

**Министерство науки и высшего образования Российской Федерации**  
**Федеральное государственное автономное образовательное**  
**учреждение высшего образования**  
**Национальный исследовательский**  
**Томский политехнический университет**

Инженерная школа информационных технологий и робототехники  
Отделение информационных технологий

Отчет по лабораторной работе №14 по дисциплине

**«Язык Kotlin и основы разработки»**

Фрагменты

Выполнил:

Студент группы 1A22



О.К. Кравцов

Проверил:

Ст. преп. ОИТ ИШИТР

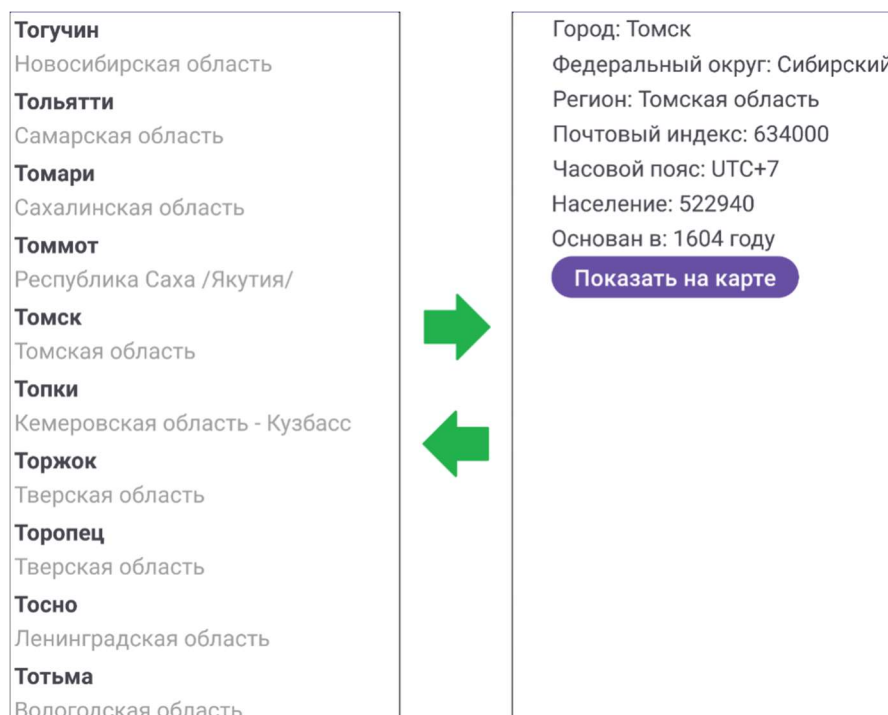


В.А. Дорофеев

Томск 2025

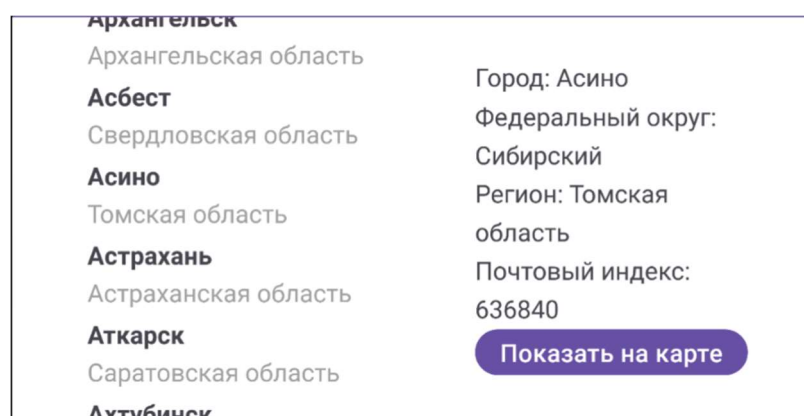
## Задание

Создайте приложение, отображающее информацию о выбранном городе, а также показывающее его на карте. Да-да, как в прошлой лабораторной работе – можно даже код оттуда взять за основу! Но интерфейс на сей раз должен быть реализован через фрагменты. Если устройство находится в портретной ориентации, то весь экран занимает фрагмент со списком городов, а при нажатии на город – фрагмент со списком заменяется на фрагмент с информацией о городе:



При нажатии кнопки «Назад» происходит возврат на список городов, причём позиция прокрутки списка должна сохраняться.

