id/action_HomeFragment_to_MapFragment"
    app:destination="@id/MapFragment" />
</fragment>
<fragment
    android:id="@+id/ListFragment"
    android:name="elfak.mosis.mypplaces.ListFragment"
    android:label="My Places"
    tools:layout="@layout/fragment_list">
    <action
        android:id="@+id/action_ListFragment_to_EditFragment"
        app:destination="@id/EditFragment" />
    <action
        android:id="@+id/action_ListFragment_to_ViewFragment"
        app:destination="@id/ViewFragment" />
    <action
        android:id="@+id/action_ListFragment_to_MapFragment"
        app:destination="@id/MapFragment" />
</fragment>
<fragment
    android:id="@+id/EditFragment"
    android:name="elfak.mosis.mypplaces.EditFragment"
    android:label="Edit My Place"
    tools:layout="@layout/fragment_edit">
</fragment>
<fragment
    android:id="@+id/ViewFragment"
    android:name="elfak.mosis.mypplaces.ViewFragment"
    android:label="View My Place"
    tools:layout="@layout/fragment_view">
</fragment>
<fragment
    android:id="@+id/MapFragment"
    android:name="elfak.mosis.mypplaces.MapFragment"
    android:label="@string/map_fragment_label"
    tools:layout="@layout/fragment_map">
</fragment>
</navigation>
```

h. Dodati navigaciju na fragment za prikaz mape u *MainActivity.kt*.

```
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    when (item.itemId) {
        R.id.action_show_map -> findNavController(R.id.nav_host_fragment_content_main).navigate(R.id.action_HomeFragment_to_MapFragment)
        R.id.action_about -> {
            val i: Intent = Intent(packageContext, About::class.java)
            startActivity(i)
        }
    }
    return super.onOptionsItemSelected(item)
}
```

i. Pokrenuti aplikaciju i otvoriti mapu



- j. Dodati u *NavGraph* akciju koja sa mape, klikom na *fab* dugme otvara *EditFragment* kao i kod koji vrši navigaciju.

```
<fragment
    android:id="@+id/MapFragment"
    android:name="elfak.mosis.mypplaces.MapFragment"
    android:label="@string/map_fragment_label"
    tools:layout="@layout/fragment_map">
    <action
        android:id="@+id/action_MapFragment_to_EditFragment"
        app:destination="@id/EditFragment" />
</fragment>

binding.fab.setOnClickListener { view ->
    if(navController.currentDestination?.id == R.id.HomeFragment)
        navController.navigate(R.id.action_HomeFragment_to_EditFragment)
    else if(navController.currentDestination?.id == R.id.ListFragment)
        navController.navigate(R.id.action_ListFragment_to_EditFragment)
    else if(navController.currentDestination?.id == R.id.MapFragment)
        navController.navigate(R.id.action_MapFragment_to_EditFragment)
}
```

- k. Dodati i ostale vidove navigacije na mapu sa ostalih relevantnih fragmenata.

## Laboratorijske vežbe - Mobilni sistemi i servisi

```
class MainActivity : AppCompatActivity() {

    private lateinit var appBarConfiguration: AppBarConfiguration
    private lateinit var binding: ActivityMainBinding
    private lateinit var navController: NavController

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setSupportActionBar(binding.toolbar)

        navController = findNavController(R.id.nav_host_fragment_content_main)
        appBarConfiguration = AppBarConfiguration(navController.graph)
        setupActionBarWithNavController(navController, appBarConfiguration)

        navController.addOnDestinationChangedListener { controller, destination, arguments ->
            if (destination.id == R.id.EditFragment || destination.id == R.id.ViewFragment)
                binding.fab.hide()
            else
                binding.fab.show()
        }

