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 №17 по дисциплине

«Язык Kotlin и основы разработки»

Вкладки

Выполнил:

Студент группы 1А22 _____ О.К. Кравцов

Проверил:

Ст. преп. ОИТ ИШИТР _____ В.А. Дорофеев

Томск 2025

Задание

Напишите приложение-суперапп, которое, как и многие другие супераппы, сочетает в себе малосочетаемое: музыку, книги и новости.

В нижней части приложения должен располагаться элемент BottomNavigationView с кнопками разделов: Музыка, Книги, Новости. При нажатии одной из кнопок происходит переход в соответствующий раздел. Можно реализовать разделы на фрагментах с навигационным графом, или другим способом.

В разделе Музыка в верхней части должен располагаться элемент TabLayout со вкладками, соответствующим разным жанрам музыки. Вкладок должно быть достаточно много, чтобы они не помещались на экран, элемент TabLayout должен работать в режиме scrollable.

Содержимое экрана должно располагаться в слайдере ViewPager2, каждая вкладка связана с соответствующим фрагментом (можно использовать один фрагмент и передавать ему информацию о жанре на выбранной вкладке):

В разделе Книги в верхней части должен располагаться элемент TabLayout со вкладками Новое и Прочитанное, элемент TabLayout должен работать в режиме fixed. Содержимое экрана должно располагаться во фрагменте, в котором каким-либо образом отображается информация о выбранной вкладке:

При запуске программы каждые несколько секунд должно увеличиваться количество непрочитанных новостей в бейдже на кнопке Новости:

Если зайти в раздел Новости, то количество новостей прекращает увеличиваться, а если уйти из раздела – то снова начинает расти. В нижней части раздела располагается кнопка Очистить новости, которая сбрасывает счётчик в 0, и скрывает его. Когда количество новостей снова начинает увеличиваться – счётчик должен снова появиться.

Это пример программы, можно выбрать какую-то свою тематику приложения, но описанный функционал должен быть реализован и там.

Ход работы

1. Создан проект Lab17 на основе Empty Views Activity.
2. Настроены ресурсы приложения:

– strings.xml

```
<resources>
    <string name="app_name">Lab17</string>

    <!-- Нижняя навигация -->
    <string name="music">Музыка</string>
    <string name="books">Книги</string>
    <string name="news">Новости</string>

    <!-- Жанры музыки (обновленные) -->
    <string name="rock">Рок</string>
    <string name="rap">Реп</string>
    <string name="pop">Поп</string>
    <string name="jazz">Джаз</string>
    <string name="classic">Классика</string>
    <string name="edm">EDM</string>
    <string name="techno">Техно</string>
    <string name="house">Хаус</string>
    <string name="alternative">Альтернатива</string>
    <string name="folk">Фолк</string>

    <!-- Книги -->
    <string name="new_books">Новое</string>
    <string name="read_books">Прочитанное</string>
    <string name="new_books_content">Новые книги</string>
    <string name="read_books_content">Прочитанные книги</string>
    <string name="unknown_tab">Неизвестная вкладка</string>

    <!-- Новости -->
    <string name="news_section">Раздел новостей</string>
    <string name="clear_news">Очистить новости</string>

    <!-- Общие -->
    <string name="select_tab">Выберите вкладку</string>
    <string name="music_genre_pattern">Жанр: %s</string>
</resources>
```

– colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="primary_color">#2196F3</color>
</resources>
```

– dimens.xml

```
<resources>
    <!-- Размеры текста -->
    <dimen name="text_size_large">24sp</dimen>
    <dimen name="text_size_medium">18sp</dimen>
    <dimen name="text_size_small">14sp</dimen>

    <!-- Отступы -->
    <dimen name="padding_small">8dp</dimen>
    <dimen name="padding_medium">16dp</dimen>
    <dimen name="padding_large">24dp</dimen>

    <!-- Размеры элементов -->
    <dimen name="button_height">48dp</dimen>
    <dimen name="tab_height">48dp</dimen>
</resources>
```

3. Реализована нижняя панель навигации с использованием BottomNavigationView. Создано меню с тремя разделами:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu_music"
        android:icon="@drawable/ic_music"
        android:title="@string/music" />
    <item
        android:id="@+id/menu_books"
        android:icon="@drawable/ic_books"
        android:title="@string/books" />
    <item
        android:id="@+id/menu_news"
        android:icon="@drawable/ic_news"
        android:title="@string/news" />
</menu>
```

