**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 1А22 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.К. Кравцов

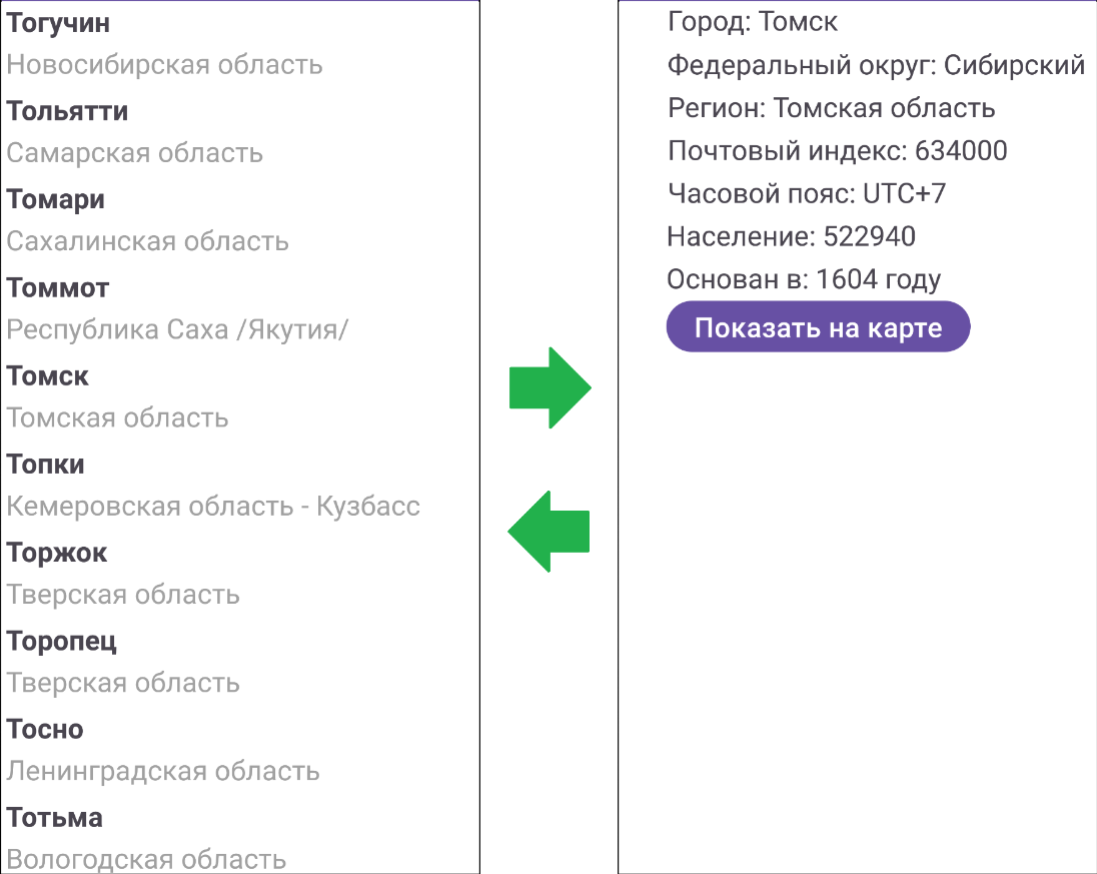
Проверил:

Ст. преп. ОИТ ИШИТР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Дорофеев

Томск 2025

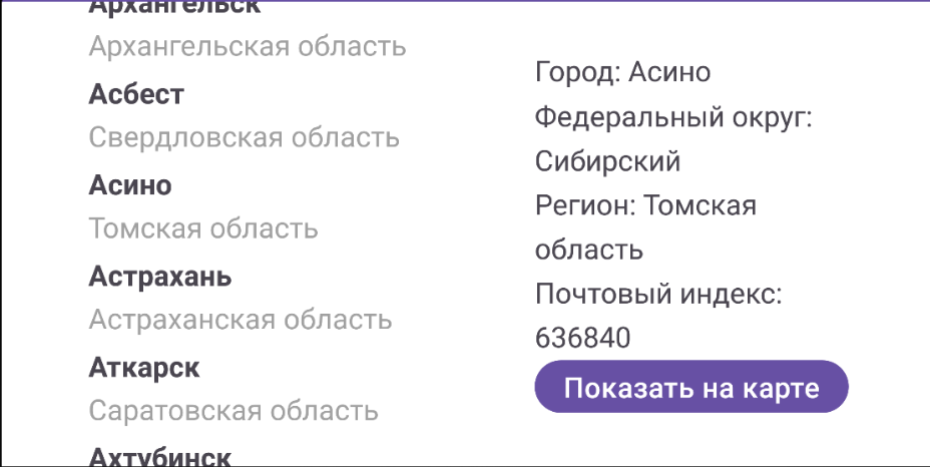
# Задание

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">#FFFFFFFF</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>

* dimens.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>

1. 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

1. 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)  
 }  
 }  
 }  
}

1. 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  
 }  
}

1. 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  
 }  
 }  
}

1. 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  
}

1. 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" />  
  
 <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>

1. 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).

