**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>

1. 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>

1. 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()  
 }  
}

1. 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\_books)))  
 binding.tabs.addTab(binding.tabs.newTab().setText(getString(R.string.read\_books)))  
  
 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

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  
 }  
}

1. 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.

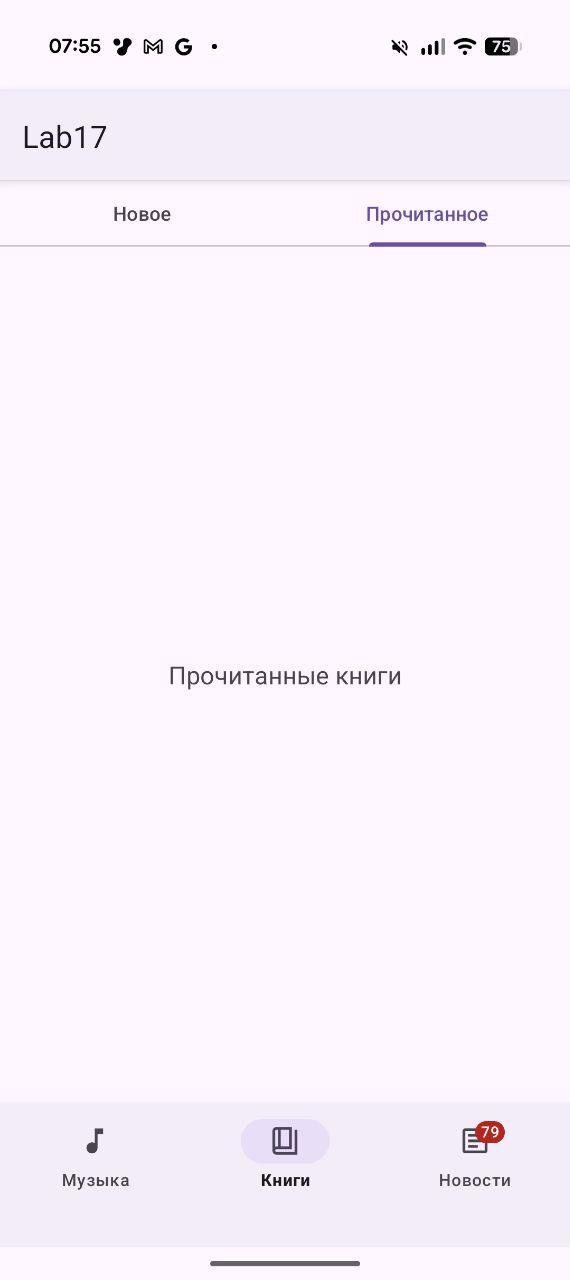
 

Рисунок 3 – Раздел «Книги»

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.