Разработка мобильных приложений

Практическая работа №5

Выполнил: Алеев А.В. БСБО-07-22

В ходе данной работы был создан проект «ru.mirea.AleevAV.Lesson5», в котором были созданы следующие модули: «accelerometer», «audiorecord», «camera» и «Sensor» (см. рис .1).

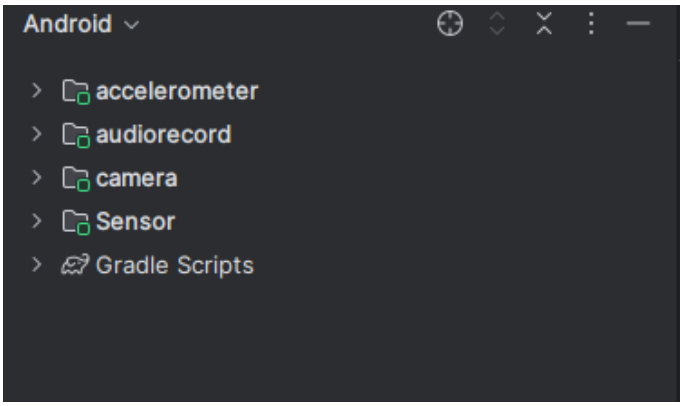

Рисунок 1. Модули проекта

В первом модуле «Sensor» был создан экран, в котором показываются все датчики телефона (см. рис. 2 и листинг 1).

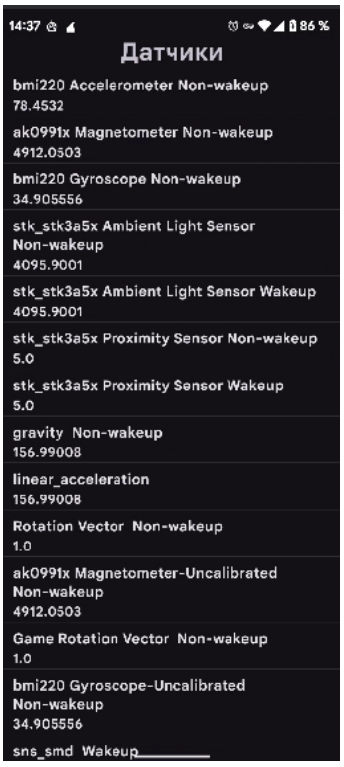

Рисунок 2. Список датчиков

```
public class MainActivity extends AppCompatActivity {
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
            Insets systemBars =
```

```
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });

        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        SensorManager sensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
        List<Sensor> sensors = sensorManager.getSensorList(Sensor.TYPE_ALL);
        ListView listSensor = binding.LVSensors;

        ArrayList<HashMap<String, Object>> arrayList = new ArrayList<>();
        for (int i = 0; i <sensors.size(); i++){
            HashMap<String, Object> sensorTypeList = new HashMap<>();
            sensorTypeList.put("Name", sensors.get(i).getName());
            sensorTypeList.put("Value", sensors.get(i).getMaximumRange());
            arrayList.add(sensorTypeList);
        }

        SimpleAdapter mHistory =
                new SimpleAdapter(this, arrayList,
android.R.layout.simple_list_item_2,
                        new String[]{"Name","Value"},
                        new int[]{android.R.id.text1, android.R.id.text2});
        listSensor.setAdapter(mHistory);

    }
}
```

Листинг 1. Класс для показа датчиков

Далее был создан модуль «Accelerometer» в котором был реализован экран с 3-мя текстовыми полями, в которых показываются данные с акселерометра (см. рис. 3 и листинг 2).

Рисунок 3. Данные с акселерометра

```
public class MainActivity extends AppCompatActivity implements
SensorEventListener {
    private ActivityMainBinding binding;
    private SensorManager sensorManager;
    private Sensor accelerometer;
    private TextView azimuthTextView;
    private TextView pitchTextView;
    private TextView rollTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });

        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
        accelerometer =
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        sensorManager.registerListener(this, accelerometer,
SensorManager.SENSOR_DELAY_NORMAL);
        azimuthTextView = binding.TextAzimuth;
        pitchTextView = binding.textPitch;
        rollTextView = binding.textRoll;

    }

    @Override
    protected void onPause(){
        super.onPause();
        sensorManager.unregisterListener(this);
    }

    @Override
    protected void onResume(){
        super.onResume();
        sensorManager.registerListener(this, accelerometer,
SensorManager.SENSOR_DELAY_NORMAL);

    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
            float x = event.values[0];
            float y = event.values[1];
            float z = event.values[2];
            azimuthTextView.setText("Azimuth: " + x);
            pitchTextView.setText("Pitch: " + y);
            rollTextView.setText("Roll: " + z);
        }

    }
```

```
    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {

    }
}
```

Листинг 2. Класс экрана с данными

Далее был создан модуль «camera», в котором был реализован вызов системного приложения «камера», сохранение изображения в папку приложения и отображение снимка на экране. Камера вызывается при нажатии на изображение (см. рис.4 и листинг 3).