        binding.fab.setOnClickListener { view ->
            if (navController.currentDestination?.id == R.id.HomeFragment)
                navController.navigate(R.id.action_HomeFragment_to_EditFragment)
            else if (navController.currentDestination?.id == R.id.ListFragment)
                navController.navigate(R.id.action_ListFragment_to_EditFragment)
            else if (navController.currentDestination?.id == R.id.MapFragment)
                navController.navigate(R.id.action_MapFragment_to_EditFragment)
        }
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        when (item.itemId) {
            R.id.action_show_map -> {
                if (navController.currentDestination?.id == R.id.HomeFragment)
                    navController.navigate(R.id.action_HomeFragment_to_MapFragment)
                else if (navController.currentDestination?.id == R.id.ListFragment)
                    navController.navigate(R.id.action_ListFragment_to_MapFragment)
            }
            R.id.action_about -> {
                val i: Intent = Intent(packageContext, this, About::class.java)
                startActivity(i)
            }
        }
    }
}
```

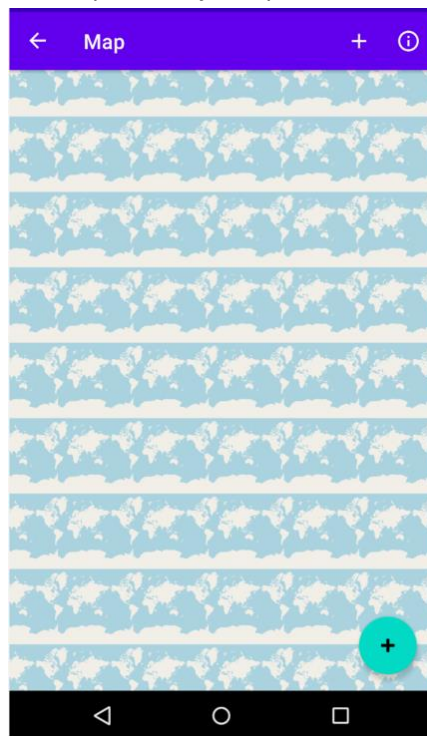
- I. Na MapFragmentu je potrebno da postoje i dugmad toolbar-a.



## Android

```
class MapFragment : Fragment() {
    lateinit var map: MapView
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setHasOptionsMenu(true)
    }
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_map, container, attachToRoot: false)
    }
    override fun onCreateOptionsMenu(menu: Menu, inflater: MenuInflater) {
        inflater.inflate(R.menu.menu_main, menu)
    }
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        var ctx: Context? = getActivity()?.getApplicationContext()
        Configuration.getInstance().load(ctx, PreferenceManager.getDefaultSharedPreferences((ctx)))
        map = requireView().findViewById<MapView>(R.id.map)
    }
    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return when (item.itemId) {
            R.id.action_new_place -> {
                this.findNavController().navigate(R.id.action_MapFragment_to_EditFragment)
                true
            }
            else -> super.onOptionsItemSelected(item)
        }
    }
    override fun onPrepareOptionsMenu(menu: Menu){
        super.onPrepareOptionsMenu(menu)
        var item = menu.findItem(R.id.action_my_places_list)
        item.isVisible = false;
        item = menu.findItem(R.id.action_show_map)
        item.isVisible = false;
    }
}
```

m. Nakon pokretanja mape na ekranu se prikazuje mapa sa *Toolbar*-om.





- n. Dodati kontrolu mape i postaviti je tako da ukazuje na grad Niš (koordinate latituda: 43.3209, longituda: 21.8958)

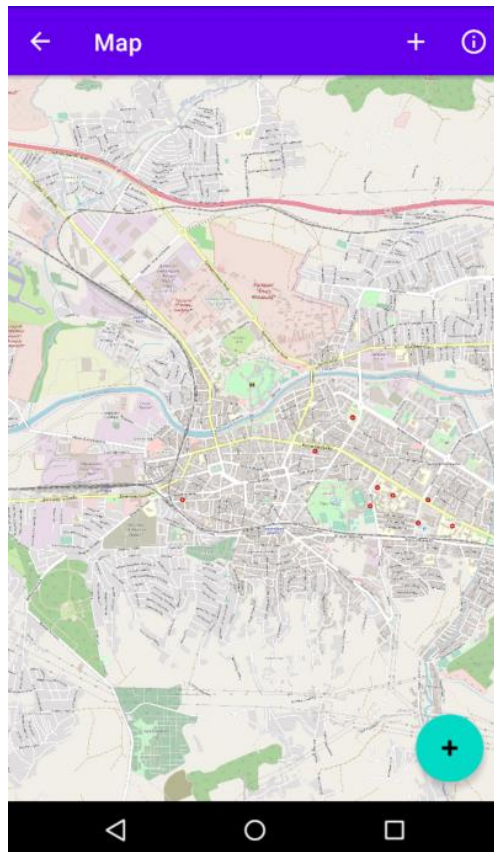
```
igment_home.xml | MapFragment.kt | land/fragment_home.xml | activity_main.xml | AndroidManifest.xml | nav_
override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_map, container, attachToRoot: false)
}

override fun onCreateOptionsMenu(menu: Menu, inflater: MenuInflater) {
    inflater.inflate(R.menu.menu_main, menu)
}

