Лабораторная работа № 7

«Работа с диалоговыми окнами в Android-приложении»

Продолжительность выполнения лабораторной работы: 2 ак. часа.

Целью лабораторной работы является знакомство с принципами создания диалоговых окон на платформе Android.

Теоретическая часть

Диалог — это обычно маленькое окно, которое появляется перед текущим активити. Диалоги обычно используются для уведомлений и коротких действий, которые непосредственно касаются приложения. Обычно в диалоге пользователю предлагается принять решение или ввести дополнительную информацию.

Класс Dialog является базовым классом для диалогов, но следует избегать создания экземпляров Dialog напрямую. Вместо этого можно использовать один из следующих подклассов: AlertDialog, DatePickerDialog и TimePickerDialog.

Названные классы определяют стиль и структуру диалога, но в качестве контейнера для диалога следует использовать DialogFragment. Класс DialogFragment был добавлен в Android 3.0 (уровень API 11) и предоставляет все элементы управления, необходимые для создания диалогового окна и управления его внешним видом, вместо вызова методов объекта Dialog.

Диалоговое окно AlertDialog состоит из трех частей:

- Заглавие — является необязательным и должно использоваться только тогда, когда область содержимого занята подробным сообщением, списком или настраиваемым макетом (компоновкой). Если нужно сформулировать простое сообщение или вопрос, заголовок не используется.
- Область содержимого — может отображать сообщение, список или другой настраиваемый макет (компоновку с произвольным дизайном).
- Кнопки действий — в диалоговом окне должно быть не более трех кнопок действий (положительный, отрицательный и нейтральный ответ).

Класс AlertDialog.Builder предоставляет API-интерфейсы, которые позволяют создавать AlertDialog с этими типами содержимого.

Практическая часть

Создадим диалоговое окно путем расширения класса DialogFragment и создания AlertDialog в методе обратного вызова onCreateDialog (). В листинге 1 приведён класс, наследуемый от DialogFragment. В нём создаётся AlertDialog с двумя кнопками («Да» и «Нет»). При нажатии на кнопку положительного ответа, активити, из которого был вызван диалог, завершается, а при нажатии на кнопку отрицательного ответа закрывается диалог.

Листинг 1. Класс MyDialogFragment.

```
package com.example.dialogapp;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.os.Bundle;
import androidx.fragment.app.DialogFragment;
public class MyDialogFragment extends DialogFragment {
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());
        builder.setMessage(R.string.dialog_message);
        builder.setPositiveButton(R.string.yes, new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id)
{
                        getActivity().finish();   }        });
        builder.setNegativeButton(R.string.no, new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id)
{
                        dialog.cancel();          }        });
        return builder.create();
    }
}
```

Для создания диалогового окна сначала нужно создать объект класса Builder, передав в качестве параметра контекст приложения:

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

Метод getActivity возвращает активити, связанную с фрагментом. Класс Activity является потомком класса Context, поэтому в конструктор класса AlertDialog.Builder можно передавать активити.

Затем, используя методы класса Builder, нужно задать для создаваемого диалога необходимые свойства, например, текстовое сообщение в окне методом setMessage:

builder.setMessage(R.string.dialog_message);

С помощью метода setTitle можно задать заглавие сообщения.

После задания свойств диалога определяют командные кнопки диалога и обработку событий нажатия на них. В AlertDialog можно добавить только по одной кнопке каждого типа: Positive (положительный ответ), Negative (отрицательный ответ) и Neutral (нейтральный ответ), т. е. максимально возможное количество кнопок в диалоге — три.

Для каждой кнопки используется один из методов: setPositiveButton, setNegativeButton или setNeutralButton, которые принимают в качестве параметров надпись для кнопки, и интерфейс DialogInterface.OnClickListener, который определяет действие, когда пользователь нажимает кнопку.

Чтобы пользователь не мог закрыть диалог кнопкой Back (Назад), вызывается метод setCancelable:

builder.setCancelable(false);

После определения класса, наследуемого от DialogFragment, в классе активити нужно создать экземпляр этого класса и вызывать метод show для этого объекта, когда потребуется отобразить активити (листинг 2).

Листинг 2. Создание и вызов диалога.

```
FragmentManager manager = getSupportFragmentManager();
MyDialogFragment myDialogFragment = new MyDialogFragment();
myDialogFragment.show(manager, "myDialog");
```

На рисунке 1 приведено получившееся диалоговое окно.