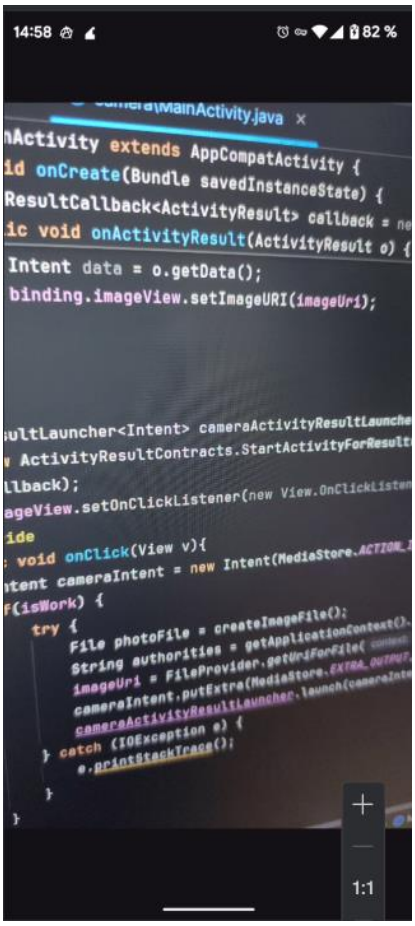

Рисунок 4. Пример работы модуля

```
public class MainActivity extends AppCompatActivity {
    private ActivityMainBinding binding;
    private static final int REQUEST_CODE_PERMISSION = 100;
    private static final int CAMERA_REQUEST = 0;
    private Uri imageUri;
    private boolean isWork;



    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
```

```
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });

        int CameraPermissionStatus = ContextCompat.checkSelfPermission(this,
android.Manifest.permission.CAMERA);
        int storagePermissionStatus = ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE); // на android 10+ не работает
        if(CameraPermissionStatus == PackageManager.PERMISSION_GRANTED){
            isWork = true;
        } else {
            ActivityCompat.requestPermissions(this, new String[]
{android.Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE}, REQUEST_CODE_PERMISSION);
        }


        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        ActivityResultCallback<ActivityResult> callback = new
ActivityResultCallback<ActivityResult>() {
            @Override
            public void onActivityResult(ActivityResult o) {
                if(o.getResultCode() == RESULT_OK){
                    Intent data = o.getData();
                    binding.imageView.setImageURI(imageUri);

                }
            }
        };
        ActivityResultLauncher<Intent> cameraActivityResultLauncher =
registerForActivityResult(
                new ActivityResultContracts.StartActivityForResult(),
                callback);
        binding.imageView.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v){
                Intent cameraIntent = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                if(isWork) {
                    try {
                        File photoFile = createImageFile();
                        String authorities =
getApplicationContext().getPackageName() + ".fileprovider";
                        imageUri =
FileProvider.getUriForFile(MainActivity.this, authorities, photoFile);
                        cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT,
imageUri);
                        cameraActivityResultLauncher.launch(cameraIntent);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }
            }
        });

    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
```

```
permissions, @NonNull int[] grantResults){
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        if(requestCode == REQUEST_CODE_PERMISSION){
            isWork = grantResults.length> 1 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED && grantResults[1] ==
PackageManager.PERMISSION_GRANTED;
            if(!isWork){
                Toast.makeText(this, "Permission denied",
Toast.LENGTH_SHORT).show();
            }
        }
    }
    private File createImageFile() throws IOException{
        String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss",
Locale.ENGLISH).format(new Date());
        String imageFileName = "IMAGE_" + timeStamp + "_";

        File storageDirectory =
getExternalFilesDir(Environment.DIRECTORY_PICTURES);
        return File.createTempFile(imageFileName, ".jpg", storageDirectory);

    }

}
```

Листинг 3. Класс для работы с камерой

Далее был создан модуль «AudioRecord» для работы с диктофоном. При нажатии на кнопку «Запись / стоп» начинается запись, если она не была включена. Затем при нажатии на «воспроизведение» начинается воспроизведение записи (см. рис. 5 и листинг 4).