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    var ctx: Context? = getActivity()?.getApplicationContext()
    Configuration.getInstance().load(ctx, PreferenceManager.getDefaultSharedPreferences((ctx)))
    map = requireView().findViewById<MapView>(R.id.map)
    map.setMultiTouchControls(true)
    map.controller.setZoom(15.0)
    val startPoint = GeoPoint(aLatitude: 43.3209, aLongitude: 21.8958)
    map.controller.setCenter(startPoint)
}

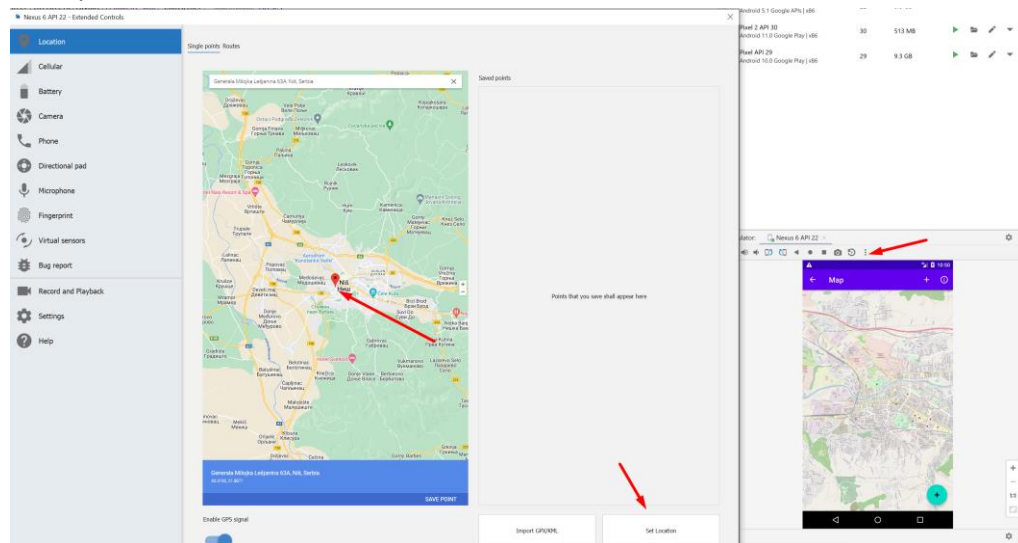
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.action_new_place -> {
            this.findNavController().navigate(R.id.action_MapFragment_to_EditFragment)
            true
        }
        else -> super.onOptionsItemSelected(item)
    }
}
```

- o. Pokrenuti mapu ponovo!

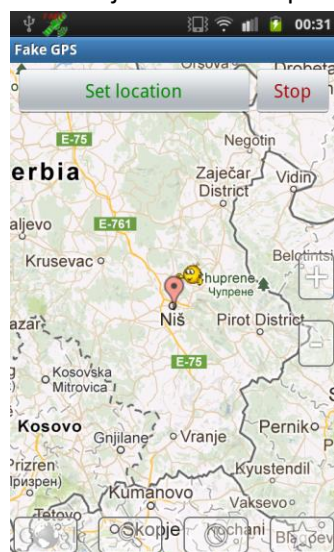


### 3. Prikaz lokacije korisnika na mapi.

- a. Implementirana mapa prikazuje samo fiksnu lokaciju koja je prosleđena u implementiranoj metodi.
- b. U cilju dobijanja lokacije u toku testiranja aplikacije, potrebno je koristiti nekakav izvor trenutne lokacije koji ne može biti GPS, jer se prilikom testiranja obično korisnik nalazi unutar objekta. U tom slučaju je, ukoliko se koristi emulator, potrebno lokaciju postaviti u Extended Controls aplikaciji. Aplikaciju je potrebno pokrenuti iz menija emulatora. Nakon što je aplikacija pokrenuta, moguće je uneti željenu lokaciju klikom na mapu. Klikom na dugme *Set Location* polja i kliknuti na *Send*. Ove koordinate će automatski biti dostupne u emulatoru.



- c. Ukoliko je za testiranje potrebno koristiti uređaj, lokaciju je moguće dobiti ili od nekog drugog provajdera (poput WiFi ili GSM) ili koristiti aplikaciju koja emulira lokacijski provajder (preporuka *FakeGPS* aplikacija).
- d. Korišćenjem *FakeGPS* aplikacije lokaciju je moguće postaviti izborom lokacije na mapi.



- e. Prepraviti kod aplikacije tako da postavlja na mapi trenutnu korisnikovu lokaciju. Da bi bilo moguće postaviti tu lokaciju, neophodno je od korisnika zahtevati dopuštanje privilegije pribavljanja lokacije. Ovakva provera za “osetljive” privilegije je eksplicitno potrebna za verzije Android-a počev od verzije 6 (API 23). *Permission* je već dodat u *AndroidManifest* fajl, ali je neophodno i da korisnik potvrdi saglasnost. Kod koji ovo omogućava je:

```

override fun onCreateView(view: View, savedInstanceState: Bundle?) {
    super.onCreateView(view, savedInstanceState)
    var ctx: Context? = getActivity()?.getApplicationContext()
    Configuration.getInstance().load(ctx, PreferenceManager.getDefaultSharedPreferences((ctx)))
    map = requireView().findViewById<MapView>(R.id.map)
    map.setMultiTouchControls(true)
    if (ActivityCompat.checkSelfPermission(requireActivity(), android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(requireActivity(), android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        requestPermissionsLauncher.launch(
            android.Manifest.permission.ACCESS_FINE_LOCATION
        )
    } else {
        setMyLocationOverlay()
    }
}

map.controller.setZoom(15.0)
val startPoint = GeoPoint(43.3209, 21.8958)
map.controller.setCenter(startPoint)

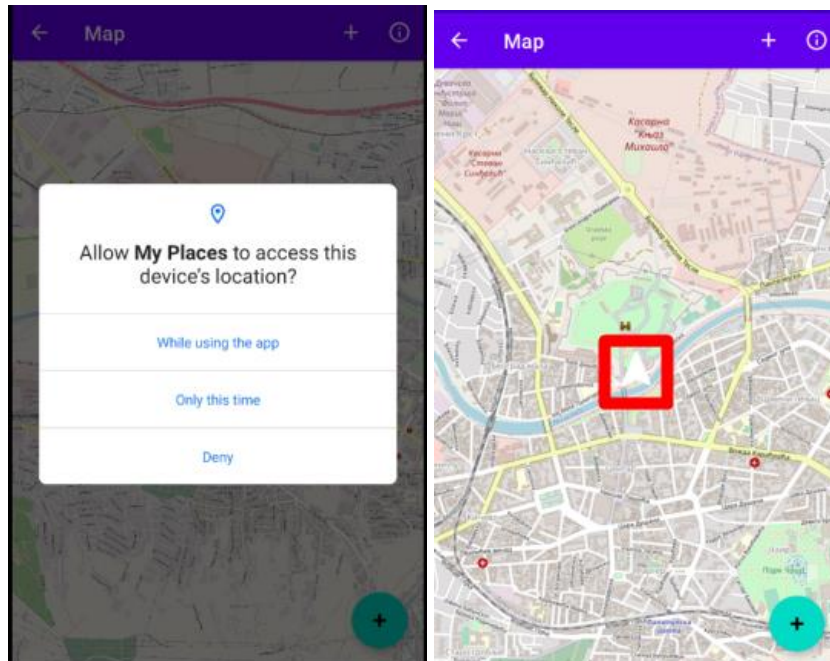
private fun setMyLocationOverlay() {
    var myLocationOverlay = MyLocationNewOverlay(GpsMyLocationProvider(activity), map)
    myLocationOverlay.enableMyLocation()
    map.overlays.add(myLocationOverlay)
}

private val requestPermissionsLauncher =
    registerForActivityResult(
        ActivityResultContracts.RequestPermission()
    ) { isGranted: Boolean ->
        if (isGranted) {
            setMyLocationOverlay()
        }
    }

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.action_new_place -> {
            this.findNavController().navigate(R.id.action_MapFragment_to_EditFragment)
            true
        }
        else -> super.onOptionsItemSelected(item)
    }
}

```

- f. Isprobati aplikaciju. Nakon pokretanja aplikacije, potrebno je odobriti aplikaciji korišćenje pristupa finoj lokaciji.



4. Dodavanje podrške za prikaz lokacije (longitude i latitude)
- Dodati u konstruktor *DataClass*-a *MyPlace* dva string polja *longitude* i *latitude*.

- b. Potrebno je izmeniti izgled *EditFragment*-a tako da je moguće uneti longitudu i latitudu. Dodati i dugme koje će biti korišćeno za očitavanje koordinata sa mape.
- i. Elemente je potrebno dodati između *EditText* widget-a i dugmića za snimanje/otkazivanje editovanja.



