

## Лабораторная работа № 6

### «Работа с уведомлениями в Android-приложении»

**Продолжительность выполнения** лабораторной работы: 2 ак. часа.

**Целью** лабораторной работы является знакомство с принципами создания всплывающих уведомлений и уведомлений в строке состояния.

### Практическая часть

Создайте новый проект **NotificationsApp** с пустой активити.

Откройте файл компоновки и создайте структуру с двумя кнопками для вызова уведомлений с идентификаторами button1 и button2 (рис. 1).

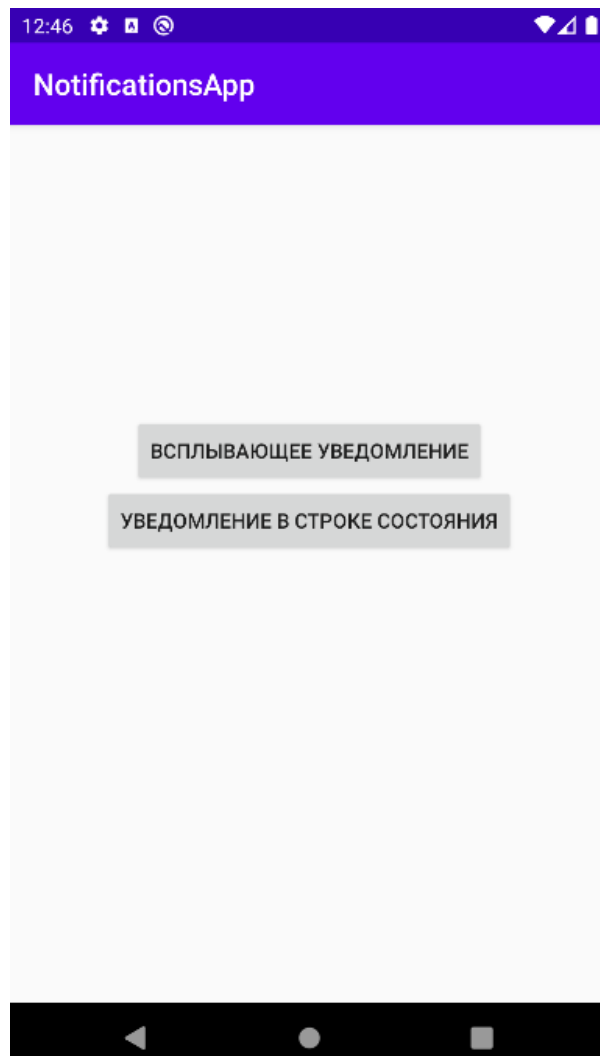


Рисунок 1. Кнопки для показа уведомлений.

Для создания новой компоновки для всплывающего уведомления щелкните правой кнопкой мыши на папке res/layout и в контекстном меню выберите **New** →

**XML → Layout XML File.** В появившемся окне введите имя нового файла компоновки — toast\_layout.

В созданном файле компоновки для уведомления корневому элементу LinearLayout присвойте идентификатор toast. Определите два дочерних элемента, ImageView и TextView. В папку drawable поместите изображение, которое будет отображаться в ImageView, а также иконку для уведомления в строке состояния.

В файле strings.xml необходимо задать строковые ресурсы, которые понадобятся для создания уведомлений (листинг 1).

#### Листинг 1. Строковые ресурсы.

```
<string name="channel_name">МойКанал</string>
<string name="channel_description">Описание моего канала</string>
<string name="notification_title">Новое уведомление</string>
<string name="notification_text">Нажмите на уведомление для перехода в
приложение</string>
<string name="toast_text">Всплывающее уведомление!</string>
```

В классе MainActivity необходимо объявить переменные notificationId и CHANNEL\_ID, который будут использоваться в качестве идентификатора уведомления в строке состояния и идентификатора канала уведомлений соответственно. Также необходимо с установить слушатель нажатия на кнопки.

#### Листинг 2. Фрагмент класса MainActivity.

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
    int notificationId = 1;
    String CHANNEL_ID = "my_channel_01";
    Button button1;
    Button button2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button1 = findViewById(R.id.button1);
        button2 = findViewById(R.id.button2);
        button1.setOnClickListener(this);
        button2.setOnClickListener(this);
    }
}
```

Далее необходимо реализовать метод создания канала уведомлений для доставки уведомлений на Android 8.0 и выше (листинг 3).

### Листинг 3. Создание канала уведомлений.

```
private void createNotificationChannel() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = getString(R.string.channel_name);
        String description = getString(R.string.channel_description);
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
        NotificationChannel channel = new
NotificationChannel(CHANNEL_ID, name, importance);
        channel.setDescription(description);
        NotificationManager notificationManager =
getSystemService(NotificationManager.class);
        notificationManager.createNotificationChannel(channel);
    }
}
```

Наконец, следует написать обработчик события onClick и, в зависимости от того, на какую кнопку было произведено нажатие, либо вызывать всплывающее уведомление, либо показывать уведомление в строке состояния (листинг 4).

### Листинг 4. Метод onClick.

```
@Override
public void onClick(View v) {
    if(v==button1) {
        LayoutInflater inflater = getLayoutInflater();
        View layout = inflater.inflate(R.layout.toast_layout,null);
        TextView text = layout.findViewById(R.id.text_view);
        text.setText(R.string.toast_text);
        Toast toast = new Toast(getApplicationContext());
        toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
        toast.setDuration(Toast.LENGTH_SHORT);
        toast.setView(layout);
        toast.show();
    }
    else if(v==button2) {
        createNotificationChannel();
        Intent intent = new Intent(this, MainActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
        PendingIntent pendingIntent = PendingIntent.getActivity(this,
0, intent, 0);
        final NotificationCompat.Builder builder = new
NotificationCompat.Builder(this, CHANNEL_ID)
.setSmallIcon(R.drawable.status_bar_icon)
```

```

        .setContentTitle(getString(R.string.notification_title))
        .setContentText(getString(R.string.notification_text))
        .setPriority(NotificationCompat.PRIORITY_DEFAULT)
        .setContentIntent(pendingIntent)
        .setAutoCancel(true);
        NotificationManagerCompat notificationManager =
        NotificationManagerCompat.from(getApplicationContext());
        notificationManager.notify(notificationId, builder.build());
    }
}

```

### Задание

Разработайте приложение, в котором будут кнопки отправки всплывающего уведомления с индивидуальным дизайном и уведомления в строке состояния. Самостоятельно дополните приложение, разработанное в практической части, добавьте поле для ввода текста на компоновку активности и изменяйте сообщение во всплывающем уведомлении, выводя текст, введённый пользователем в активности.

### Контрольные вопросы:

1. Какие типы уведомлений существуют?
2. Что такое Toast Notification?
3. Что такое уведомление в строке состояния?
4. Каково назначение метода `getApplicationContext`?
5. Каково назначение метода `Toast.makeText`?
6. С помощью каких констант задаётся продолжительность отображения Toast уведомления?
7. Каково назначение метода `setGravity`?
8. Каково назначение метода `getLayoutInflater`?
9. Каково назначение метода `setDuration`?
10. Из каких шести частей состоит шаблон уведомления на панели уведомлений?
11. Что такое канал уведомлений?
12. Для чего нужен класс `NotificationCompat`?
13. Какие уровни важности и уровни приоритета уведомлений существуют?