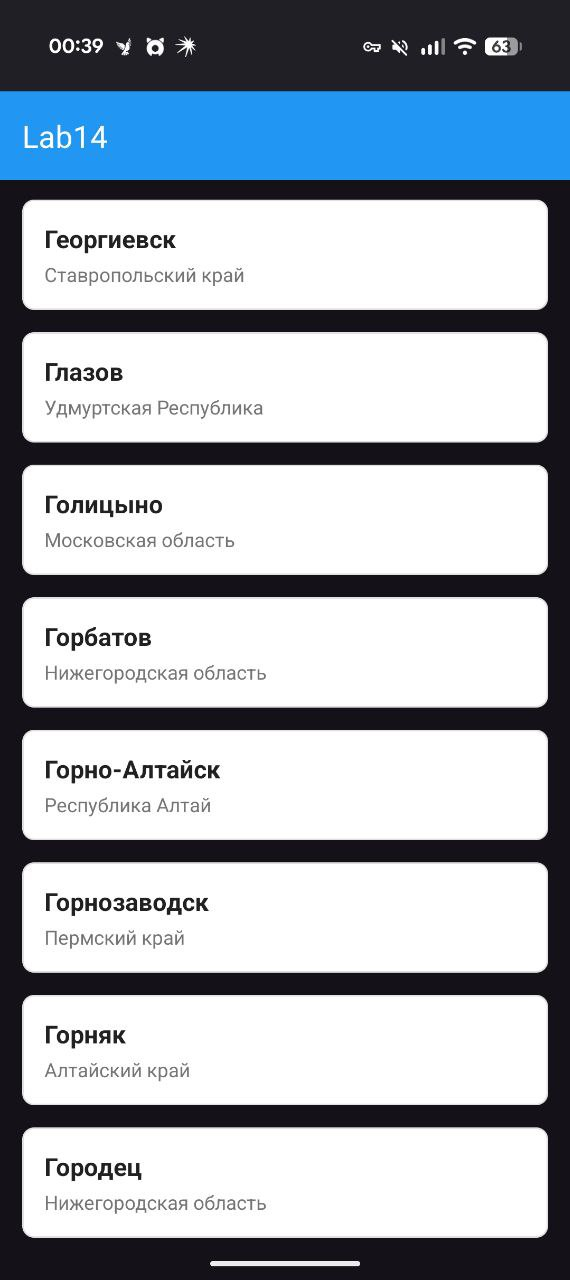


Рисунок 1 – Главный экран со списком городов (портретная ориентация)

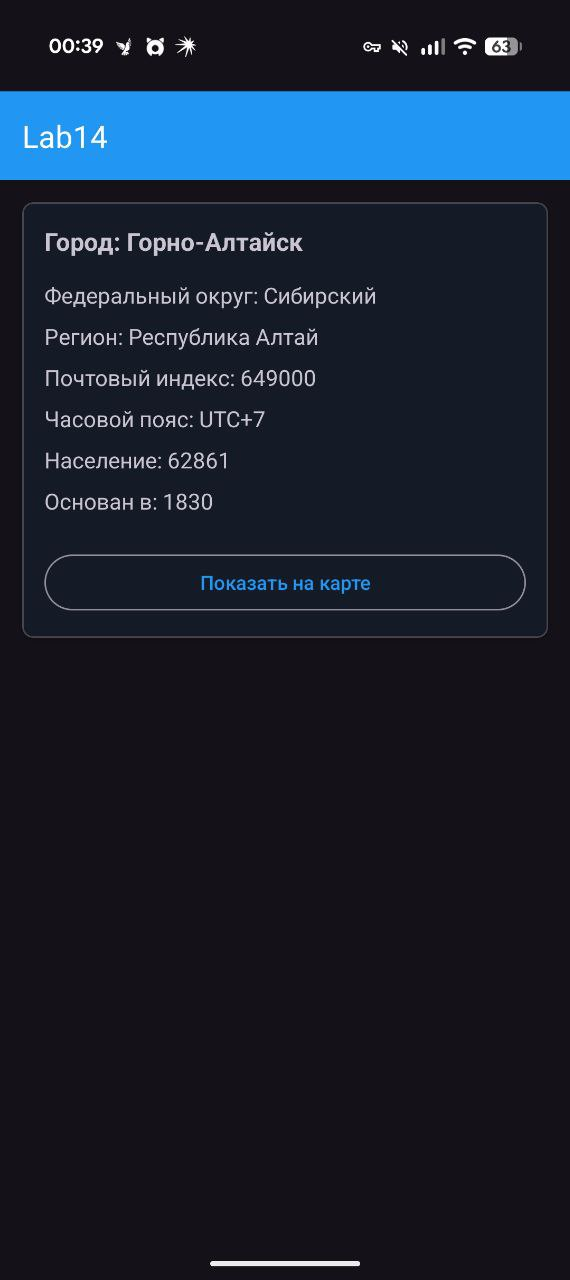


Рисунок 2 – Информация о выбранном городе (портретная ориентация)

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).

