

Разработка мобильных приложений

Практическая работа №2

Выполнил: Алеев А.В. БСБО-07-22

В ходе данной работы было созданы следующие модули: «activitylifecycle», «multiactivity», «intentfilter», «toastapp», «notificationapp», «dialog» (см. рис. 1).

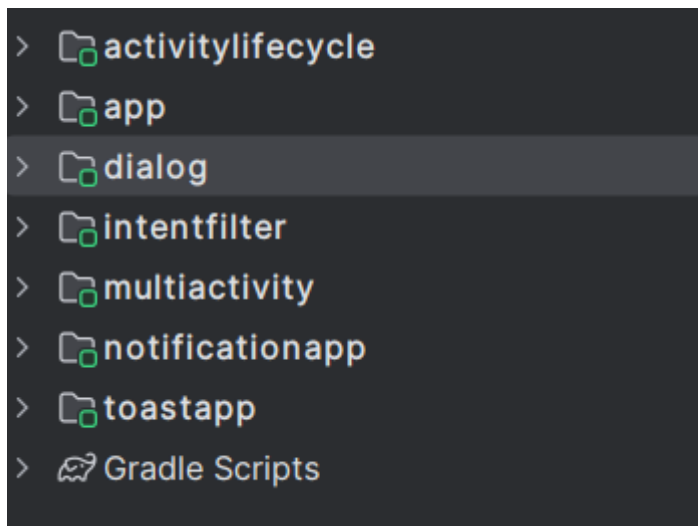


Рисунок 1. 6 созданных в ходе работы модулей

В модуле «activitylifecycle» были переопределены основные методы жизненного цикла родительского класса для просмотра жизненного цикла активности (см. листинг 1).

```
private static final String TAG = "ActivityLifeCycle";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.d(TAG, "метод onCreate сработал!");
}

@Override
protected void onStart() {
    super.onStart();
    Log.d(TAG, "метод onStart сработал!");
}

@Override
protected void onResume() {
    super.onResume();
    Log.d(TAG, "Метод onResume сработал!");
}

@Override
protected void onStop() {
    super.onStop();
    Log.d(TAG, "Метод onStop сработал!");
}

@Override
protected void onRestart() {
    super.onRestart();
}
```

```

        Log.d(TAG, "Метод onRestart Сработал!");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(TAG, "метод onDestroy Сработал!");
    }

```

Листинг 1. Перезаписанные методы

После запуска приложения в «logcat» были вызваны следующие методы «onCreate», «onStart» и «onResume». После нажатия на кнопку «Home» вызвался метод «onStop». Затем после возвращения в приложение вызвались методы «onRestart» «onStart» и «onResume» (см. рис. 2). Введённый текст в поле остался.

2025-04-29 15:17:22.435	6655-6655	ActivityLifecycle	ru.mirea.aaleev.activitylifecycle	D метод onCreate сработал!
2025-04-29 15:17:22.444	6655-6655	ActivityLifecycle	ru.mirea.aaleev.activitylifecycle	D метод onStart сработал!
2025-04-29 15:17:22.448	6655-6655	ActivityLifecycle	ru.mirea.aaleev.activitylifecycle	D Метод onResume Сработал!
2025-04-29 15:19:38.589	6655-6655	ActivityLifecycle	ru.mirea.aaleev.activitylifecycle	D Метод onStop Сработал!
2025-04-29 15:20:23.948	6655-6655	ActivityLifecycle	ru.mirea.aaleev.activitylifecycle	D Метод onRestart Сработал!
2025-04-29 15:20:23.948	6655-6655	ActivityLifecycle	ru.mirea.aaleev.activitylifecycle	D метод onStart сработал!
2025-04-29 15:20:23.949	6655-6655	ActivityLifecycle	ru.mirea.aaleev.activitylifecycle	D Метод onResume Сработал!