4. Реализована основная активность MainActivity с обработкой навигации:

```
package ru.olegkravtsov.lab17

import android.os.Bundle
import android.os.CountDownTimer
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import androidx.fragment.app.Fragment
import com.google.android.material.bottomnavigation.BottomNavigationView

class MainActivity : AppCompatActivity() {

    private lateinit var bottomNav: BottomNavigationView
    private var newsCount = 0
    private var timer: CountDownTimer? = null
    private var isInNewsFragment = false

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v,
insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
            insets
        }

        bottomNav = findViewById(R.id.bottom_navigation)

        // Загружаем первый фрагмент по умолчанию
        if (savedInstanceState == null) {
            replaceFragment(MusicFragment())
        }

        bottomNav.setOnItemSelectedListener { item ->
            when (item.itemId) {
                R.id.menu_music -> {
                    replaceFragment(MusicFragment())
                    true
                }
                R.id.menu_books -> {
                    replaceFragment(BooksFragment())
                    true
                }
                R.id.menu_news -> {
                    isInNewsFragment = true
                    replaceFragment(NewsFragment())
                    true
                }
                else -> false
            }
        }
```

```
        startNewsTimer()
    }

    private fun startNewsTimer() {
        timer = object : CountDownTimer(1000000000, 2000) {
            override fun onTick(millisUntilFinished: Long) {
                if (!isInNewsFragment) {
                    newsCount++
                    updateNewsBadge()
                }
            }

            override fun onFinish() {
                // Перезапускаем таймер
                startNewsTimer()
            }
        }.start()
    }

    private fun updateNewsBadge() {
        val badge = bottomNav.getOrCreateBadge(R.id.menu_news)
        badge.number = newsCount
        badge.isVisible = true
    }

    fun clearNewsBadge() {
        val badge = bottomNav.getOrCreateBadge(R.id.menu_news)
        badge.isVisible = false
        newsCount = 0
    }

    fun setInNewsFragment(value: Boolean) {
        isInNewsFragment = value
    }

    private fun replaceFragment(fragment: Fragment) {
        if (fragment is NewsFragment) {
            isInNewsFragment = true
        } else {
            isInNewsFragment = false
        }

        supportFragmentManager.beginTransaction()
            .replace(R.id.fragment_container, fragment)
            .commit()
    }

    override fun onDestroy() {
        super.onDestroy()
        timer?.cancel()
    }
}
```

5. Созданы фрагменты для каждого раздела приложения:

– MusicFragment.kt

```
package ru.olegkravtsov.lab17

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.viewpager2.adapter.FragmentStateAdapter
import androidx.viewpager2.widget.ViewPager2
import com.google.android.material.tabs.TabLayout
import com.google.android.material.tabs.TabLayoutMediator
import ru.olegkravtsov.lab17.databinding.FragmentMusicBinding

class MusicFragment : Fragment() {

    private var _binding: FragmentMusicBinding? = null
    private val binding get() = _binding!!

    private lateinit var genres: List<String>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        // Инициализируем список жанров в onCreate, где контекст уже доступен
        genres = listOf(
            getString(R.string.rock),
            getString(R.string.rap),
            getString(R.string.pop),
            getString(R.string.jazz),
            getString(R.string.classic),
            getString(R.string.edm),
            getString(R.string.techno),
            getString(R.string.house),
            getString(R.string.alternative),
            getString(R.string.folk)
        )
    }

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

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

        val adapter = MusicPagerAdapter(this)
        binding.pager.adapter = adapter

        TabLayoutMediator(binding.tabs, binding.pager) { tab, position ->
            tab.text = genres[position]
```

```
        }.attach()
    }

    inner class MusicPagerAdapter(fragment: Fragment) :
FragmentStateAdapter(fragment) {
        override fun getItemCount(): Int = genres.size

        override fun createFragment(position: Int): Fragment {
            return MusicGenreFragment.newInstance(genres[position])
        }
    }

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

– MusicGenreFragment.kt

```
package ru.olegkravtsov.lab17

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import ru.olegkravtsov.lab17.databinding.FragmentMusicGenreBinding

class MusicGenreFragment : Fragment() {

    companion object {
        private const val ARG_GENRE = "genre"

        fun newInstance(genre: String): MusicGenreFragment {
            val args = Bundle().apply {
                putString(ARG_GENRE, genre)
            }
            return MusicGenreFragment().apply {
                arguments = args
            }
        }
    }

    private var _binding: FragmentMusicGenreBinding? = null
    private val binding get() = _binding!!

