|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №4**

по дисциплине «Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнил:**  студент группыИКБО-28-22 | Некрасов Г.А. |
| **Проверил:**  к.э.н доцент | Степанов П.В. |

Москва 2024 г.

СОДЕРЖАНИЕ

[ПРАКТИЧЕСКОЕ ЗАДАНИЕ 3](#_Toc184915647)

[Код практического задания 4](#_Toc184915648)

[Тестирование практического задания 8](#_Toc184915649)

[Вывод 11](#_Toc184915650)

# ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Реализовать приложение, имеющее два экрана: «Камера» и «Список»:

- на экране «Камера» необходимо реализовать функционал просмотра камеры на экране, а также кнопку фотографирования, которая при нажатии будет сохранять в файл "date", находящийся в папке "photos" время и дату сделанной фотографии;

- на экране «Список» реализовать с помощью RecyclerView список, отображающий данные с файла "date" в хронологическом порядке.

# Код практического задания

В ходе выполнения работы, мы написали код для камеры, списка, а также MainActivity для выбора списка/камеры (см. Листинги 1-3)

Листинг 1. CameraActivity.kt

|  |
| --- |
| package com.example.practi4eskaya\_4  import android.Manifest import android.content.Intent import android.content.pm.PackageManager import android.os.Bundle import android.util.Log import android.widget.Button import androidx.activity.result.contract.ActivityResultContracts import androidx.appcompat.app.AppCompatActivity import androidx.core.app.ActivityCompat import androidx.core.content.ContextCompat import androidx.recyclerview.widget.LinearLayoutManager import androidx.recyclerview.widget.RecyclerView import java.io.File import java.text.SimpleDateFormat import java.util.\* import androidx.camera.core.Camera import androidx.camera.core.CameraSelector import androidx.camera.core.Preview import androidx.camera.lifecycle.ProcessCameraProvider import androidx.camera.view.PreviewView import java.util.concurrent.ExecutorService import java.util.concurrent.Executors   // CameraActivity: Handles camera preview and photo capturing class CameraActivity : AppCompatActivity() {  private val permissions = arrayOf(Manifest.permission.CAMERA, Manifest.permission.WRITE\_EXTERNAL\_STORAGE)  private lateinit var previewView: PreviewView   override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.activity\_camera)   previewView = findViewById(R.id.preview\_view)   if (!hasPermissions()) {  ActivityCompat.requestPermissions(this, permissions, 1)  } else {  startCamera()  }   val buttonCapture = findViewById<Button>(R.id.button\_capture)  buttonCapture.setOnClickListener {  savePhotoTimestamp()  }  }   private fun hasPermissions(): Boolean {  return permissions.all { perm ->  ContextCompat.checkSelfPermission(this, perm) == PackageManager.PERMISSION\_GRANTED  }  }   private fun startCamera() {  val cameraProviderFuture = ProcessCameraProvider.getInstance(this)   cameraProviderFuture.addListener({  val cameraProvider = cameraProviderFuture.get()   val preview = Preview.Builder().build().also {  it.setSurfaceProvider(previewView.surfaceProvider)  }   val cameraSelector = CameraSelector.DEFAULT\_BACK\_CAMERA   try {  cameraProvider.unbindAll()  cameraProvider.bindToLifecycle(this, cameraSelector, preview)  } catch (exc: Exception) {  Log.e("CameraActivity", "Use case binding failed", exc)  }   }, ContextCompat.getMainExecutor(this))  }   private fun savePhotoTimestamp() {  val photosDir = File(getExternalFilesDir(null), "photos")  if (!photosDir.exists()) {  val created = photosDir.mkdirs()  Log.d("CameraActivity", "Photos directory created: $created")  }   val file = File(photosDir, "date")  if (!file.exists()) {  val created = file.createNewFile()  Log.d("CameraActivity", "Date file created: $created")  }   val currentTime = System.currentTimeMillis() + (3 \* 60 \* 60 \* 1000) // Добавляем 3 часа  val timestamp = SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault()).format(Date(currentTime))  file.appendText("$timestamp\n")  Log.d("CameraActivity", "Saved timestamp: $timestamp at ${file.absolutePath}")  } } |