Рисунок 2. Вызванные методы

В модуле «MultiActivityh» был реализован переход от одной активности в другую при помощи кнопок и фильтра «intent». После ввода данных и нажатия на кнопку «Отправить» открывается другая активность, в которой видно введённый текст (см. рис 3).



Рисунок 3. Вторая активность

При переходе на новую активность в ней вызываются те же методы, что и при открытии первой, но в первой вызывается метод «onStop». При нажатии на кнопку «Back» Уничтожается вторая активность и в первой активности сохраняется текст. При очередном нажатии на кнопку «Back» приложение закрывается и удаляется активность (см. рис. 4).

```
----- PROCESS STARTED (10015) for package ru.mirea.AleevAV.multiactivity -----
2025-04-29 15:33:53.457 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onCreate сработал!
2025-04-29 15:33:53.465 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onStart сработал!
2025-04-29 15:33:53.468 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onResume сработал!
2025-04-29 15:34:05.272 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onCreate сработал! (Second Activity)
2025-04-29 15:34:05.276 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onStart сработал! (Second Activity)
2025-04-29 15:34:05.278 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onResume сработал! (Second Activity)
2025-04-29 15:34:05.932 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onStop сработал!
2025-04-29 15:34:27.476 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onRestart сработал!
2025-04-29 15:34:27.477 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onStart сработал!
2025-04-29 15:34:27.478 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onResume сработал!
2025-04-29 15:34:28.159 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onStop сработал! (Second Activity)
2025-04-29 15:34:28.162 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onDestroy сработал! (Second Activity)
2025-04-29 15:34:55.656 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onStop сработал!
2025-04-29 15:34:55.660 10015-10015 ActivityLifecycle ru.mirea.AleevAV.multiactivity D метод onDestroy сработал!
```

Рисунок 4. Логи второго модуля

Передача текста с поля производится с помощью метода «putExtra». Затем во второй активности возвращается значение через метод «getIntent» и передается в метод «setText» (см. листинг 2).

```
String resieveddata = (String) getIntent().getSerializableExtra("key");
TextView textSecond = findViewById(R.id.TextSecond);
textSecond.setText(resieveddata);
```

Листинг 2. Передача текста

Также был создан модуль «IntentFilter» (см. рис. 4).

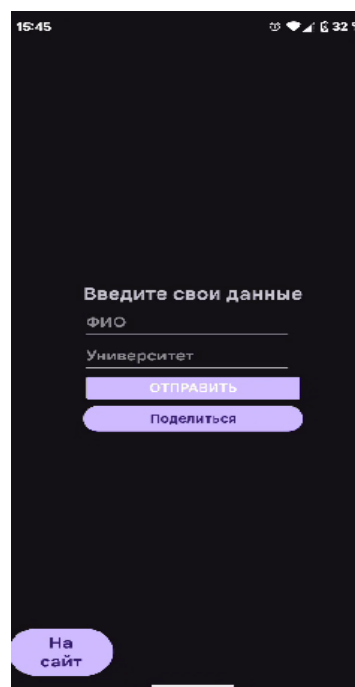


Рисунок 4. Модуль «IntentFilter»

В нём была создана кнопка, перенаправляющая на сайт «РТУ МИРЭА» (см. листинг 3).

```
public void OnClickURL(View view){
    Uri address = Uri.parse("https://www.mirea.ru/");
    Intent openLink = new Intent(Intent.ACTION_VIEW, address);
    startActivity(openLink);
}
```

Листинг 3. Метод кнопки «На сайт»

Затем были добавлены два поля для ввода и две кнопки: «Отправить» и «Поделиться». Первая кнопка работает по аналогичному принципу из прошлого модуля, а вторая – открывает окно «Поделиться текстом» (см. листинг 4).

```
public void OnClickShare(View view){
    EditText textFIO = findViewById(R.id.editTextFIO);
    EditText textUniver = findViewById(R.id.editTextUniver);

    Intent shareIntent = new Intent(Intent.ACTION_SEND);
    shareIntent.setType("text/plain");
    shareIntent.putExtra(Intent.EXTRA_SUBJECT, "MIREA");
    shareIntent.putExtra(Intent.EXTRA_TEXT, "ФИО: " +
textFIO.getText().toString()+ " | Универ: " +
textUniver.getText().toString());
    startActivity(Intent.createChooser(shareIntent, "МОИ ФИО и название
универа"));
}
```

Листинг 4. Метод кнопки «Поделиться»

Далее было создано несколько модулей для работы с диалоговыми окнами. Первым модулем стал «ToatApp» (см. рис. 5).