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

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

        val genre = arguments?.getString(ARG_GENRE) ?:
getString(R.string.unknown_tab)
        binding.genreText.text = getString(R.string.music_genre_pattern,
genre)
    }

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

– BooksFragment.kt

```
package ru.olegkravtsov.lab17

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import com.google.android.material.tabs.TabLayout
import ru.olegkravtsov.lab17.databinding.FragmentBooksBinding

class BooksFragment : Fragment() {

    private var _binding: FragmentBooksBinding? = null
    private val binding get() = _binding!!

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

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


binding.tabs.addTab(binding.tabs.newTab().setText(getString(R.string.new_book
s)))

binding.tabs.addTab(binding.tabs.newTab().setText(getString(R.string.read_boo
ks)))

        updateContent(0)

        binding.tabs.addOnTabSelectedListener(object :
TabLayout.OnTabSelectedListener {
            override fun onTabSelected(tab: TabLayout.Tab?) {
                updateContent(tab?.position ?: 0)
            }

            override fun onTabUnselected(tab: TabLayout.Tab?) {}
            override fun onTabReselected(tab: TabLayout.Tab?) {}
        })
    }

    private fun updateContent(position: Int) {
        val text = when (position) {
            0 -> getString(R.string.new_books_content)
            1 -> getString(R.string.read_books_content)
```

```
            else -> getString(R.string.unknown_tab)
        }
        binding.contentText.text = text
    }

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

– NewsFragment.kt

```kotlin
package ru.olegkravtsov.lab17

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import ru.olegkravtsov.lab17.databinding.FragmentNewsBinding

class NewsFragment : Fragment() {

    private var _binding: FragmentNewsBinding? = null
    private val binding get() = _binding!!

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

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

        binding.newsText.text = getString(R.string.news_section)
        binding.clearNewsButton.text = getString(R.string.clear_news)

        binding.clearNewsButton.setOnClickListener {
            (requireActivity() as MainActivity).clearNewsBadge()
        }
    }

    override fun onResume() {
        super.onResume()
        (requireActivity() as MainActivity).setInNewsFragment(true)
    }

    override fun onPause() {
        super.onPause()
        (requireActivity() as MainActivity).setInNewsFragment(false)
    }

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

6. Настроены макеты активности и фрагментов:

– activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <FrameLayout
        android:id="@+id/fragment_container"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toTopOf="@+id/bottom_navigation"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:menu="@menu/bottom_navigation_menu" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

– fragment_music.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MusicFragment">

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabs"
        android:layout_width="0dp"
        android:layout_height="@dimen/tab_height"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:tabMode="scrollable" />

    <androidx.viewpager2.widget.ViewPager2
        android:id="@+id/pager"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tabs" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

– fragment_music_genre.xml

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

    <TextView
        android:id="@+id/genre_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="@dimen/text_size_medium"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

– fragment_books.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".BooksFragment">

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabs"
        android:layout_width="0dp"
        android:layout_height="@dimen/tab_height"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:tabMode="fixed" />

    <TextView
        android:id="@+id/content_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/select_tab"
        android:textSize="@dimen/text_size_medium"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tabs" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

– fragment_news.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".NewsFragment">

    <TextView
        android:id="@+id/news_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="@dimen/text_size_large"
        app:layout_constraintBottom_toTopOf="@+id/clear_news_button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/clear_news_button"
        android:layout_width="wrap_content"
        android:layout_height="@dimen/button_height"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Результат работы

Приложение успешно запускается и отображает нижнюю панель навигации с тремя разделами. При переходе в раздел "Музыка" отображаются прокручиваемые вкладки с музыкальными жанрами (рис. 1). Переключение между вкладками осуществляется как по нажатию, так и с помощью горизонтального свайпа.