```

android:id="@+id/editmyplace_desc_edit"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="1"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="bottom"
    android:orientation="horizontal">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="bottom"
        android:orientation="vertical"
        android:layout_weight="1">
        <TextView
            android:id="@+id/textView4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/editmyplace_longitude_label" />
        <EditText
            android:id="@+id/editmyplace_longitude_edit"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
        <TextView
            android:id="@+id/textView5"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/editmyplace_latitude_label" />
        <EditText
            android:id="@+id/editmyplace_latitude_edit"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            />
    </LinearLayout>
    <Button
        android:id="@+id/editmyplace_location_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:layout_gravity="center_vertical"
        android:text="@string/editmyplace_button_set_coordinates"/>
    </LinearLayout>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    . . . . .

```

Name:

Description:

Longitude:

Latitude:

SET

ADD CANCEL

c. Potrebno je povezati *EditMyPlaceActivity* tako da se vrši ažuriranje podataka o lokaciji.

```
nt.kt x MyPlace.kt x fragment_edit.xml x EditFragment.kt x strings.xml x AndroidManifest.xml x ListFragment.kt x

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    val editName: EditText = requireView().findViewById<EditText>(R.id.editmyplace_name_edit)
    val editDesc: EditText = requireView().findViewById<EditText>(R.id.editmyplace_desc_edit)
    val editLongitude: EditText = requireView().findViewById<EditText>(R.id.editmyplace_longitude_edit)
    val editLatitude: EditText = requireView().findViewById<EditText>(R.id.editmyplace_latitude_edit)
    if(myPlacesViewModel.selected!=null){
        editName.setText(myPlacesViewModel.selected?.name)
        editDesc.setText(myPlacesViewModel.selected?.description)
    }
    val addButton: Button = requireView().findViewById<Button>(R.id.editmyplace_finished_button)
    addButton.isEnabled = false
    if(myPlacesViewModel.selected!=null)
        addButton.setText("Save")
    editName.addTextChangedListener(object : TextWatcher {
        override fun afterTextChanged(s: Editable?) {
            addButton.isEnabled=(editName.text.length>0)
        }
        override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {
        }
        override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
        }
    })
    addButton.setOnClickListener { it: View!
        val name: String = editName.text.toString()
        val desc: String = editDesc.text.toString()
        val longitude:String = editLongitude.text.toString()
        val latitude:String = editLatitude.text.toString()
        if(myPlacesViewModel.selected!=null){
            myPlacesViewModel.selected?.name = name
            myPlacesViewModel.selected?.description = desc
            myPlacesViewModel.selected?.longitude = longitude
            myPlacesViewModel.selected?.latitude = latitude
        }
        else
            myPlacesViewModel.addPlace( MyPlace(name, desc, longitude, latitude))
        findNavController().popBackStack()
    }
    val cancelButton: Button = requireView().findViewById<Button>(R.id.editmyplace_cancel_button)
    cancelButton.setOnClickListener { it: View!
        findNavController().popBackStack()
    }
}
```

- d. Dodati mogućnost brisanja omiljene lokacije u kontekstni meni *MyPlacesList.java* klase. Potrebno je obraditi stavku menija u metodi *onCreateContextMenu()*.

```

override fun onContextItemSelected(item: MenuItem): Boolean {
    val info = item.menuInfo as AdapterContextMenuInfo
    if (item.itemId == 1) {
        myPlacesViewModel.selected = myPlacesViewModel.myPlacesList[info.position]
        this.findNavController().navigate(R.id.action_ListFragment_to_ViewFragment)
    } else if (item.itemId == 2) {
        myPlacesViewModel.selected = myPlacesViewModel.myPlacesList[info.position]
        this.findNavController().navigate(R.id.action_ListFragment_to_EditFragment)
    } else if (item.itemId == 3) {
        myPlacesViewModel.myPlacesList.removeAt(info.position)
        val myPlacesList: ListView = requireView().findViewById<ListView>(R.id.my_places_list)
        myPlacesList.adapter = this@ListFragment.context?.let { ArrayAdapter<MyPlace>(it, android.R.layout.simple_list_item_1, myPlacesViewModel.myPlacesList) }
    }
    return super.onContextItemSelected(item)
}

```

5. Za domaći: dodati prikaz koordinata u *ViewMyPlaceActivity.java*.
6. Dodavanje podrške za izbor koordinata omiljene lokacije sa mape u *EditFragment.kt*.
- a. Kako bi bila podržana komunikacija između dva fragmenta tako da se izabere lokacija na mapi, potrebno je dodati novi *ViewModel* kojim će komunicirati isključivo *EditFragment* i *MapFragment*, po imenu *LocationViewModel* u folder. U ovom *ViewModel*-u će biti iskorišćen koncept *LiveData* kod koga se komponente korisničkog interfejsa mogu prijaviti za dobijanje „obaveštenja“ o promenama vrednosti podataka koji se nalaze u *ViewModel*-u. U te svrhe se koriste *MutableLiveData* i *LiveData* tipovi objekata. Napraviti *LocationViewModel* kao na slici.