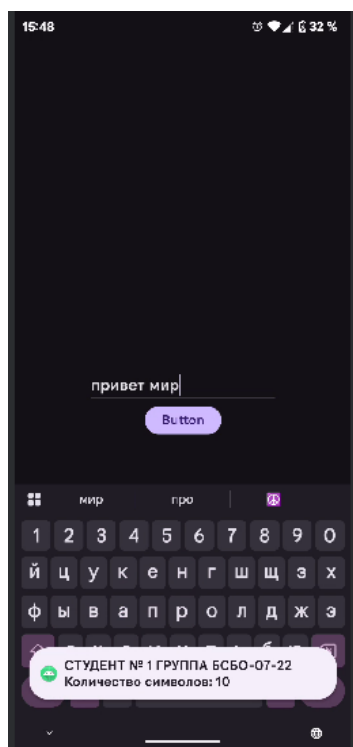


Рисунок 5. Пример работы модуля

При нажатии на кнопку вызывается метод, отображающий уведомление (см. листинг 5).

```
public void OnClickToast(View view){
    EditText text = findViewById(R.id.editTextText);
    Toast toast = Toast.makeText(getApplicationContext(),
        "СТУДЕНТ № 1 ГРУППА БСБО-07-22 Количество символов: " +
        text.getText().length(),
        Toast.LENGTH_SHORT);
    toast.show();
}
```

Листинг 5. Метод кнопки

Был создан ещё один модуль «NotificationApp, вызывающий уведомление при нажатии на кнопку (см. рис. 6).

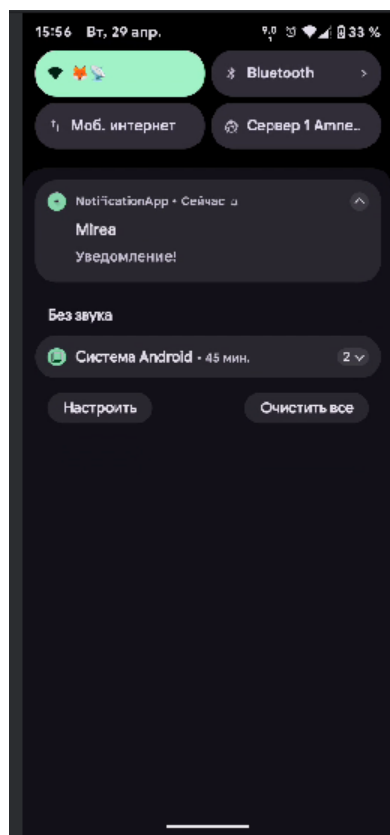


Рисунок 6. Уведомление

В методе кнопки определяется название и текст уведомления (см. листинг 6).

```
public void OnClickSendNotification(View view){

    if (ActivityCompat.checkSelfPermission(this, POST_NOTIFICATIONS) !=
        PackageManager.PERMISSION_GRANTED) {
        return;
    }

    NotificationCompat.Builder builder = new NotificationCompat.Builder(this,
        CHANNEL_ID)
        .setContentText("Congratulation!")
        .setSmallIcon(R.drawable.ic_launcher_foreground)
```

```

        .setPriority(NotificationCompat.PRIORITY_HIGH)
        .setStyle(new NotificationCompat.BigTextStyle()
            .bigText("Уведомление!"))
        .setContentTitle("Mirea");

    int importance = NotificationManager.IMPORTANCE_DEFAULT;
    NotificationChannel channel = null;
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        channel = new NotificationChannel(CHANNEL_ID, "Student FIO
Notification", importance);
    }
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        channel.setDescription("MIREA Channel");
    }

    NotificationManagerCompat notificationManager =
NotificationManagerCompat.from(this);
    notificationManager.createNotificationChannel(channel);
    notificationManager.notify(1, builder.build());
}

