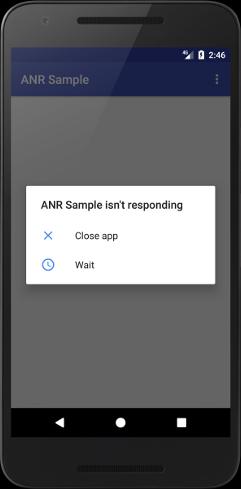
Лабораторная работа № 6

Потоки в Android.

Общие сведения.

При запуске приложения Android главным потоком исполнения команд является поток пользовательского интерфейса **UI thread**.

Этот поток в процессе работы приложения нельзя блокировать операциями, которые требуют большого или заранее не прогнозируемого времени исполнения, т.к. если основной поток приложения будет недоступен пользователю в течение 5 сек., то ОС Android выбросит сообщение ANR (Application Not Responding).



Для избежания блокировки UI thread такие операции следует выполнять в отдельном потоке.

Чтобы создать поток, необходимо создать экземпляр классаJava  **Thread**, внутри которого переопределить метод *run()* интерфейса **Runnable.**

В методе *run()* помещается код, который должен выполняться в потоке.

Запуск на исполнение потока осуществляется методом *start()* экземпляра класса **Thread**. Для примера приведем код запуска потока по нажатию кнопки:

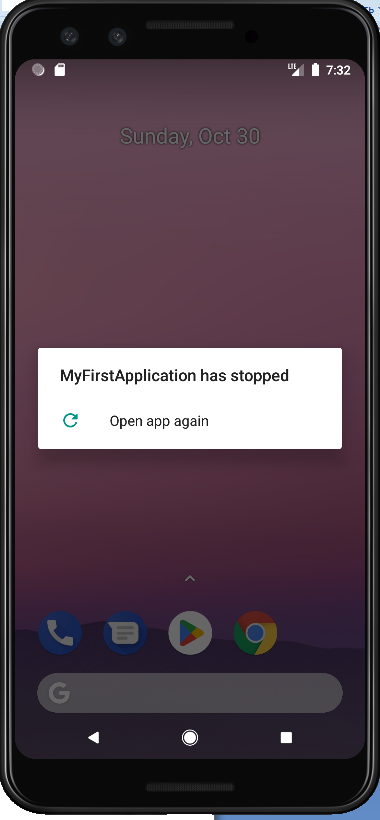
button1.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 *// do something* }  
 }).start();  
 }  
});

Часто необходимо результат выполнения кода в потоке отобразить в пользовательском интерфейсе, например, результат запроса к БД или загрузки контента по сети.

Так как из потоков нельзя напрямую взаимодействовать с компонентами View, элементы пользовательского интерфейса имеют специальный метод  **post()**.

Этот метод вызывается внутри потока у элемента пользовательского интерфейса к которому необходимо обратиться. При этом в метод  **post()** передается реализация интерфейса **Runnable** с кодом, взаимодействующим с элементом View.

Например, для того, чтобы результат выполнения кода в потоке отобразить в **TextView** необходимо выполнить следующий код:

**new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 *// do something useful;* String str = **"result of thread code"**;  
  
 textView1.post(**new** Runnable() {  
 @Override  
 **public void** run() {  
 textView1.setText(str);  
 }  
 });  
 }  
}).start();

Если же попытаться обратиться к **TextView** напрямую из кода потока,

**new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 *// do something useful;* String str = **"result of thread code"**;

textView1.setText(str);  
 }  
}).start();

то возникнет ошибка приложения, которая приведет к его завершению.

Ход выполнения проекта.

1 Откройте проект, созданный в лабораторной работе № 5.

2 Измените способ работы с БД таким образом, чтобы запросы из пользовательского интерфейса к БД выполнялись в отдельных потоках.