

Тема 11. «Toast. Всплывающие сообщения»¹

В телефоне, когда вы увлечены какой-то задачей, вдруг всплывает сообщение, привлекая ваше внимание. Это и есть Toast. Посмотрим, как это работает в Android.

Всплывающее уведомление (Toast Notification) является сообщением, которое появляется на поверхности окна приложения, заполняя необходимое ему количество пространства, требуемого для сообщения. При этом текущая деятельность приложения остается работоспособной для пользователя. В течение нескольких секунд сообщение плавно закрывается. Всплывающее уведомление также может быть создано службой, работающей в фоновом режиме. Как правило, всплывающее уведомление используется для показа коротких текстовых сообщений.

Для создания всплывающего уведомления необходимо инициализировать объект Toast при помощи метода `Toast.makeText()`, а затем вызвать метод `show()` для отображения сообщения на экране:

```
Toast toast = Toast.makeText(getApplicationContext(),  
"Пора покормить кота!", Toast.LENGTH_SHORT);  
toast.show();
```

У метода `makeText()` есть три параметра:

1. Контекст приложения;
2. Текстовое сообщение;
3. Продолжительность времени показа уведомления. Можно использовать только две константы.

Константы для указания продолжительности показа сообщения:

LENGTH_SHORT – (по умолчанию) показывает текстовое уведомление на короткий промежуток времени;

LENGTH_LONG – показывает текстовое уведомление в течение длительного периода времени.

Если покопаться в исходниках Android, то можно найти такие строчки:

```
private static final int LONG_DELAY = 3500; // 3.5 seconds  
private static final int SHORT_DELAY = 2000; // 2 seconds
```

Как видите, уведомления выводятся на 3 с половиной секунды или на 2 секунды. Других вариантов нет, не пытайтесь использовать другие значения - у вас ничего не получится.

По умолчанию стандартное всплывающее уведомление появляется в нижней части экрана. Изменить место появления уведомления можно с помощью метода `setGravity(int, int, int)`. Метод принимает три параметра:

- стандартная константа для размещения объекта в пределах большого контейнера (например, `GRAVITY.CENTER`, `GRAVITY.TOP` и др.);
- смещение по оси X;
- смещение по оси Y.

Например, если вы хотите, чтобы уведомление появилось в центре экрана, то используйте следующий код:

```
toast.setGravity(Gravity.CENTER, 0, 0);
```

Если нужно сместить уведомление направо, то просто увеличьте значение второго параметра. Для смещения вниз нужно увеличить значение последнего параметра. Соответственно, для смещения вверх и влево используйте отрицательные значения.

Типичная ошибка начинающих программистов - забывают добавить вызов метода `show()` для отображения сообщения на экране. К счастью, в студии, если вы пропустите метод `show()`, то строка будет подсвечена, а при подведении указателя мыши к строке увидите, рис. 1.

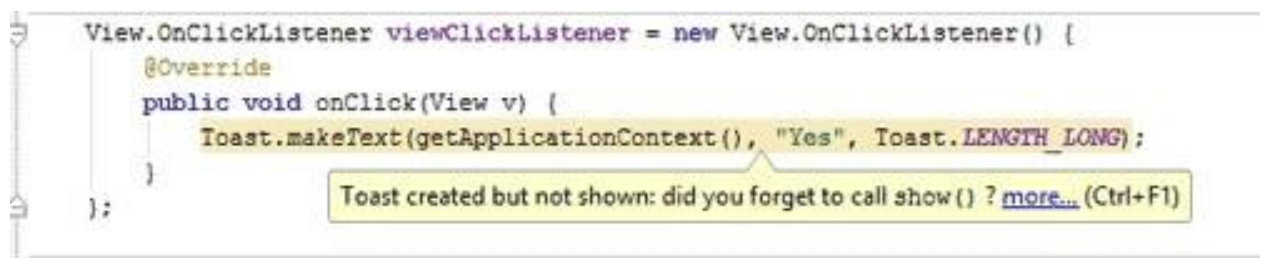


Рис. 1. Выделение строки

Теперь поэкспериментируем.

Создайте новый проект или используйте любой старый проект из предыдущих занятий. Добавьте на экран активности кнопку и присвойте ей текст Показать Toast, а также присвойте атрибуту `android:onClick` значение `showToast`. Теперь напишем код:

```
public void showToast(View view) {
    //создаем и отображаем текстовое уведомление
    Toast toast = Toast.makeText(getApplicationContext(),
    "Пора покормить кота!",
    Toast.LENGTH_SHORT);
    toast.setGravity(Gravity.CENTER, 0, 0);    toast.show(); }
```

Запустите приложение и нажмите кнопку. В центре экрана появится на короткое время текстовое сообщение, которое само исчезнет, рис. 2.