```

Листинг 6. Настройка уведомления

Наконец был создан модуль «Dialog» с один текстом и 4 кнопками (см. рис. 7)

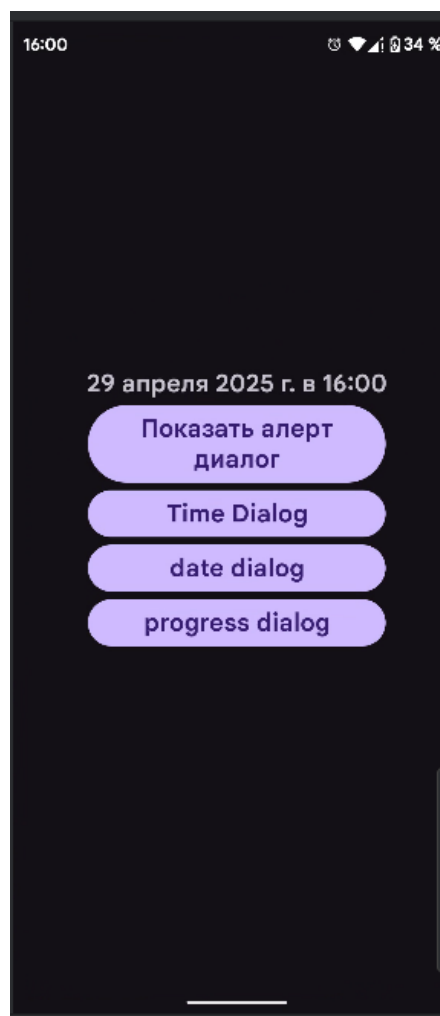


Рисунок 7. Модуль «Dialog»

Первая кнопка вызывает алерт диалог, в котором после нажатия на кнопки вызывается всплывающее уведомление (см. рис. 8 и листинг 7).

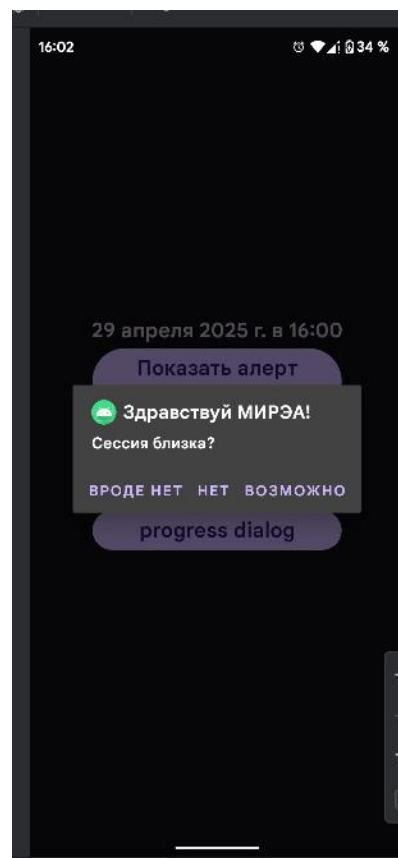


Рисунок 8. Алерт диалог

```
public class AlertDialogFragment extends DialogFragment {

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setTitle("Здравствуй МИРЭА!")
            .setMessage("Сессия близка?")
            .setIcon(R.mipmap.ic_launcher_round)
            .setPositiveButton("ВОЗМОЖНО", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    ((MainActivity)getActivity()).onOkClicked();
                    dialog.cancel();
                }
            })
            .setNeutralButton("вроде нет", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    ((MainActivity)getActivity()).onNeutralClicked();
                    dialog.cancel();
                }
            })
            .setNegativeButton("Нет", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    ((MainActivity)getActivity()).onCancelClicked();
                    dialog.cancel();
                }
            })
    }
}
```



```

    }
    });
    return builder.create();
}
}

```

Листинг 7. Класс «AlertDialogFragment»

Вторая кнопка вызывает диалог, в котором можно настраивать время (см. рис.9 и листинг 8).