При альбомной ориентации устройства на экране присутствуют сразу два фрагмента: слева со списком городов, а справа – с информацией о городе. При нажатии на название города не происходит никакой замены фрагментов, просто в правом фрагменте отображается информация о новом выбранном городе:



В данной лабораторной работе дополнительным затруднение может стать требование сохранения позиции прокрутки при возврате фрагмента со списком. Одним из вариантов решения этого затруднения является сохранение состояния списка – точнее даже не самого списка, а его менеджера разметки:

```
state = (list.layoutManager as LinearLayoutManager).onSaveInstanceState()
```

В приведённом коде у менеджера разметки вызывается метод `onSaveInstanceState`. Это немного обескураживает, ведь это функция обратного вызова, и обычно мы сами никогда её не вызываем, её всегда вызывает система! Зато после вызова этого метода (например, в событии `onPause`) в переменной `state` окажется внутреннее состояние менеджера разметки. А затем, когда фрагмент со списком снова создаст свою разметку перед появлением на экране (в методе `onCreateView`) можно восстановить состояние таким образом:

```
if (state != null)
    (list.layoutManager as LinearLayoutManager).onRestoreInstanceState(state)
```

.

## Ход работы

1. Создан проект Lab14 на основе Empty Views Activity.
2. Настроены ресурсы приложения:

– colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="white">#FFFFFF</color>
    <color name="primary_color">#2196F3</color>
    <color name="accent_color">#FF4081</color>
    <color name="card_background">#FFFFFF</color>
    <color name="text_primary">#212121</color>
    <color name="text_secondary">#757575</color>
    <color name="divider_color">#E0E0E0</color>
</resources>
```

– dims.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="padding_medium">16dp</dimen>
    <dimen name="padding_small">8dp</dimen>
    <dimen name="padding_very_small">4dp</dimen>
    <dimen name="margin_bottom_large">24dp</dimen>
    <dimen name="margin_bottom_medium">16dp</dimen>
    <dimen name="margin_bottom_small">8dp</dimen>
    <dimen name="text_size_large">18sp</dimen>
    <dimen name="text_size_medium">16sp</dimen>
    <dimen name="text_size_small">14sp</dimen>
    <dimen name="item_corner_radius">8dp</dimen>
    <dimen name="item_elevation">2dp</dimen>
</resources>
```

– strings.xml

```
<resources>
    <string name="app_name">Lab14</string>
    <string name="show_on_map">Показать на карте</string>
    <string name="city">Город: %s</string>
    <string name="federal_district">Федеральный округ: %s</string>
    <string name="region">Регион: %s</string>
    <string name="postal_code">Почтовый индекс: %s</string>
    <string name="timezone">Часовой пояс: %s</string>
    <string name="population">Население: %s</string>
    <string name="founded">Основан в: %s</string>
    <string name="unknown">Неизвестно</string>
</resources>
```

## – themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.Lab14" parent="Theme.Material3.DayNight.NoActionBar">
        <item name="colorPrimary">@color/primary_color</item>
        <item name="colorPrimaryVariant">@color/primary_color</item>
        <item name="colorOnPrimary">@color/white</item>
        <item name="colorSecondary">@color/accent_color</item>
        <item name="colorSecondaryVariant">@color/accent_color</item>
        <item name="colorOnSecondary">@color/white</item>
        <item name="android:statusBarColor">?attr/colorPrimaryVariant</item>
        <item name="android:windowDrawsSystemBarBackgrounds">true</item>
        <item name="android:windowLightStatusBar">false</item>
    </style>

    <style name="Theme.Lab14" parent="Base.Theme.Lab14" />
</resources>
```

## 3. Реализована модель данных:

### – City.kt

```
package ru.olegkravtsov.lab14

import android.os.Parcelable
import kotlinx.parcelize.Parcelize

@Parcelize
data class City(
    val title: String,
    val region: String,
    val district: String,
    val postalCode: String,
    val timezone: String,
    val population: String,
    val founded: String,
    val lat: Float,
    val lon: Float
) : Parcelable
```

#### 4. Создан синглтон для работы с данными:

– Common.kt