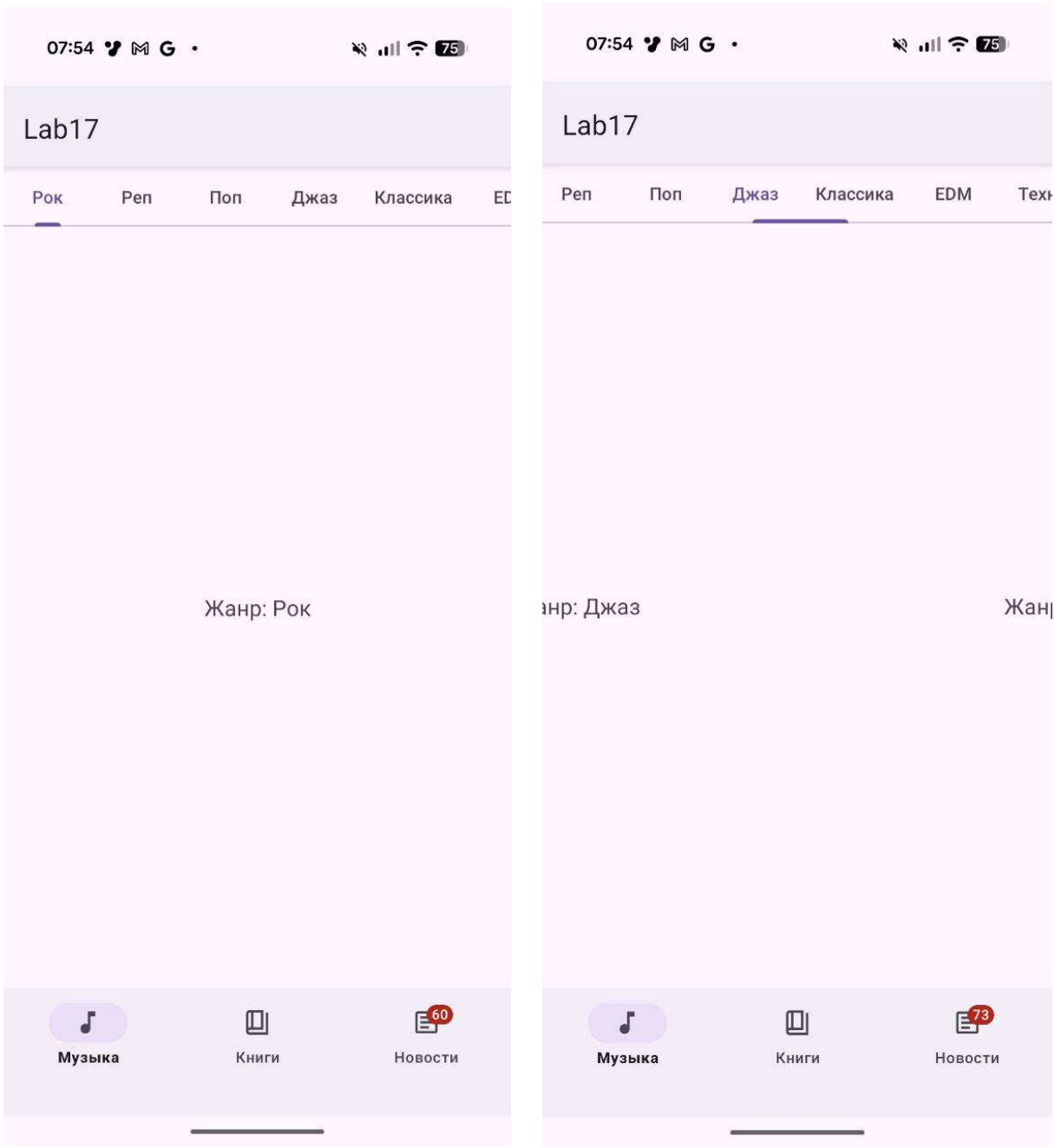

Рисунок 1 – Раздел «Музыка»

В разделе "Книги" присутствуют две фиксированные вкладки "Новое" и "Прочитанное" (рис. 2). При выборе вкладки изменяется отображаемая информация в соответствии с выбранной категорией.