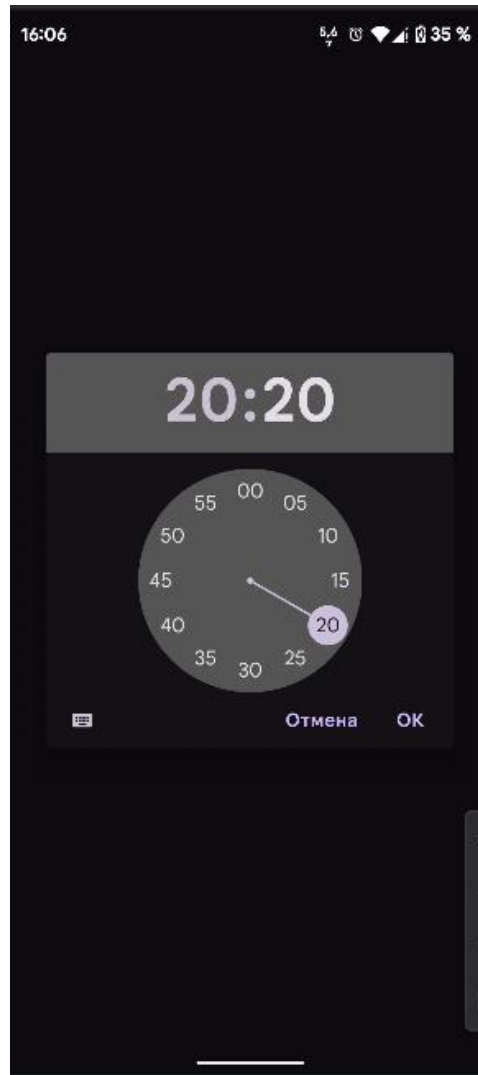


Рисунок 9. Настройка времени

```

class Timerclass extends TimePickerDialog {
    private final Calendar initialTime;
    private final OnTimeSetListener externalListener;

    public Timerclass(Context context, Calendar initialTime,
        OnTimeSetListener listener) {
        super(context, (TimePicker, hourOfDay, minute) -> {
            if (listener != null) {
                listener.onTimeSet(TimePicker, hourOfDay, minute);
                initialTime.set(Calendar.HOUR_OF_DAY, hourOfDay);
                initialTime.set(Calendar.MINUTE, minute);
            }
        },
        initialTime.get(Calendar.HOUR_OF_DAY),

```

```

        initialTime.get(Calendar.MINUTE),
        DateFormat.is24HourFormat(context));

        this.initialTime = initialTime;
        this.externalListener = listener;
    }

    @Override
    public void onTimeChanged(TimePicker timePicker, int hourOfDay, int
minute) {
        super.onTimeChanged(timePicker, hourOfDay, minute);
    }
}

```

Листинг 8. Класс «TimerClass»

Третья кнопка вызывает диалог с выбором даты (см. рис 10 и листинг 9).