```
package ru.olegkravtsov.lab14

import android.content.Context
import android.util.Log

object Common {
    val cities = mutableListOf<City>()
    var selectedCity: City? = null

    fun initCities(ctx: Context) {
        if (cities.isEmpty()) {
            try {
                val inputStream = ctx.resources.openRawResource(R.raw.cities)
                val lines = inputStream.bufferedReader().readLines()

                for (i in 1 until lines.size) {
                    val line = lines[i].trim()
                    if (line.isNotEmpty()) {
                        val parts = line.split(";")
                        if (parts.size >= 9) {
                            try {
                                val city = City(
                                    title = parts[3].trim(),
                                    region = parts[2].trim(),
                                    district = parts[1].trim(),
                                    postalCode = parts[0].trim(),
                                    timezone = parts[4].trim(),
                                    population = parts[7].trim(),
                                    founded = parts[8].trim(),
                                    lat = parts[5].trim().toFloat(),
                                    lon = parts[6].trim().toFloat()
                                )
                                cities.add(city)
                            } catch (e: NumberFormatException) {
                                Log.e("Common", "Error parsing numeric data", e)
                            } catch (e: Exception) {
                                Log.e("Common", "Error parsing data", e)
                            }
                        }
                    }
                }
                cities.sortBy { it.title }
                Log.d("Common", "Loaded ${cities.size} cities")
            } catch (e: Exception) {
                Log.e("Common", "Error loading cities", e)
            }
        }
    }
}
```

## 5. Реализована основная активность:

### – MainActivity.kt

```
package ru.olegkravtsov.lab14

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import ru.olegkravtsov.lab14.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding
    private var isDualPane = false

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setSupportActionBar(binding.toolbar)

        Common.initCities(this)

        // Проверяем, есть ли контейнер для деталей (альбомная ориентация)
        isDualPane =
            findViewById<androidx.fragment.app.FragmentContainerView>(R.id.fragment_detail_container) != null

        // Всегда устанавливаем фрагменты заново при создании
        setupFragments()
    }

    private fun setupFragments() {
        // В портретной ориентации очищаем back stack при запуске
        if (!isDualPane) {
            supportFragmentManager.popBackStackImmediate(null,
                androidx.fragment.app.FragmentManager.POP_BACK_STACK_INCLUSIVE)
        }

        // Всегда показываем список городов в левом контейнере
        supportFragmentManager.beginTransaction()
            .replace(R.id.fragment_list_container, CityListFragment())
            .commit()

        // Если альбомная ориентация, загружаем фрагмент деталей
        if (isDualPane) {
            val cityToShow = Common.selectedCity ?: Common.cities.firstOrNull()
            if (cityToShow != null) {
                supportFragmentManager.beginTransaction()
                    .replace(R.id.fragment_detail_container,
                        CityDetailFragment.newInstance(cityToShow))
                    .commit()
            } else {
                // Если нет выбранного города, показываем пустой фрагмент деталей
                supportFragmentManager.beginTransaction()
                    .replace(R.id.fragment_detail_container, CityDetailFragment())
            }
        }
    }
}
```

```
        .commit()
    }
}

fun onCitySelected(city: City) {
    // Сохраняем выбранный город
    Common.selectedCity = city

    if (isDualPane) {
        // Альбомная ориентация - заменяем фрагмент деталей
        supportFragmentManager.beginTransaction()
            .replace(R.id.fragment_detail_container,
CityDetailFragment.newInstance(city))
            .commit()
    } else {
        // Портретная ориентация - заменяем фрагмент списка на детали
        supportFragmentManager.beginTransaction()
            .replace(R.id.fragment_list_container,
CityDetailFragment.newInstance(city))
            .addToBackStack("city_detail")
            .commit()
    }
}

override fun onSupportNavigateUp(): Boolean {
    onBackPressed()
    return true
}
}
```



## 6. Реализованы фрагменты:

### – CityListFragment.kt

```
package ru.olegkravtsov.lab14

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.recyclerview.widget.LinearLayoutManager
import ru.olegkravtsov.lab14.databinding.FragmentCityListBinding

class CityListFragment : Fragment() {

    private var _binding: FragmentCityListBinding? = null
    private val binding get() = _binding!!
    private lateinit var adapter: CityAdapter
    private var layoutManagerState: android.os.Parcelable? = null

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        _binding = FragmentCityListBinding.inflate(inflater, container, false)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        // Восстанавливаем состояние прокрутки
        if (savedInstanceState != null) {
            layoutManagerState = savedInstanceState.getParcelable("list_state")
        }

        adapter = CityAdapter(Common.cities) { city ->
            (requireActivity() as MainActivity).onCitySelected(city)
        }

        binding.rvCities.layoutManager = LinearLayoutManager(requireContext())
        binding.rvCities.adapter = adapter