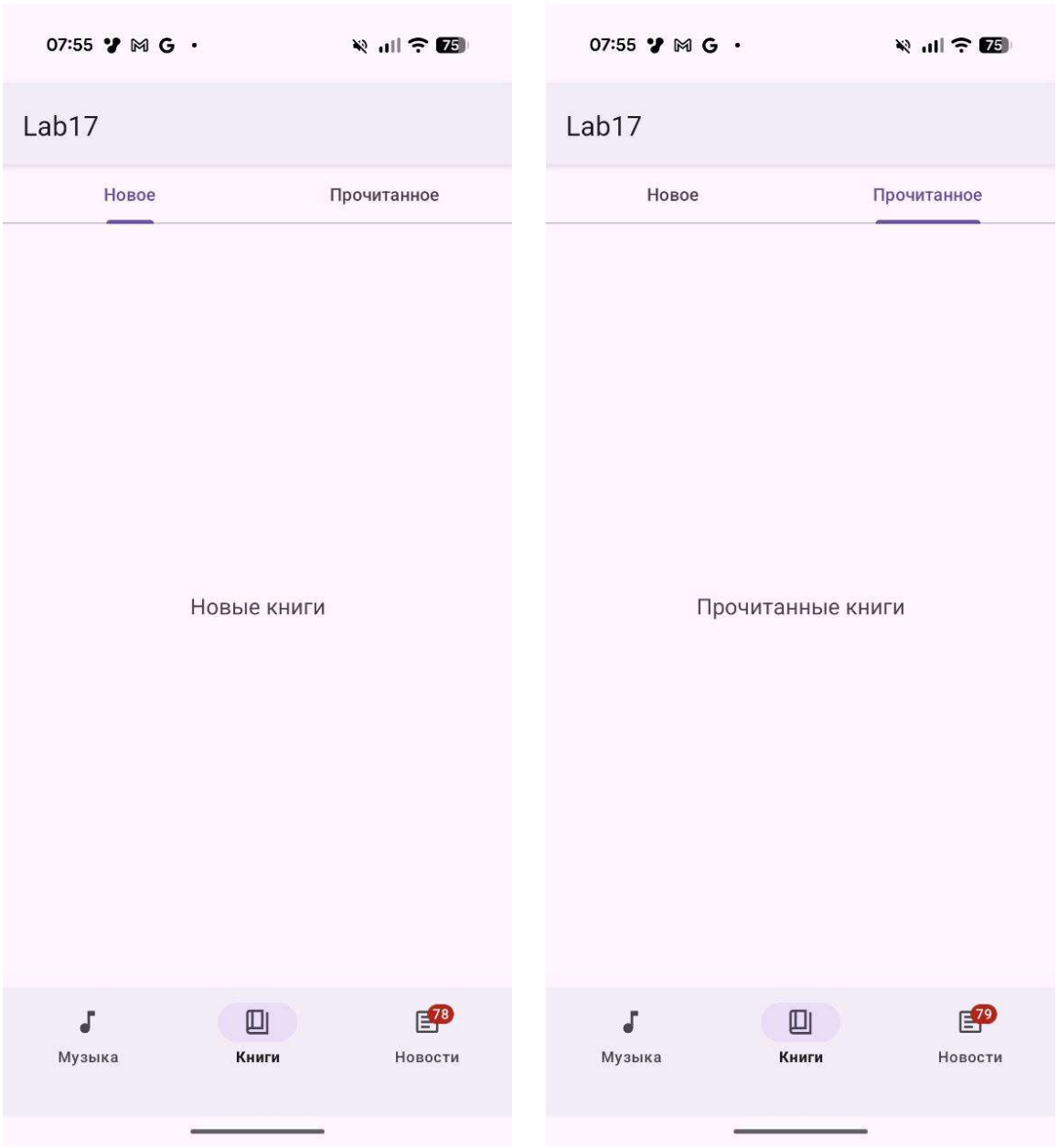

Рисунок 3 – Раздел «Книги»

Раздел "Новости" демонстрирует работу с бейджами. Счётчик непрочитанных новостей автоматически увеличивается каждые 2 секунды, когда пользователь не находится в данном разделе. При входе в раздел увеличение счётчика приостанавливается. Кнопка "Очистить новости" сбрасывает счётчик и скрывает бейдж до появления новых уведомлений (рис. 3).

