Hilos en Android

Thread, Runnable

android.os.Handler

android.os.AsyncTask<Params, Progress, Result>

Hilos en Android

- Usamos las hebras o los hilos de ejecución en aquellas situaciones en las que el tiempo de ejecución de una tarea compromete el funcionamiento de la aplicación completa.
- Cuando una aplicación tarda demasiado tiempo en ejecutar alguna tarea, se muestra un mensaje de error ANR, *Applicaton Not Responding*. Este tipo de situaciones se tienen que evitar.

UI Thread

- Cada aplicación que se ejecuta en un dispositivo Android se ejecuta en un hilo diferente, el hilo UI de la interfaz de usuario.
- Todos los componentes de una misma aplicación se ejecutarán en ese mismo hilo.
- Hay que tener en cuenta que los hilos no pueden acceder a los elementos de la interfaz gráfica.

Acceso a la UI Thread

- Algunos métodos ofrecen la posibilidad de acceder a la interfaz de usuario desde el hilo de ejecución:

- Activity.runOnUiThread(Runnable)
- View.post(Runnable)
- View.postDelayed(Runnable, long)

android.os.Handler

- Un objeto de la clase Handler es un hilo que se ejecuta en el hilo principal, el UI Thread.
- Si creamos un hilo nuevo, podremos enviar mensajes desde el hilo al objeto de la clase Handler.
- Como el objeto Handler está en el hilo UI Thread puede acceder a la interfaz.

CalledFromWrongThreadException

- Al acceder desde cualquier hilo diferente del UI Thread a la interfaz de usuario, se obtiene esta excepción.

```
                                    00)
com.ejemplo.age...  AndroidRuntime  FATAL EXCEPTION: Thread-99
com.ejemplo.age...  AndroidRuntime  android.view.ViewRootImpl$CalledFromWrongThreadException: Only the  [
                                    original thread that created a view hierarchy can touch its views.
com.ejemplo.age...  AndroidRuntime  at android.view.ViewRootImpl.checkThread(ViewRootImpl.java:4609)
```

Implementar el hilo

```
public class Hilo implements Runnable{
   @Override
   public void run() {
        …
        Message msg = new Message();
        msg.obj = "Resultado: " + resultado;
        objetoHandler.sendMessage(msg);
   }
}
```

Implementar el Handler

```
public class Manejador extends
                                        Handler{
    @Override
    public void handleMessage(Message
                                            msg) {
        tv.setText((String)msg.obj);
    }
}
```

Consideraciones adicionales

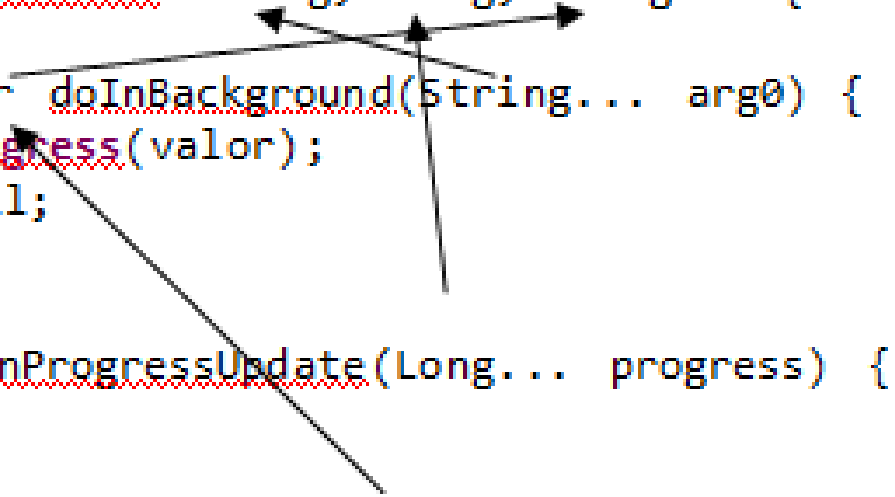
- La clase del hilo tiene que tener acceso al objeto Handler.
- La clase del Handler tiene que tener acceso a los objetos de la Activity.
- Lo más indicado será crear ambas clases como clases internas de la actividad.

AsyncTask

- Simplifica la programación de hilos.
- Facilita el acceso a los controles de la interfaz de usuario.
- Previene la sobrecarga de hilos.
- El método **doInBackground**() ejecuta las tareas en segundo plano.
- El método **onPostExecute**() implementa el acceso a los controles de la interfaz de usuario una vez que ha finalizado el hilo.

AsyncTask

```
class Tarea extends AsyncTask<String, Long, Integer> {
        @Override
        protected Integer doInBackground(String... arg0) {
                publishProgress(valor);
                return null;
        }
        @Override
        protected void onProgressUpdate(Long... progress) {
        }
        @Override
        protected void onPostExecute(Integer time) {
        }
}
```



AsyncTask

```
public class Tarea extends AsyncTask<String, Integer, Boolean> {
    @Override
    protected Boolean doInBackground(String... params) {
        publishProgress(valor);
        if(isCancelled())
            break;
        return true;
    }
    @Override
    protected void onProgressUpdate(Integer... values) {
    }
    @Override
    protected void onPreExecute() {
    }
    @Override
    protected void onPostExecute(Boolean result) {
    }

    @Override

    protected void onCancelled() {
    }
}
```

Utilización

- Para mostrar el progreso se utiliza el método publishProgress().
- Para lanzar el hilo, se crea una instancia de la clase y se ejecuta con execute().

```
Tarea t = new Tarea();
t.execute(new String[] {"…", "…"});
```