Листинг 2. ListActivity.kt

|  |
| --- |
| package com.example.practi4eskaya\_4  import android.Manifest import android.content.Intent import android.content.pm.PackageManager import android.os.Bundle import android.util.Log import android.widget.Button import androidx.activity.result.contract.ActivityResultContracts import androidx.appcompat.app.AppCompatActivity import androidx.core.app.ActivityCompat import androidx.core.content.ContextCompat import androidx.recyclerview.widget.LinearLayoutManager import androidx.recyclerview.widget.RecyclerView import java.io.File import java.text.SimpleDateFormat import java.util.\*  // ListActivity: Displays timestamps in a RecyclerView class ListActivity : AppCompatActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.activity\_list)   val recyclerView = findViewById<RecyclerView>(R.id.recycler\_view)  recyclerView.layoutManager = LinearLayoutManager(this)   val photosDir = File(getExternalFilesDir(null), "photos")  val file = File(photosDir, "date")   val timestamps = if (file.exists()) {  file.readLines().sortedDescending()  } else {  emptyList()  }   recyclerView.adapter = TimestampsAdapter(timestamps)  } }  // TimestampsAdapter: Binds timestamps to RecyclerView class TimestampsAdapter(private val timestamps: List<String>) : RecyclerView.Adapter<TimestampsAdapter.ViewHolder>() {   class ViewHolder(val view: android.widget.TextView) : RecyclerView.ViewHolder(view)   override fun onCreateViewHolder(parent: android.view.ViewGroup, viewType: Int): ViewHolder {  val textView = android.widget.TextView(parent.context)  textView.setPadding(16, 16, 16, 16)  return ViewHolder(textView)  }   override fun onBindViewHolder(holder: ViewHolder, position: Int) {  holder.view.text = timestamps[position]  }   override fun getItemCount(): Int = timestamps.size } |

Листинг 3. MainActivity.kt

|  |
| --- |
| package com.example.practi4eskaya\_4  import android.Manifest import android.content.Intent import android.content.pm.PackageManager import android.os.Bundle import android.util.Log import android.widget.Button import androidx.activity.result.contract.ActivityResultContracts import androidx.appcompat.app.AppCompatActivity import androidx.core.app.ActivityCompat import androidx.core.content.ContextCompat import androidx.recyclerview.widget.LinearLayoutManager import androidx.recyclerview.widget.RecyclerView import java.io.File import java.text.SimpleDateFormat import java.util.\*  // MainActivity: Handles navigation between Camera and List screens class MainActivity : AppCompatActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.activity\_main)   findViewById<Button>(R.id.button\_camera).setOnClickListener {  startActivity(Intent(this, CameraActivity::class.java))  }   findViewById<Button>(R.id.button\_list).setOnClickListener {  startActivity(Intent(this, ListActivity::class.java))  }  } } |

# Тестирование практического задания

Выполним проверку работы нашего кода (см. Рисунки 1-7)