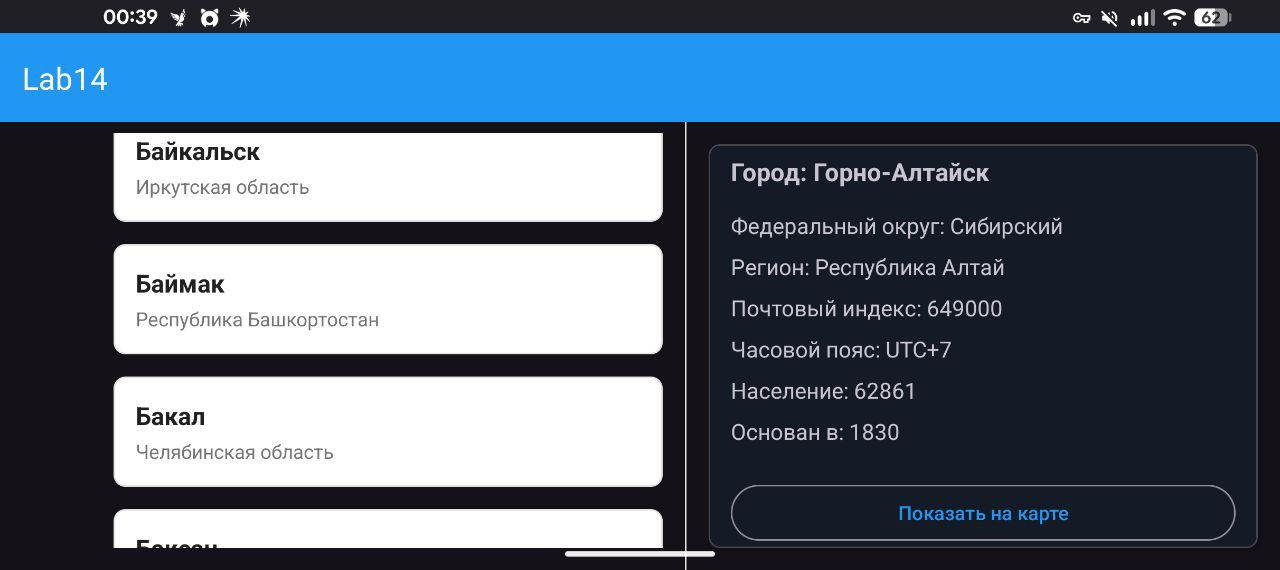


Рисунок 3 – Двухпанельный интерфейс (альбомная ориентация)

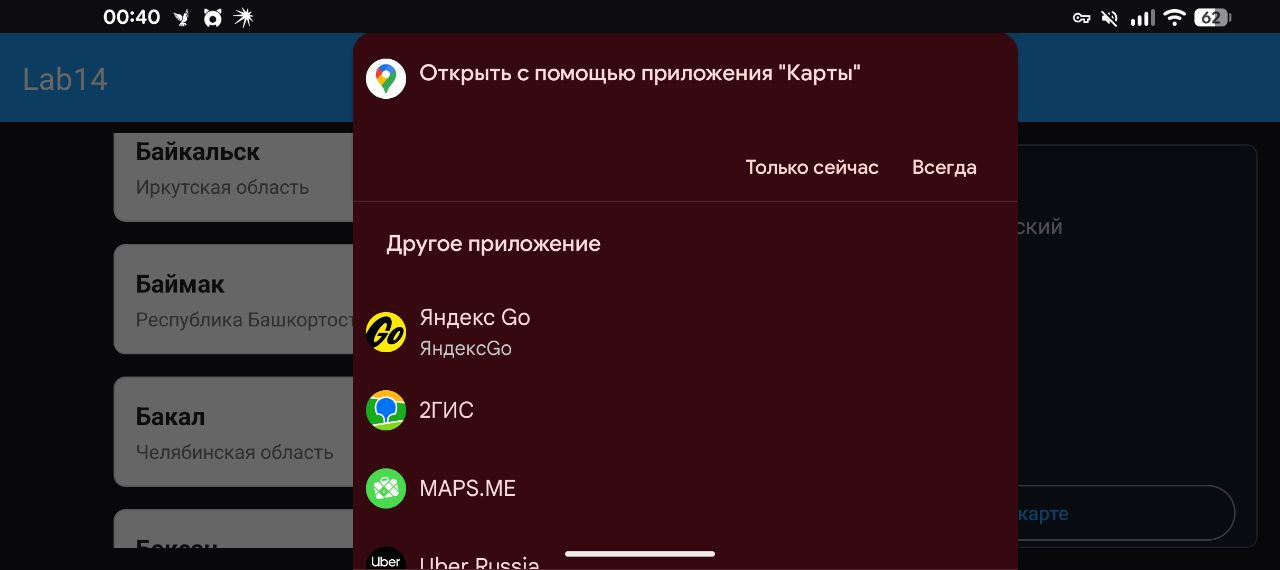


Рисунок 4 – Отображение города на карте

# 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.