        // Восстанавливаем позицию прокрутки после создания разметки
        if (layoutManagerState != null) {
            binding.rvCities.layoutManager?.onRestoreInstanceState(layoutManagerState)
        }

        override fun onSaveInstanceState(outState: Bundle) {
            super.onSaveInstanceState(outState)
            // Сохраняем состояние прокрутки только если binding доступен
            _binding?.let { binding ->
                val state = binding.rvCities.layoutManager?.onSaveInstanceState()
            }
        }
    }
}
```

```
        outState.putParcelable("list_state", state)
    }
}

override fun onDestroyView() {
    super.onDestroyView()
    _binding = null
}
}
```

## – CityDetailFragment.kt

```
package ru.olegkravtsov.lab14

import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import ru.olegkravtsov.lab14.databinding.FragmentCityDetailBinding

class CityDetailFragment : Fragment() {

    private var _binding: FragmentCityDetailBinding? = null
    private val binding get() = _binding!!
    private var selectedCity: City? = null

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        _binding = FragmentCityDetailBinding.inflate(inflater, container, false)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        // Получаем переданный город из аргументов
        arguments?.let {
            selectedCity = it.getParcelable("city")
        }

        // Если город не передан, пытаемся взять из Common
        if (selectedCity == null) {
            selectedCity = Common.selectedCity
        }

        binding.btnShowOnMap.setOnClickListener {
            selectedCity?.let { city ->
                showOnMap(city)
            }
        }

        updateCityInfo()
    }

    private fun updateCityInfo() {
        selectedCity?.let { city ->
            binding.tvCity.text = getString(R.string.city, city.title)
            binding.tvDistrict.text = getString(R.string.federal_district,
city.district)
            binding.tvRegion.text = getString(R.string.region, city.region)
            binding.tvPostalCode.text = getString(R.string.postal_code,
```

```

city.postalCode)
    binding.tvTimezone.text = getString(R.string.timezone, city.timezone)
    binding.tvPopulation.text = getString(R.string.population,
city.population)
    binding.tvFounded.text = getString(R.string.founded, city.founded)
    binding.btnShowOnMap.isEnabled = true
} ?: run {
    val unknown = getString(R.string.unknown)
    binding.tvCity.text = getString(R.string.city, unknown)
    binding.tvDistrict.text = getString(R.string.federal_district, unknown)
    binding.tvRegion.text = getString(R.string.region, unknown)
    binding.tvPostalCode.text = getString(R.string.postal_code, unknown)
    binding.tvTimezone.text = getString(R.string.timezone, unknown)
    binding.tvPopulation.text = getString(R.string.population, unknown)
    binding.tvFounded.text = getString(R.string.founded, unknown)
    binding.btnShowOnMap.isEnabled = false
}
}

private fun showOnMap(city: City) {
    val geoUri = "geo:${city.lat},${city.lon}?z=12"
    val mapIntent = Intent(Intent.ACTION_VIEW, Uri.parse(geoUri))

    try {
        startActivity(mapIntent)
    } catch (e: Exception) {
        // Fallback
        val browserUri =
"https://www.google.com/maps/@${city.lat},${city.lon},12z"
        val browserIntent = Intent(Intent.ACTION_VIEW, Uri.parse(browserUri))
        startActivity(browserIntent)
    }
}

override fun onDestroyView() {
    super.onDestroyView()
    _binding = null
}

companion object {
    fun newInstance(city: City): CityDetailFragment {
        val fragment = CityDetailFragment()
        val args = Bundle()
        args.putParcelable("city", city)
        fragment.arguments = args
        return fragment
    }
}
}

```

## 7. Реализован адаптер для списка:

### – CityAdapter.kt

```
package ru.olegkravtsov.lab14

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import ru.olegkravtsov.lab14.databinding.ItemCityBinding

class CityAdapter(
    private val cities: List<City>,
    private val onItemClick: (City) -> Unit
) : RecyclerView.Adapter<CityAdapter.CityViewHolder>() {

    class CityViewHolder(private val binding: ItemCityBinding) :
        RecyclerView.ViewHolder(binding.root) {
        fun bind(city: City) {
            binding.tvCityName.text = city.title
            binding.tvRegion.text = city.region
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): CityViewHolder {
        val binding = ItemCityBinding.inflate(LayoutInflater.from(parent.context),
parent, false)
        return CityViewHolder(binding)
    }

    override fun onBindViewHolder(holder: CityViewHolder, position: Int) {
        val city = cities[position]
        holder.bind(city)