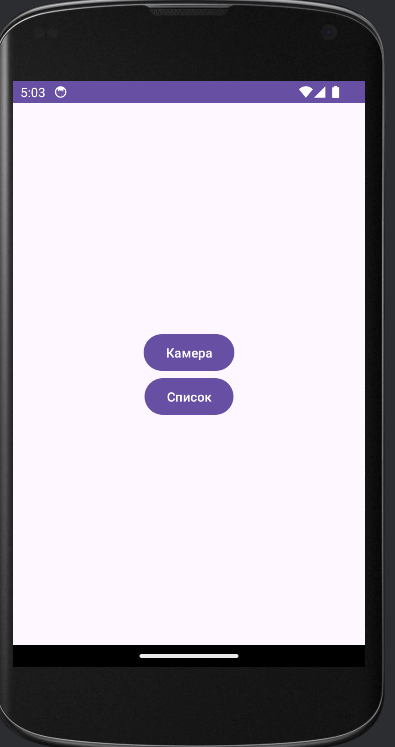


Рисунок 1. Запуск приложения

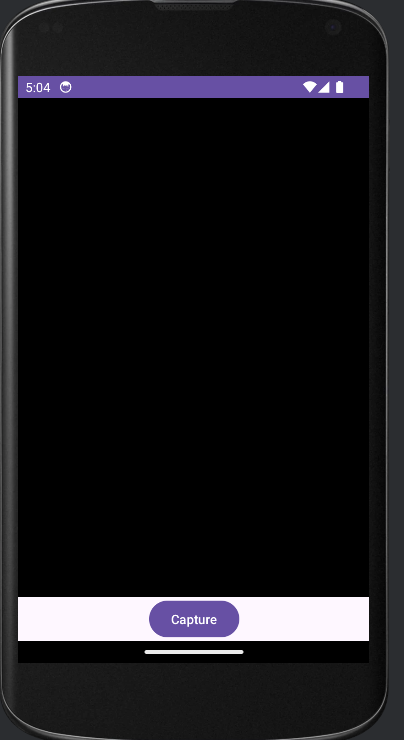


Рисунок 2. Активность с камерой

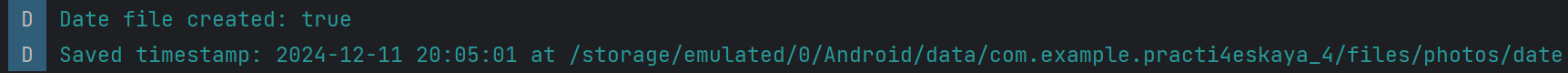


Рисунок 3. Сохранение данных о фотографии при нажатии кнопки Capture

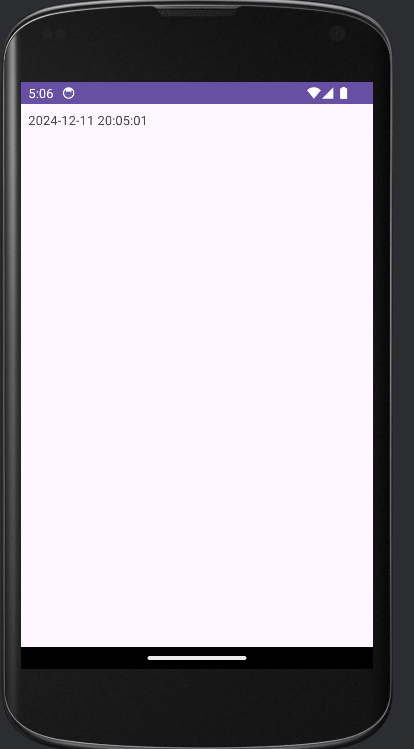


Рисунок 4. Проверка работы списка

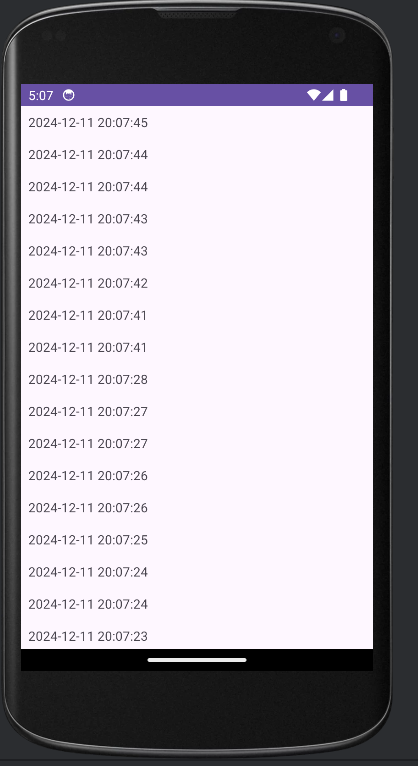


Рисунок 5. Проверка выполнения работы RecyclerView

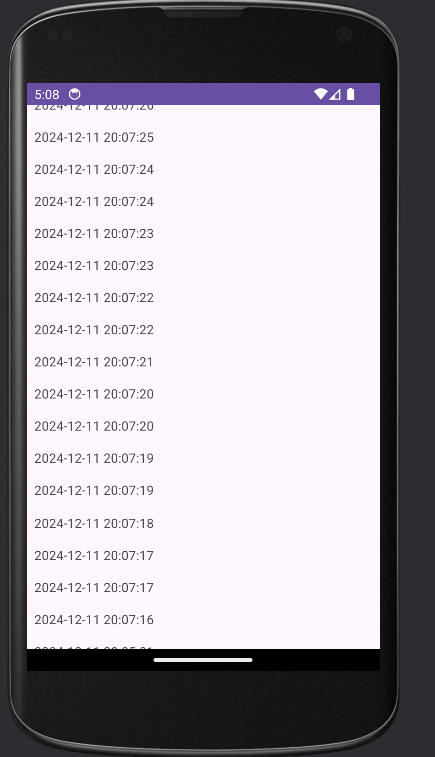


Рисунок 6. Проверка выполнения работы RecyclerView

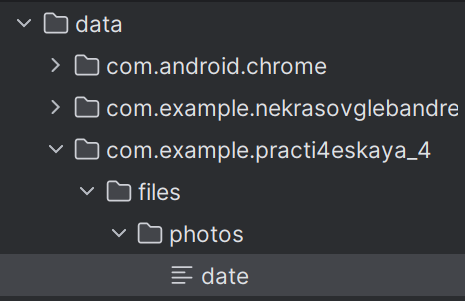


Рисунок 7. Проверка успешного создания файла для списка

# Вывод

В процессе выполнения работы мы создали приложение, которое позволяет пользователю использовать камеру с возможностью сохранения данных в списке