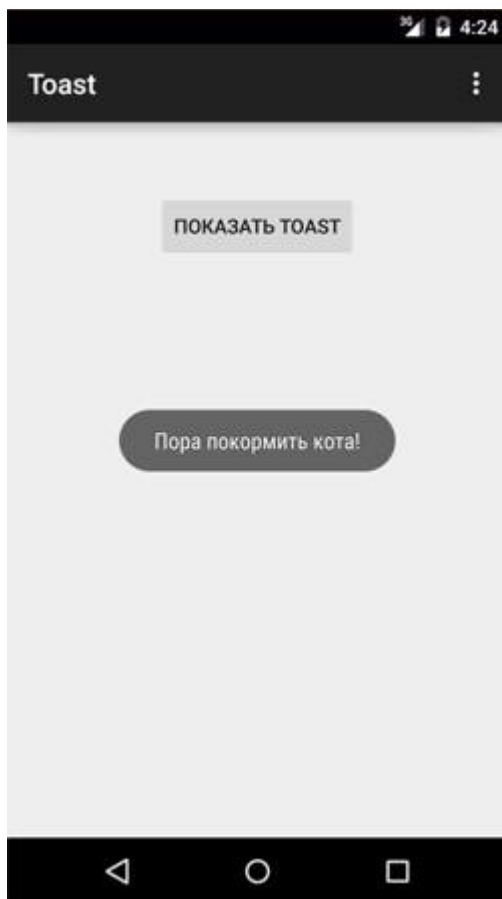


Рис. 2. Исчезающее сообщение

Для закрепления материала напишем еще один пример. Удалим предыдущий код для щелчка кнопки и напишем такой код:

```
int duration = Toast.LENGTH_LONG;  
Toast toast2 = Toast.makeText(getApplicationContext(),  
R.string.catfood,  
duration);  
toast2.setGravity(Gravity.TOP, 0, 0);  
toast2.show();
```

Отличие от предыдущего примера в том, что время показа мы увеличили, а также поместили текст сообщения в XML-ресурсы, как это рекомендуется делать. Кроме того, сообщение будет выводиться в верхней части экрана.

Как правило, для Toast используются короткие текстовые сообщения. При необходимости вы можете добавить к сообщению и картинку. Используя метод `setView()`, принадлежащий объекту `Toast`, вы можете задать любое представление (включая разметку) для отображения.

Начнем с приготовлений. Подготовьте картинку и разместите её в папке `res/drawable`. Картинка будет доступна приложению как ресурс через название файла без расширения. Например, я добавил в папку файл с изображением

кота `hungrycat.jpg` и могу получить к нему доступ через выражение `R.drawable.hungrycat`.

Чтобы изображение появилось в стандартном Toast-сообщении, нам потребуется программно создать объект класса `ImageView` и задать для него изображение из ресурсов с помощью метода `setImageResource`.

Сам по себе стандартный внешний вид Toast состоит из контейнера `LinearLayout`, в который нужно добавить созданный объект `ImageView`. Можно задать также позицию, в которую следует вывести изображение. Если указать значение 0, то изображение будет показано выше текста, рис. 3. Код для создания Toast с изображением выглядит следующим образом:

```
public void showToast(View view) {  
    Toast toast3 = Toast.makeText(getApplicationContext(),  
    R.string.catfood, Toast.LENGTH_LONG);  
    toast3.setGravity(Gravity.CENTER, 0, 0);  
    LinearLayout toastContainer = (LinearLayout) toast3.getView();  
    ImageView catImageView = new ImageView(getApplicationContext());  
    catImageView.setImageResource(R.drawable.hungrycat);  
    toastContainer.addView(catImageView, 0);  
    toast3.show();  
}
```

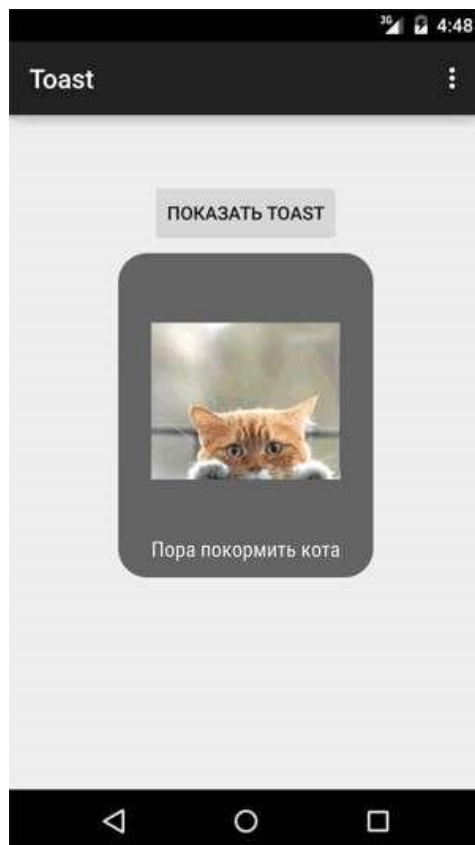


Рис. 3. Изображение в сообщении