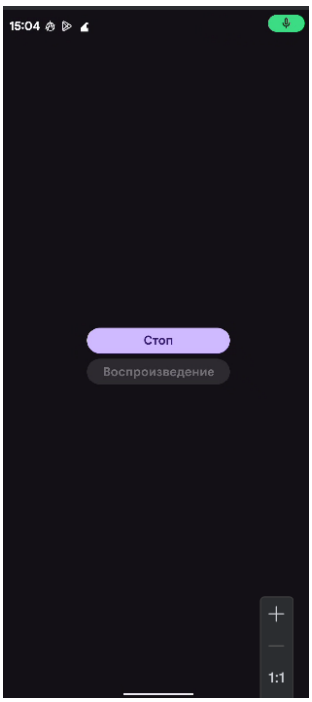

Рисунок 5. Пример работы модуля

```
public class MainActivity extends AppCompatActivity {
    private ActivityMainBinding binding;
    private static final int REQUEST_CODE_PERMISSION = 200;
    //private final String TAG = MainActivity.class.getSimpleName();
```

```
    private boolean isWork;
    private String fileName = null;
    private Button recordButton = null;
    private Button playButton = null;
    private MediaRecorder recorder = null;
    private MediaPlayer player = null;
    boolean isStartRecord = true;
    boolean isStartPlay = true;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });

        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        recordButton =binding.RecordButton;
        playButton = binding.PlayButton;
        playButton.setEnabled(false);

        recordButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(isStartRecord){
                    recordButton.setText("Стоп");
                    playButton.setEnabled(false);
                    startRecord();
                } else{
                    recordButton.setText("Запись");
                    playButton.setEnabled(true);
                    stopRecord();
                }
                isStartRecord = !isStartRecord;

            }
        });

        playButton.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v){
                if(isStartPlay){
                    playButton.setText("Стоп");
                    recordButton.setEnabled(false);
                    startPlay();
                } else {
                    playButton.setText("Воспроизведение");
                    recordButton.setEnabled(true);
                    stopPlay();
                }
                isStartPlay = !isStartPlay;
            }
        });
```

```
        int audioRecordPermissionStatus =
ContextCompat.checkSelfPermission(this, Manifest.permission.RECORD_AUDIO);
        if(audioRecordPermissionStatus == PackageManager.PERMISSION_GRANTED){
            isWork = true;
        } else {
            ActivityCompat.requestPermissions(this, new String[]
{Manifest.permission.RECORD_AUDIO}, REQUEST_CODE_PERMISSION);
        }
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults){
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
        switch(requestCode){
            case REQUEST_CODE_PERMISSION:
                isWork = grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
                break;
        }
        if(!isWork){
            finish();
        }
    }

    private void startRecord(){
        fileName = getExternalFilesDir(null).getAbsolutePath() +
"/audiorecord.3gp";
        recorder = new MediaRecorder();
        recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
        recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
        recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
        recorder.setOutputFile(fileName);
        try{
            recorder.prepare();
            recorder.start();
        } catch (Exception e) {
            e.printStackTrace();
        }


    }

    private void stopRecord(){
        recorder.stop();
        recorder.release();
        recorder = null;
    }

    private void startPlay() {
        player = new MediaPlayer();
        try {
            player.setDataSource(fileName);
            player.prepare();
            player.start();
        } catch (Exception e) {
        }
    }

    private void stopPlay() {
```

```
        player.release();
        player = null;
    }

}
```

Листинг 4. Класс для работы с микрофоном

Далее в проекте «MireaProject» были добавлены 3 фрагмента, а также был реализован механизм запроса разрешений.

В первом фрагменте был реализован компас, при помощи датчиков (см. рис. 6 и листинг 5).