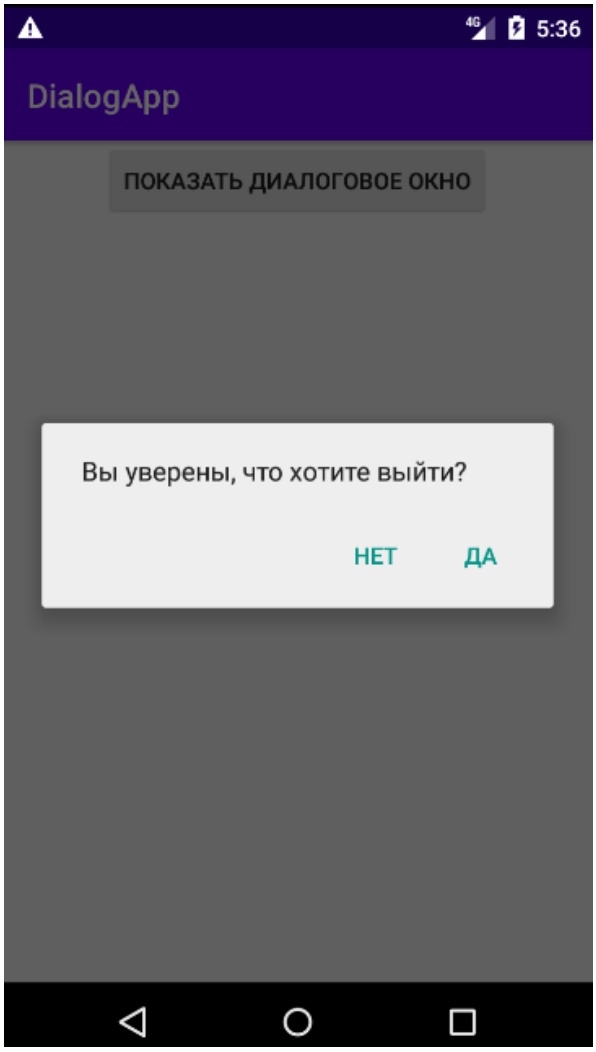

Рисунок 1. Диалоговое окно AlertDialog с двумя кнопками.

Также как при работе с Toast Notification, можно использовать компоновки для создания произвольного дизайна диалогового окна с использованием дополнительных элементов, например, изображений ImageView, полей для ввода текста EditText и т.д.

После создания компоновки необходимо передать корневой объект компоновки в код создания диалога. Чтобы получить корневой объект компоновки, используется класс LayoutInflater. Этот класс нужен для преобразования xml-компоновки в соответствующие объекты View в коде программы. Далее создается объект AlertDialog.Builder и методом setView () для него устанавливается полученная ранее компоновка.

Рассмотрим, как передать в активити информацию, введённую пользователем в диалоговом окне. В начале необходимо создать компоновку dialog_layout.xml с полем EditText с идентификатором name, а на компоновке активити добавить поле TextView с идентификатором textView.

В коде фрагмента MyDialogFragment необходимо добавить интерфейс MyDialogFragmentListener:

```
public interface MyDialogFragmentListener {
    public void onReturnValue(String value);
}
```

Далее в коде фрагмента MyDialogFragment необходимо после вызова метода inflate найти на компоновке диалогового окна данное поле для ввода текста по его идентификатору:

```
EditText myText = (EditText) layout.findViewById(R.id.name);
```

В методе onClick кнопки положительного ответа мы возвращаем результат в активити:

```
MyDialogFragmentListener activity = (MyDialogFragmentListener) getActivity();
activity.onReturnValue(String.valueOf(myText.getText()));
```

Затем в активити реализуем этот интерфейс, получаем значение и выводим его в текстовое поле myText:

```
@Override
  public void onReturnValue(String value) {
    myText.setText(value);
  }
```

Класс MyDialogFragment приведён в листинге 3. Класс MainActivity приведён в листинге 4.

Листинг 3. MyDialogFragment.

```
public class MyDialogFragment extends DialogFragment {
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());
        builder.setMessage(R.string.dialog_message);
        LayoutInflater inflater =
requireActivity().getLayoutInflater();
        View layout = inflater.inflate(R.layout.dialog_layout,null);
        builder.setView(layout);
        final EditText myText = (EditText)
layout.findViewById(R.id.name);
        builder.setPositiveButton(R.string.yes, new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                MyDialogFragmentListener activity =
(MyDialogFragmentListener) getActivity();

activity.onReturnValue(String.valueOf(myText.getText()));
            }
        });
        builder.setNegativeButton(R.string.no, new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }
        });
        return builder.create();
    }
    public interface MyDialogFragmentListener {
        void onReturnValue(String value);
    }
}
```

Листинг 4. MainActivity.

```
public class MainActivity extends AppCompatActivity implements
MyDialogFragment.MyDialogFragmentListener {
    private TextView myText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button myButton = findViewById(R.id.button);
        myText = findViewById(R.id.textView);
        myButton.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v)
            {
```

```
                FragmentManager manager = getSupportFragmentManager();
                MyDialogFragment myDialogFragment = new
MyDialogFragment();
                myDialogFragment.show(manager, "myDialog");
            }
        });
    }
    @Override
    public void onReturnValue(String value) {
        myText.setText("Здравствуйте, "+value+"!");
    }
}
```

Задание

Разработайте приложение, в котором будут кнопка для отображения диалогового окна и текстовое поле. В диалоговом окне должно присутствовать поле для ввода текста и три кнопки:

1. Кнопка, чтобы закрыть диалоговое окно.
2. Кнопка, чтобы отобразить в активити текст, введённый в диалоговом окне.
3. Кнопка, чтобы завершить активити.

Контрольные вопросы:

1. Что такое диалог?
2. Какие классы диалогов существуют?
3. Что может содержать AlertDialog?
4. Сколько кнопок может быть в AlertDialog?
5. Каково назначение метода setCancelable?
6. Каково назначение метода setView?