        holder.itemView.setOnClickListener {
            onItemClick(city)
        }
    }

    override fun getItemCount(): Int = cities.size
}
```

## 8. Настроены макеты:

– activity\_main.xml (портретная ориентация)

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
        android:fitsSystemWindows="true">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/ThemeOverlay.AppCompat.Light"
            app:title="@string/app_name" />

    </com.google.android.material.appbar.AppBarLayout>

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragment_list_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

– a

– ctivity\_main.xml (альбомная ориентация)

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
        android:fitsSystemWindows="true">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/ThemeOverlay.AppCompat.Light"
            app:title="@string/app_name" />

    </com.google.android.material.appbar.AppBarLayout>

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragment_list_container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

## – fragment\_city\_list.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <androidx.coordinatorlayout.widget.CoordinatorLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/rvCities"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:paddingTop="@dimen/padding_small"
            android:paddingBottom="@dimen/padding_medium"
            app:layout_behavior="@string/appbar_scrolling_view_behavior" />

    </androidx.coordinatorlayout.widget.CoordinatorLayout>
</layout>
```



## – fragment\_city\_detail.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <androidx.coordinatorlayout.widget.CoordinatorLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <com.google.android.material.card.MaterialCardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="@dimen/padding_medium"
            app:cardCornerRadius="@dimen/item_corner_radius"
            app:cardElevation="@dimen/item_elevation"
            app:layout_behavior="@string/appbar_scrolling_view_behavior">

            <androidx.core.widget.NestedScrollView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:fillViewport="true">

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical"
                    android:padding="@dimen/padding_medium">

                    <TextView
                        android:id="@+id/tvCity"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:textSize="@dimen/text_size_large"
                        android:textStyle="bold"
                        android:layout_marginBottom="@dimen/margin_bottom_medium" />

                    <TextView
                        android:id="@+id/tvDistrict"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:textSize="@dimen/text_size_medium"
                        android:layout_marginBottom="@dimen/margin_bottom_small" />

                    <TextView
                        android:id="@+id/tvRegion"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:textSize="@dimen/text_size_medium"
                        android:layout_marginBottom="@dimen/margin_bottom_small" />

                    <TextView
                        android:id="@+id/tvPostalCode"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:textSize="@dimen/text_size_medium"
                        android:layout_marginBottom="@dimen/margin_bottom_small" />

                </LinearLayout>
            </NestedScrollView>
        </MaterialCardView>
    </CoordinatorLayout>
</layout>
```

```
<TextView
    android:id="@+id/tvTimezone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="@dimen/text_size_medium"
    android:layout_marginBottom="@dimen/margin_bottom_small" />

<TextView
    android:id="@+id/tvPopulation"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="@dimen/text_size_medium"
    android:layout_marginBottom="@dimen/margin_bottom_small" />

<TextView
    android:id="@+id/tvFounded"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="@dimen/text_size_medium"
    android:layout_marginBottom="@dimen/margin_bottom_large" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btnShowOnMap"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/show_on_map"
    style="@style/Widget.Material3.Button.OutlinedButton" />

</LinearLayout>

</androidx.core.widget.NestedScrollView>

</com.google.android.material.card.MaterialCardView>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
</layout>
```

## – item\_city.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="@dimen/padding_medium"
    android:background="@drawable/city_item_background"
    android:layout_marginStart="@dimen/padding_medium"
    android:layout_marginEnd="@dimen/padding_medium"
    android:layout_marginTop="@dimen/padding_small"
    android:layout_marginBottom="@dimen/padding_small">

    <TextView
        android:id="@+id/tvCityName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="@dimen/text_size_large"
        android:textStyle="bold"
        android:textColor="@color/text_primary" />

    <TextView
        android:id="@+id/tvRegion"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="@dimen/text_size_small"
        android:textColor="@color/text_secondary"
        android:layout_marginTop="@dimen/padding_very_small" />

</LinearLayout>
```