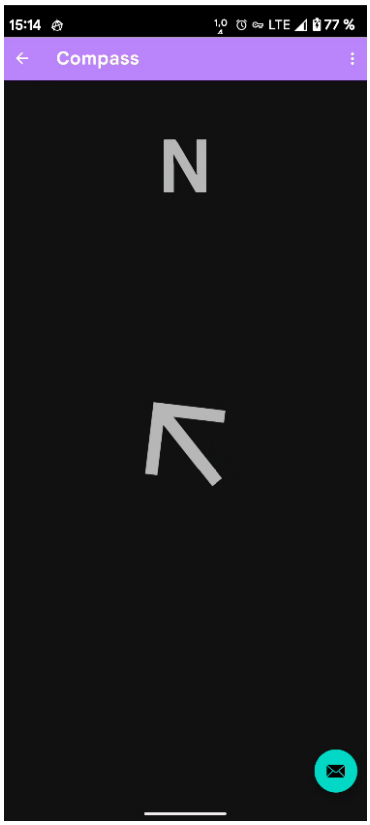

Рисунок 6. Компас

```
public class CompassFragment extends Fragment implements SensorEventListener
{
    private FragmentCompassBinding binding;
    private SensorManager sensorManager;
    private Sensor accelerometer, magnetometer;

    private float[] accelReadings = new float[3];
    private float[] magnetReadings = new float[3];
    private float currentAzimuth = 0f;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                             @Nullable ViewGroup container,
                             @Nullable Bundle savedInstanceState) {
        binding = FragmentCompassBinding.inflate(inflater, container, false);
        View root = binding.getRoot();
        ViewCompat.setOnApplyWindowInsetsListener(root, (v, insets) -> {
            Insets sys =
```

```
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(sys.left, sys.top, sys.right, sys.bottom);
            return insets;
        });
        return root;
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        sensorManager =
requireActivity().getSystemService(SensorManager.class);
        accelerometer =
sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        magnetometer  =
sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
    }

    @Override
    public void onResume() {
        super.onResume();
        sensorManager.registerListener(this, accelerometer,
SensorManager.SENSOR_DELAY_UI);
        sensorManager.registerListener(this, magnetometer,
SensorManager.SENSOR_DELAY_UI);
    }

    @Override
    public void onPause() {
        super.onPause();
        sensorManager.unregisterListener(this);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
            System.arraycopy(event.values, 0, accelReadings, 0,
accelReadings.length);
        } else if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD) {
            System.arraycopy(event.values, 0, magnetReadings, 0,
magnetReadings.length);
        }

        float[] R = new float[9];
        float[] I = new float[9];
        if (SensorManager.getRotationMatrix(R, I, accelReadings,
magnetReadings)) {
            float[] orientation = new float[3];
            SensorManager.getOrientation(R, orientation);
            float azimuth = (float) Math.toDegrees(orientation[0]);

            RotateAnimation anim = new RotateAnimation(
                    -currentAzimuth,
                    -azimuth,
                    Animation.RELATIVE_TO_SELF, 0.5f,
                    Animation.RELATIVE_TO_SELF, 0.5f
            );
            anim.setDuration(250);
            anim.setFillAfter(true);
            binding.imageView3.startAnimation(anim);
            currentAzimuth = azimuth;
        }
    }
```

```
    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
        // No-op
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
```

Листинг 5. Класс для работы компаса

Во втором фрагменте был реализован экран для камеры. В нём можно сделать снимок и написать текст (см. рис.7 и листинг 6).

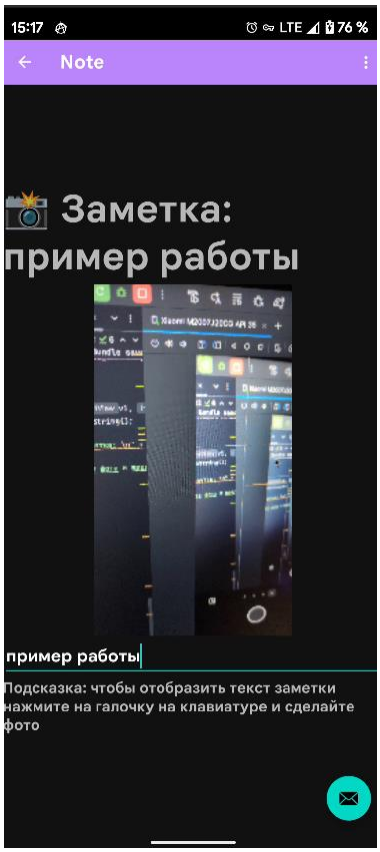

Рисунок 7. Пример работы заметки

```
public class NoteFragment extends Fragment {

    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    private FragmentNoteBinding binding;
    private Bitmap capturePhoto;
    private ActivityResultLauncher<Intent> cameraLauncher;

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;
```

```
    public NoteFragment() {
        // Required empty public constructor


    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment NoteFragment.
     */
    // TODO: Rename and change types and number of parameters
    public static NoteFragment newInstance(String param1, String param2) {
        NoteFragment fragment = new NoteFragment();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        binding = FragmentNoteBinding.inflate(inflater, container, false);
        return binding.getRoot();
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState){
        super.onViewCreated(view, savedInstanceState);
        if(ContextCompat.checkSelfPermission(requireContext(),
Manifest.permission.CAMERA)
                != PackageManager.PERMISSION_GRANTED){
            ActivityCompat.requestPermissions(requireActivity(), new
String[]{Manifest.permission.CAMERA}, 100);


        }
        cameraLauncher = registerForActivityResult(
                new ActivityResultContracts.StartActivityForResult(),
                new ActivityResultCallback<ActivityResult>() {
                    @Override
                    public void onActivityResult(ActivityResult o) {
                        if(o.getResultCode() == Activity.RESULT_OK &&
o.getData() != null){
                            Bundle extras = o.getData().getExtras();
                            capturePhoto = (Bitmap) extras.get("data");
                            binding.imageView4.setImageBitmap(capturePhoto);
                    }
                }
```

```
        });

        binding.imageView4.setOnClickListener(v -> {
            Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            cameraLauncher.launch(intent);
        });

        binding.editTextText.setOnEditorActionListener((v1, actionId, event)
->{
            String text = binding.editTextText.getText().toString();
            if (capturePhoto != null && !text.isEmpty()){
                binding.textView4.setText("\uD83D\uDCF8 Заметка: \n" + text);
            }else {
                binding.textView4.setText("Сначала сделайте фото и введите
текст");
            }
            return true;
        });
    }

    @Override
    public void onDestroyView(){
        super.onDestroyView();
        binding = null;
    }



}
```