```

package elfak.mosis.mypplaces.model

import androidx.lifecycle.LiveData
import androidx.lifecycle.MutableLiveData
import androidx.lifecycle.ViewModel

class LocationViewModel: ViewModel() {
    private val _longitude = MutableLiveData<String>() { value: "" }
    val longitude: LiveData<String>
        get() = _longitude
    private val _latitude = MutableLiveData<String>() { value: "" }
    val latitude: LiveData<String>
        get() = _latitude
    var setLocation: Boolean = false
    fun setLocation(lon:String, lat:String){
        _longitude.value = lon
        _latitude.value = lat
        setLocation = false
    }
}

```

- b. U *MapFragment* i *EditFragment* dodati objekte za deljenje *ViewModel*-e.
- c. Dodati akciju u *NavGraph* koja vrši tranziciju između *EditFragment*-a i *MapFragment*-a.

```

<fragment
    android:id="@+id/EditFragment"
    android:name="elfak.mosis.mypplaces.EditFragment"
    android:label="Edit My Place"
    tools:layout="@layout/fragment_edit">
    <action
        android:id="@+id/action_EditFragment_to_MapFragment"
        app:destination="@id/MapFragment" />
</fragment>

```

- d. Dodati obzervere na odgovarajuće kontrole koji će da prate promene u *ViewModel*-u.



```

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
    //Toast.makeText(view.context, locationViewModel.longitude, Toast.LENGTH_SHORT).show()
    val editName: EditText = requireView().findViewById<EditText>(R.id.editmyplace_name_edit)
    val editDesc: EditText = requireView().findViewById<EditText>(R.id.editmyplace_desc_edit)
    val editLongitude: EditText = requireView().findViewById<EditText>(R.id.editmyplace_longitude_edit)
    val lonObserver = Observer<String> { newValue ->
        editLongitude.setText(newValue.toString())
    }
    locationViewModel.longitude.observe(viewLifecycleOwner, lonObserver)
    val editLatitude: EditText = requireView().findViewById<EditText>(R.id.editmyplace_latitude_edit)
    val latObserver = Observer<String> { newValue ->
        editLatitude.setText(newValue.toString())
    }
    locationViewModel.latitude.observe(viewLifecycleOwner, latObserver)
    if(myPlacesViewModel.selected!=null){
        editName.setText(myPlacesViewModel.selected?.name)
        editDesc.setText(myPlacesViewModel.selected?.description)
        editLongitude.setText(myPlacesViewModel.selected?.longitude)
        editLatitude.setText(myPlacesViewModel.selected?.latitude)
    }
}

```

- e. Dodati kod koji dugmetu Set daje funkcionalnost prelaska na MapFragment.

```

val cancelButton: Button = requireView().findViewById<Button>(R.id.editmyplace_cancel_button)
cancelButton.setOnClickListener {
    locationViewModel.setLocation(lon = "", lat = "")
    findNavController().popBackStack()
}

val setButton: Button = requireView().findViewById<Button>(R.id.editmyplace_location_button)
setButton.setOnClickListener {
    locationViewModel.setLocation = true;
    findNavController().navigate(R.id.action_EditFragment_to_MapFragment)
}

override fun onDestroyView() {
    //super.onDestroyView()
}

```

- f. U MapFragment-u dodati podršku za klik na mapu. Obratiti pažnju da poziv `locationViewModel.setLocation` metod zapravo ažurira lokaciju u `LocationViewModel`-u