## 9. Настроен манифест:

### – AndroidManifest.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">

    <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:theme="@style/Theme.Lab14"
        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>
```

## Результат работы

Приложение успешно запускается и работает в двух режимах. В портретной ориентации отображается список городов (рис. 1), при выборе города происходит переход к информации о нем с возможностью возврата назад (рис. 2).

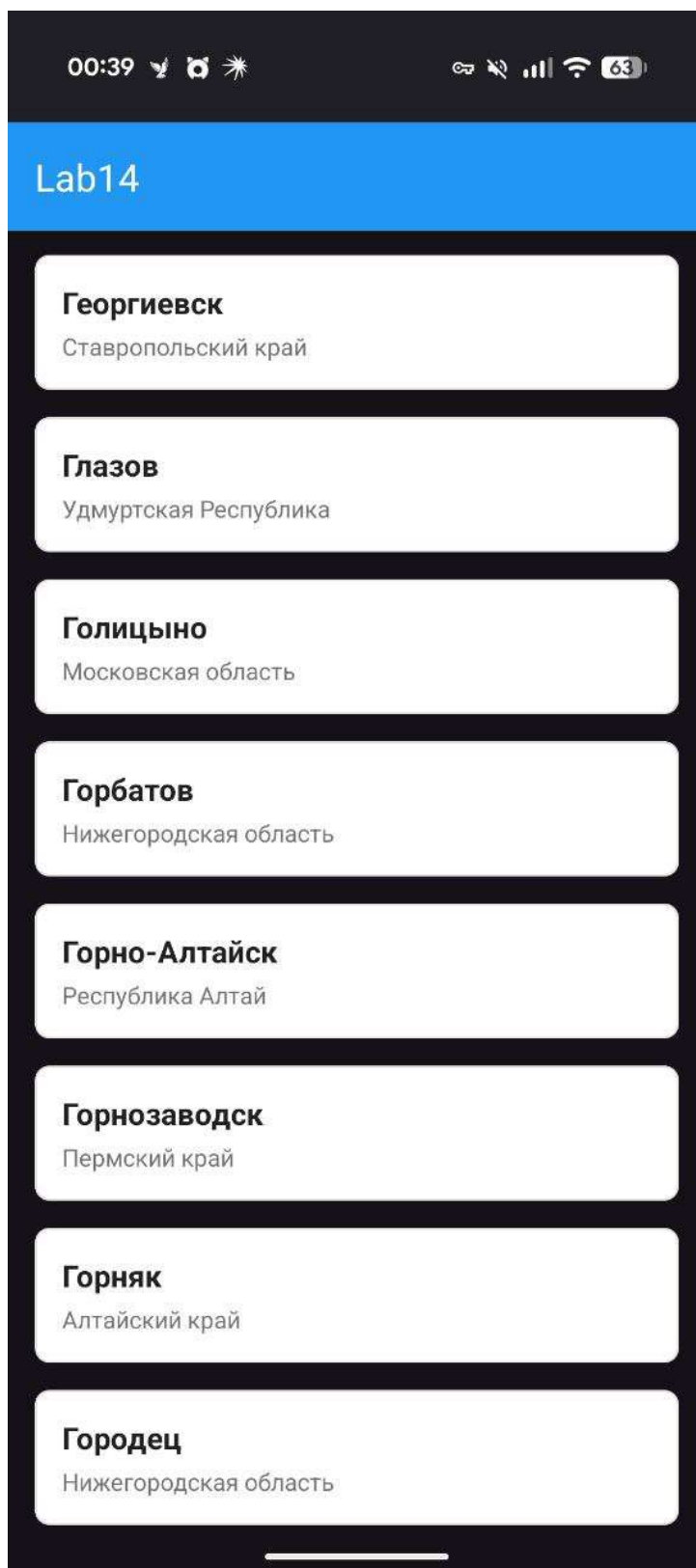