Листинг 6. Код для работы заметки

Затем был сделан третий фрагмент, в котором был реализован функционал диктофона (см. рис. 8 и листинг 7).

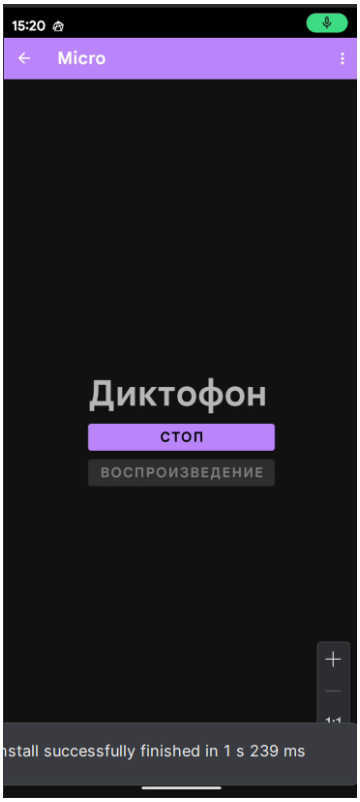

Рисунок 8. Диктофон

```
public class MicroFragment extends Fragment {
    private FragmentMicroBinding binding;
    private static final int REQUEST_CODE_PERMISSION = 200;

    private boolean isWork = false;
    private String fileName = null;
    private MediaRecorder recorder = null;
    private MediaPlayer player = null;
    private boolean isStartRecord = true;
    private boolean isStartPlay = true;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container, @Nullable Bundle savedInstanceState) {
        binding = FragmentMicroBinding.inflate(inflater, container, false);
        return binding.getRoot();
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        fileName =
requireContext().getExternalFilesDir(null).getAbsolutePath() +
"/audiorecord.3gp";

        int audioPermissionStatus =
ContextCompat.checkSelfPermission(requireContext(),
Manifest.permission.RECORD_AUDIO);
        if (audioPermissionStatus == PackageManager.PERMISSION_GRANTED) {
            isWork = true;
        } else {
            ActivityCompat.requestPermissions(requireActivity(), new
String[]{Manifest.permission.RECORD_AUDIO}, REQUEST_CODE_PERMISSION);
        }

        binding.RecordButton.setOnClickListener(v -> {
            if (isStartRecord) {
                startRecord();
                binding.RecordButton.setText("Стоп");
                binding.PlayButton.setEnabled(false);
            } else {
                stopRecord();
                binding.RecordButton.setText("Запись");
                binding.PlayButton.setEnabled(true);
            }
            isStartRecord = !isStartRecord;
        });

        binding.PlayButton.setOnClickListener(v -> {
            if (isStartPlay) {
                startPlay();
                binding.PlayButton.setText("Стоп");
                binding.RecordButton.setEnabled(false);
            } else {
                stopPlay();
                binding.PlayButton.setText("Воспроизведение");
                binding.RecordButton.setEnabled(true);
            }
            isStartPlay = !isStartPlay;
        });
    }
```

```
    private void startRecord() {
        recorder = new MediaRecorder();
        recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
        recorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
        recorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
        recorder.setOutputFile(fileName);
        try {
            recorder.prepare();
            recorder.start();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void stopRecord() {
        if (recorder != null) {
            recorder.stop();
            recorder.release();
            recorder = null;
        }
    }

    private void startPlay() {
        player = new MediaPlayer();
        try {
            player.setDataSource(fileName);
            player.prepare();
            player.start();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void stopPlay() {
        if (player != null) {
            player.release();
            player = null;
        }
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
```

Листинг 7. Код для работы диктофона