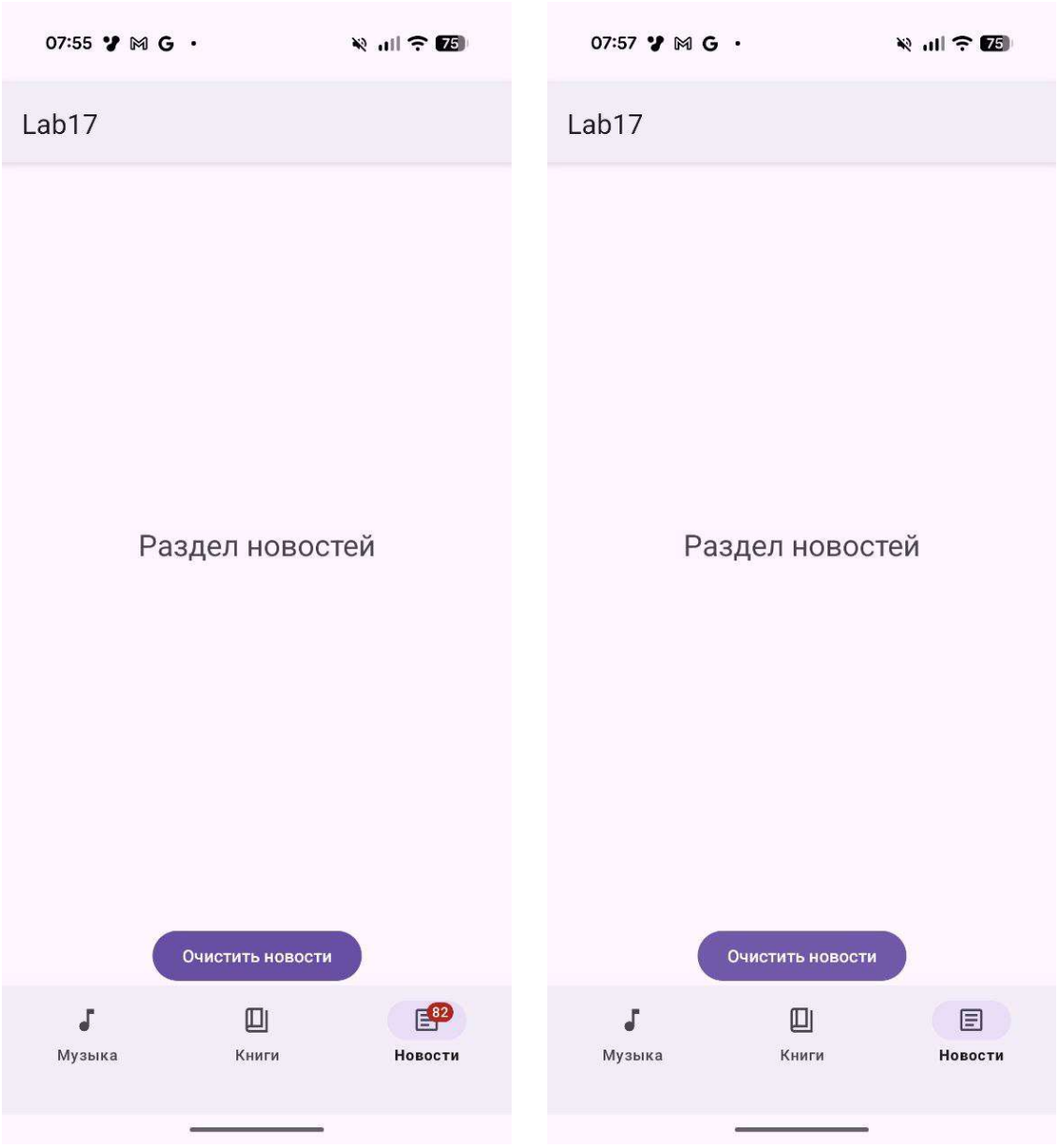

Рисунок 3 – Раздел «Новости»

Выводы

В ходе выполнения лабораторной работы было разработано многомодульное приложение с использованием современных компонентов навигации Android. Реализована нижняя панель навигации BottomNavigationView для переключения между основными разделами приложения. Освоены принципы работы с TabLayout в различных режимах (scrollable и fixed) и его интеграция с ViewPager2 для организации горизонтальной навигации. Также был успешно применён механизм бейджей для уведомлений и реализована логика управления состоянием счётчика с использованием CountDownTimer. Приложение демонстрирует правильную организацию сложной навигационной структуры и адаптацию интерфейса под различный контент. Полученные навыки позволяют создавать современные пользовательские интерфейсы с многоуровневой навигацией в соответствии с рекомендациями Material Design.