Рисунок 1 – Главный экран со списком городов (портретная ориентация)

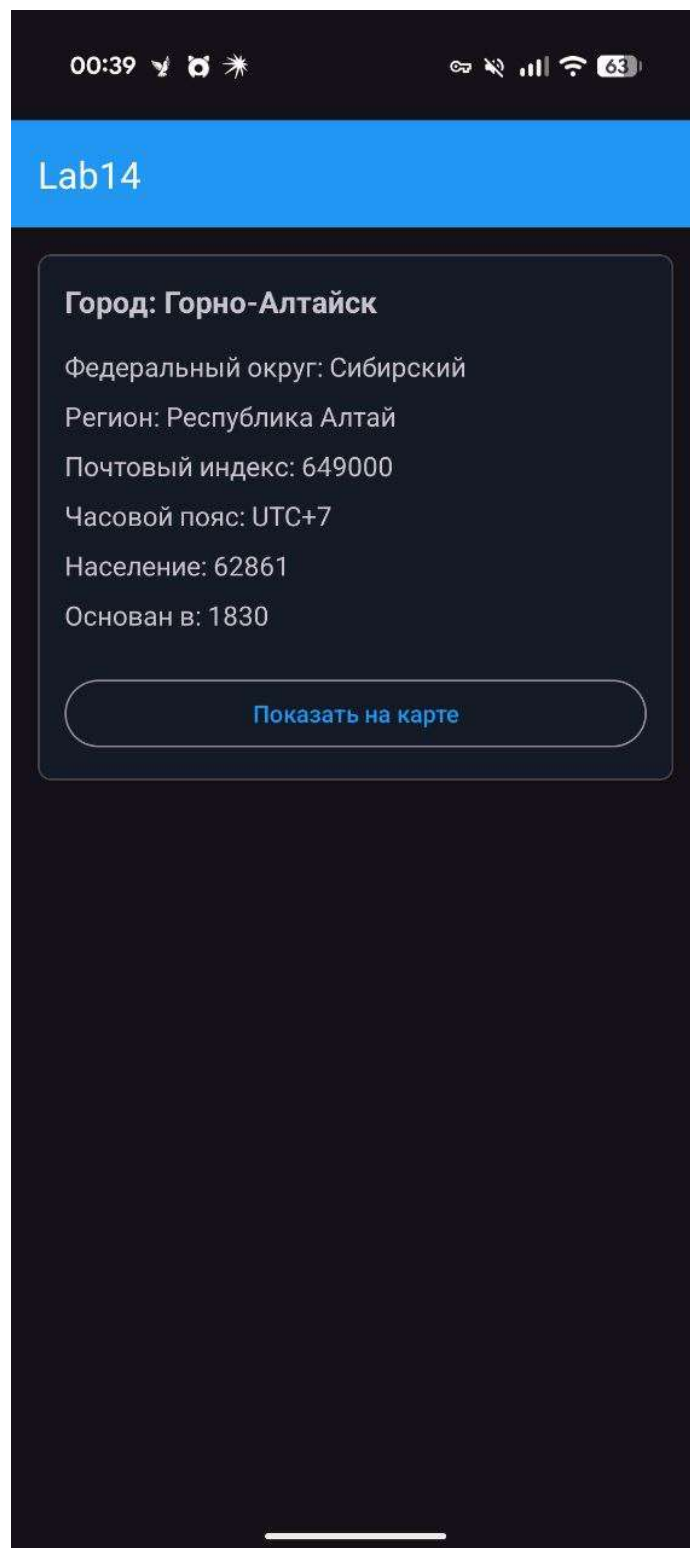


Рисунок 2 – Информация о выбранном городе (портретная ориентация)

В альбомной ориентации одновременно отображаются список городов и информация о выбранном городе (рис. 3). При нажатии кнопки "Показать на карте" открывается карта с местоположением города (рис. 4).

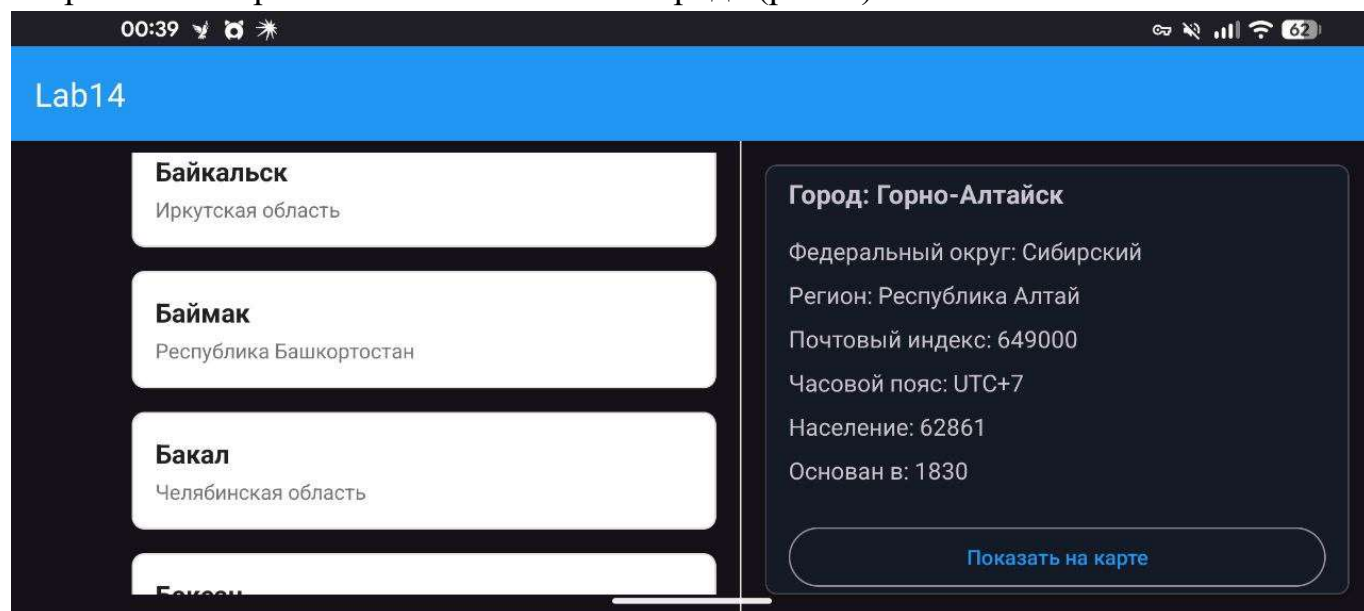


Рисунок 3 – Двухпанельный интерфейс (альбомная ориентация)

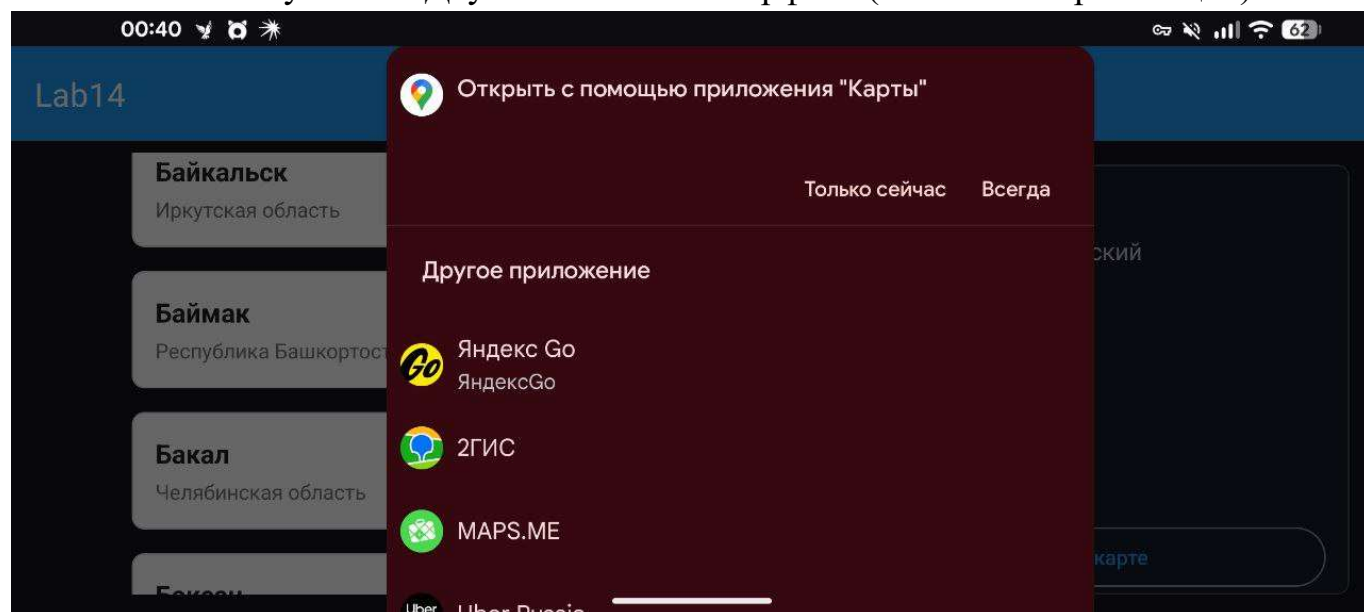


Рисунок 4 – Отображение города на карте

## **Выводы**

В ходе выполнения лабораторной работы было разработано приложение для отображения информации о городах с использованием фрагментов. Были освоены принципы работы с фрагментами, их жизненным циклом и взаимодействием между ними. Реализована адаптация интерфейса под разные ориентации устройства. Приложение корректно обрабатывает смену конфигурации и сохраняет состояние элементов интерфейса. Полученные навыки позволяют создавать гибкие и адаптивные пользовательские интерфейсы для Android-приложений.