```

setMyLocationOverlay()
setOnMapClickOverlay()
}

map.controller.setZoom(15.0)
val startPoint = GeoPoint( aLatitude: 43.3209, aLongitude: 21.8958)
map.controller.setCenter(startPoint)
}

private fun setMyLocationOverlay() {
    var myLocationOverlay = MyLocationNewOverlay(GpsMyLocationProvider(activity))
    myLocationOverlay.enableMyLocation()
    map.overlays.add(myLocationOverlay)
}

private val requestPermissionLauncher =
    registerForActivityResult(
        ActivityResultContracts.RequestPermission()
    ) { isGranted: Boolean ->
        if (isGranted) {
            setMyLocationOverlay()
            setOnMapClickOverlay()
        }
    }

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.action_new_place -> {
            this.findNavController().navigate(R.id.action_MapFragment_to_EditFrc
                true
            }
        }
        else -> super.onOptionsItemSelected(item)
    }
}

private fun setOnMapClickOverlay(){
    var receive = object:MapEventsReceiver{
        override fun singleTapConfirmedHelper(p:GeoPoint):Boolean {
            var lon = p.longitude.toString()
            var lat = p.latitude.toString()
            locationViewModel.setLocation(lon,lat)
            findNavController().popBackStack()
            return true
        }
        override fun longPressHelper(p: GeoPoint ):Boolean {
            return false
        }
    }
    var overlayEvents = MapEventsOverlay(receive)
    map.overlays.add(overlayEvents)
}

```

- g. U *EditFragment*-u obratiti pažnju da se lokacija briše nakon povratka iz *EditFragment*-a. Dodati i snimanje longitude i latitude u objekat.

```

    })
    addButton.setOnClickListener { it: View?
        val name: String = editName.text.toString()
        val desc: String = editDesc.text.toString()
        val longitude: String = editLongitude.text.toString()
        val latitude: String = editLatitude.text.toString()
        if (myPlacesViewModel.selected != null) {
            myPlacesViewModel.selected?.name = name
            myPlacesViewModel.selected?.description = desc
            myPlacesViewModel.selected?.longitude = longitude
            myPlacesViewModel.selected?.latitude = latitude
        }
        else {
            myPlacesViewModel.addPlace(MyPlace(name, desc, longitude, latitude))
            myPlacesViewModel.selected = null
            locationViewModel.setLocation(lon: "", lat: "")
            findNavController().popBackStack()
        }
    }

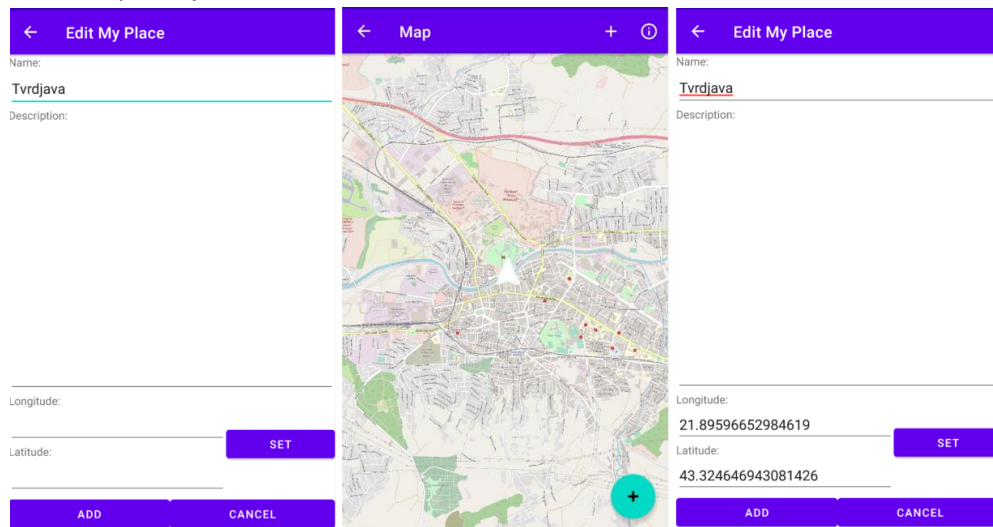
    val cancelButton: Button = requireView().findViewById<Button>(R.id.editmyplace_cancel_button)
    cancelButton.setOnClickListener { it: View?
        myPlacesViewModel.selected = null
        locationViewModel.setLocation(lon: "", lat: "")
        findNavController().popBackStack()
    }