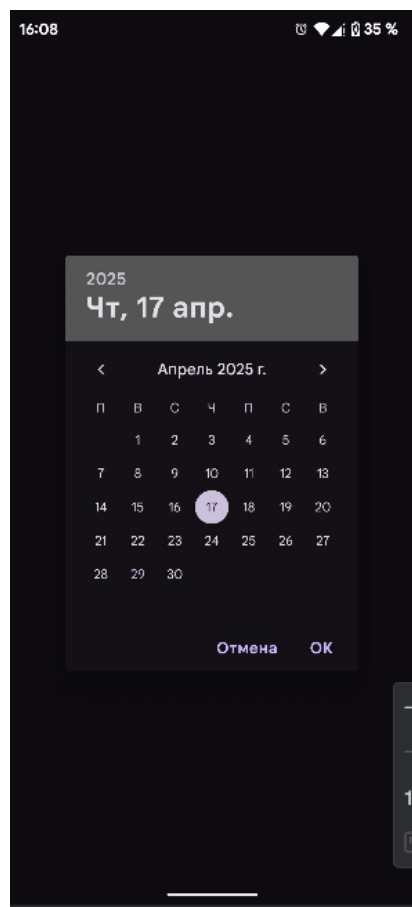


Рисунок 10. Выбор даты

```

public class DateClass extends DatePickerDialog {
    private final Calendar initialDate;

    public DateClass(Context context, Calendar initialDate, OnDateSetListener
listener) {
        super(context, (datePicker, year, month, day) -> { // Исправленное
имя параметра
            if (listener != null) {
                listener.onDateSet(datePicker, year, month, day);
                initialDate.set(Calendar.YEAR, year);
                initialDate.set(Calendar.MONTH, month);
                initialDate.set(Calendar.DAY_OF_MONTH, day);
            }
        });
    }
}

```

```

    },
    initialDate.get(Calendar.YEAR),
    initialDate.get(Calendar.MONTH),
    initialDate.get(Calendar.DAY_OF_MONTH));

    this.initialDate = initialDate;
}
}

```

Листинг 9. Класс «DateClass»

Четвёртая кнопка показывает окно загрузки, исчезающее спустя 4 секунды (см. рис. 11 и листинг 10).

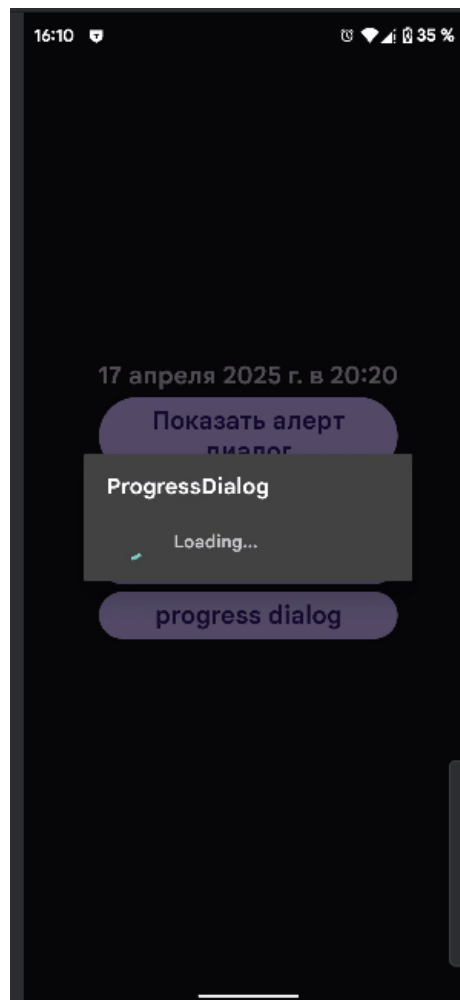


Рисунок 11. Окно загрузки

```

public class ProgressBarClass extends ProgressDialog {

    public ProgressBarClass(Context context) {
        super(context);
        configureDialog();
    }

    public ProgressBarClass(Context context, int theme) {
        super(context, theme);
        configureDialog();
    }
}

```

```
private void configureDialog() {
    setTitle("ProgressDialog");
    setMessage("Loading...");
    setCancelable(false);
}

public void showWithSafetyCheck() {
    if (!isShowing()) {
        show();
    }
}

@Override
public void dismiss() {
    if (isShowing()) {
        super.dismiss();
    }
}
}
```

Листинг 10. Класс «ProgressBarClass»