**Анимация диалоговых окон**

иалоговые окна появляются на экране мгновенно, а мы попробуем сделать так, чтобы они плавно выходили с одной стороны и также плавно уходили на другую сторону.

**Системная анимация (windowAnimations)**

Создадим стиль **DialogAnimation** для будущего диалогового окна в файле **res/values/styles.xml**:

<resources>

<style name="AppTheme" parent="android:Theme.Light" />

<style name="DialogAnimation">

<item name="android:windowEnterAnimation">@android:anim/slide\_in\_left</item>

<item name="android:windowExitAnimation">@android:anim/slide\_out\_right</item>

</style>

</resources>

Мы задали два элемента для стиля с использованием готовых системных анимаций, которые отвечают за появление и исчезновения окна.

Теперь создадим простую разметку для диалога в файле **res/layout/dialog.xml**:

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/animdialog" />

<Button

android:id="@+id/dismiss"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="Отмена" />

</LinearLayout>

Осталось написать код для создания и вызова диалогового окна:

// нажатие кнопки для вызова диалога

public void onClick(View v) {

openDialog();

}

private void openDialog() {

final Dialog dialog = new Dialog(TestActivity.this);

dialog.setTitle("Пустите, пожалуйста");

dialog.setContentView(R.layout.dialog);

dialog.getWindow().getAttributes().windowAnimations = R.style.DialogAnimation;

Button btnDismiss = (Button) dialog.getWindow().findViewById(

R.id.dismiss);

btnDismiss.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

dialog.dismiss();

}

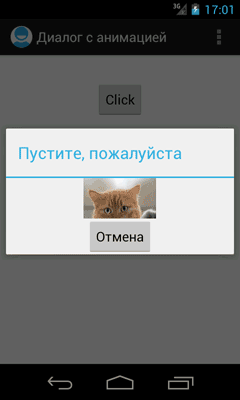
});

dialog.show();

}

Мы присваиваем анимацию свойству **windowAnimations** и теперь при нажатии на кнопку в активности диалоговое окно будет выезжать слева, а при нажатии кнопки в самом диалоговом окне, окно уедет вправо.

На картинке анимацию не увидите, поэтому придётся проверять вживую.



**Собственная анимация**

В предыдущем примере мы могли использовать только системные анимации, так как **windowAnimations** не позволяет использовать собственные. Но мы можем обойти это ограничение. Создадим свою анимацию для диалога **AlertDialog**.

Для начала создадим два файла в папке **res/anim**:

**dialog\_enter.xml**

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android"

android:interpolator="@android:anim/bounce\_interpolator" >

<scale

android:duration="700"

android:fromXScale="0.0"

android:fromYScale="0.0"

android:pivotX="50%"

android:pivotY="50%"

android:toXScale="1.0"

android:toYScale="1.0" />

</set>

**dialog\_exit.xml**

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android"

android:interpolator="@android:anim/accelerate\_interpolator" >

<scale

android:duration="500"

android:fromXScale="1.0"

android:fromYScale="1.0"

android:pivotX="50%"

android:pivotY="50%"

android:toXScale="0.0"

android:toYScale="0.0" />

<rotate

android:duration="500"

android:fromDegrees="0"

android:pivotX="50%"

android:pivotY="50%"

android:toDegrees="-720" />

</set>

Пропишем теперь стили и тему для диалогового окна:

**res/values/styles.xml**

<resources>

<style name="AppTheme" parent="android:Theme.Light" />

<style name="CustomDialogAnimationTheme" parent="@android:style/Theme.Dialog">

<item name="android:windowAnimationStyle">@style/CustomDialogAnim</item>

</style>

<style name="CustomDialogAnim" parent="@android:style/Animation.Dialog">

<item name="android:windowEnterAnimation">@anim/dialog\_enter</item>

<item name="android:windowExitAnimation">@anim/dialog\_exit</item>

</style>

</resources>

Осталось написать код для создания диалогового окна с подключением созданной темы:

// выводим на экран диалоговое окно

public void onClick(View v) {

openDialog();

}

private void openDialog() {

final CustomAlertDialog dialog = new CustomAlertDialog(TestActivity.this);

dialog.setTitle("Диалог с анимацией");

dialog.setCancelable(true);

dialog.setMessage("Суперанимация");

dialog.show();

}

public class CustomAlertDialog extends AlertDialog {

public CustomAlertDialog(Context context) {

super(context, R.style.CustomDialogAnimationTheme);

}

}

Для варианта с кнопками и использованием **Builder**:

private void openCustomDialog() {

final CustomAlertDialog builder = (CustomAlertDialog) new CustomAlertDialog(MainActivity.this)

.setTitle("Диалог с анимацией")

.setCancelable(true)

.setMessage("Кот накормлен")

.setNegativeButton("Закрыть",

new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

dialog.cancel();

}

});

builder.show();

}

public class CustomAlertDialog extends AlertDialog.Builder {

public CustomAlertDialog(Context context) {

super(context, R.style.CustomDialogAnimationTheme);

}

}