    val setButton: Button = requireView().findViewById<Button>(R.id.editmyplace_location_button)
    setButton.setOnClickListener { it: View?
        locationViewModel.setLocation = true;
        findNavController().navigate(R.id.action_EditFragment_to_MapFragment)
    }
}

override fun onDestroyView() {
    super.onDestroyView()
}

```

- h. Probati aplikaciju.



7. Prilagođavanje *MapFragment*-a za različita stanja rada (prikaz lokacija na mapi, centriranje prikaza omiljene lokacije na mapi, izbor koordinata omiljene lokacije na mapi).
- a. Trenutno je moguće izabrati koordinate sa mape ali nije moguće pomerati mapu (*pan*) i uvećavati/umanjivati mapu (*zoom*). Potrebno je implementirati u *MapFragment*-u različita stanja. Stanja će biti određena na osnovu stanja *ViewModel*-a koji se koriste. Ukoliko je u *LocationViewModel*-u postavljena promenljiva *setLocation* na *true*, *MapFragment* treba da ima ulogu selekcije lokacije. Ukoliko je postavljen na *false* a postoji selektovan objekat u *MyPlacesViewModel*-u, tada je potrebno taj objekat centrirati na

mapi. Ukoliko ni jedno ni drugo nije postavljeno, onda se mapa centrira na gradu Nišu i slobodno se može panovati.

```

override fun onCreateView(view: View, savedInstanceState: Bundle?) {
    super.onCreateView(view, savedInstanceState)
    var ctx: Context? = getActivity()?.getApplicationContext()
    Configuration.getInstance().load(ctx, PreferenceManager.getDefaultSharedPreferences((ctx)))
    map = requireView().findViewById<MapView>(R.id.map)
    map.setMultiTouchControls(true)
    if (ActivityCompat.checkSelfPermission(requireActivity(), android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(requireActivity(), android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        requestPermissionLauncher.launch(
            android.Manifest.permission.ACCESS_FINE_LOCATION
        )
    } else {
        setupMap()
    }
}

private fun setupMap() {
    var startPoint: GeoPoint = GeoPoint(43.3289, 21.8958)
    map.controller.setZoom(15.0)
    if (locationViewModel.selectedLocation != null) {
        setOnMapClickOverlay()
    } else {
        if (myPlacesViewModel.selected != null) {
            startPoint = GeoPoint(myPlacesViewModel.selected!!.latitude.toDouble(), myPlacesViewModel.selected!!.longitude.toDouble())
        } else {
            setMyLocationOverlay()
        }
    }
    map.controller.animateTo(startPoint)
}

```

- b. Dodati logiku koja kontroliše da li je mapa centrirana na objekat u zavisnosti da li je on selektovan.

```

myPlacesList.setOnCreateContextMenuListener(object : View.OnCreateContextMenuListener {
    override fun onCreateContextMenu(menu: ContextMenu, v: View?, menuInfo: ContextMenuInfo) {
        val info = menuInfo as AdapterContextMenuInfo
        val myPlace: MyPlace = myPlacesViewModel.myPlacesList[info.position]
        menu.setHeaderTitle(myPlace.name)
        menu.add(0, 1, 1, "View place")
        menu.add(0, 2, 2, "Edit place")
        menu.add(0, 3, 3, "Delete place")
        menu.add(0, 4, 4, "Show on map")
    }
})

override fun onContextItemSelected(item: MenuItem): Boolean {
    val info = item.menuInfo as AdapterContextMenuInfo
    if (item.itemId == 1) {
        myPlacesViewModel.selected = myPlacesViewModel.myPlacesList[info.position]
        this.findNavController().navigate(R.id.action_ListFragment_to_ViewFragment)
    } else if (item.itemId == 2) {
        myPlacesViewModel.selected = myPlacesViewModel.myPlacesList[info.position]
        this.findNavController().navigate(R.id.action_ListFragment_to_EditFragment)
    } else if (item.itemId == 3) {
        myPlacesViewModel.myPlacesList.removeAt(info.position)
        val myPlacesList: ListView = requireView().findViewById<ListView>(R.id.my_places_list)
        myPlacesList.adapter = this@ListFragment.context?.let { ArrayAdapter<MyPlace>(it, android.
    } else if (item.itemId == 4) {
        myPlacesViewModel.selected = myPlacesViewModel.myPlacesList[info.position]
        this.findNavController().navigate(R.id.action_ListFragment_to_MapFragment)
    }
    return super.onContextItemSelected(item)
}

```