Desenvolvimento de Software para Dispositivos Móveis

Aula 9 - Thread - Handler - AsyncTask

Professor: Anderson Almada

Thread

- Quando um aplicativo é executado, o sistema cria um thread de execução para ele, que é chamado de “principal”.

- Esse thread é muito importante porque ele é encarregado de despachar eventos para os widgets adequados da interface do usuário, incluindo eventos de desenho.

- Quase sempre, é também o thread em que o aplicativo interage com componentes do kit de ferramentas da IU do Android (thread de IU)

Thread

```
new Thread(new Runnable() {
    public void run(){

    }
}).start();
```

Thread

```
new Thread(new Runnable() {
    public void run(){
      txt1.setText("" + i);
    }
}).start();
```

Thread

```
new Thread(new Runnable() {
    public void run(){
        while(true) {
          txt1.setText("" + i); i++;
          try {
            Thread.sleep(1000);
          } catch (InterruptedException e) {
             e.printStackTrace();
          }
      }
    }
}).start();
```

Thread

```
new Thread(new Runnable() {
    public void run(){
        while(true) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    txt1.setText("" + i); i++;
                }
            });
...
```

Handle

```
Handler handler = new Handler();
...
handler.postDelayed(new Runnable() {
    @Override
    public void run() {
                                Toast.makeText(getApplicationContext(),       "check",
Toast.LENGTH_SHORT).show();
      handler.postDelayed(this, 5000);
    }
}, 1500);
```

AsyncTask

- Para fornecer uma boa experiência do usuário em nosso aplicativo, precisamos usar a classe AsyncTask que é executada em um thread separado.

- Essa classe executará tudo no método **doInBackground** () dentro de outro thread que não tenha acesso à GUI onde todas as visualizações estão presentes.

AsyncTask

- O método onPostExecute () dessa classe se sincroniza novamente com o thread principal da interface do usuário e permite fazer algumas atualizações.

- Este método é chamado automaticamente após o método doInBackground terminar seu trabalho.

AsyncTask

- Para usar o AsyncTask, você deve subclassificá-lo. Os parâmetros são o seguinte AsyncTask <TypeOfVarArgParams, ProgressValue, ResultValue>

```
private class DownloadFilesTask extends AsyncTask<URL, Integer, Long> {
…
}
-----------------------------------------
new DownloadFilesTask().execute(url1, url2, url3);
```

AsyncTask

```
private class DownloadFilesTask extends AsyncTask<URL, Integer, Long> {
    protected Long doInBackground(URL... urls) {
        return;
    }

    protected void onProgressUpdate(Integer... progress) {
    }

    protected void onPostExecute(Long result) {
    }
}
```

AsyncTask

- **onPreExecute**() - chamado no thread principal da interface do usuário antes da tarefa ser executada

- **doInBackground(Params)** - Este método é chamado no thread em segundo plano imediatamente após onPreExecute () concluir sua execução. O principal objetivo desse método é executar as operações em segundo plano que podem levar muito tempo.

AsyncTask

- **onProgressUpdate (Progress...)** - Este método é chamado no thread principal da interface do usuário após uma chamada para **publishProgress** (Progress...).

- **onPostExecute (Result)** - Esse método é chamado no thread principal da interface do usuário após a operação em segundo plano terminar no método doInBackground

AsyncTask

- **Definição da classe**

```
private class AsyncTaskExample extends AsyncTask<String, Void, Bitmap> {
...
}
```

- **Executa a tarefa**

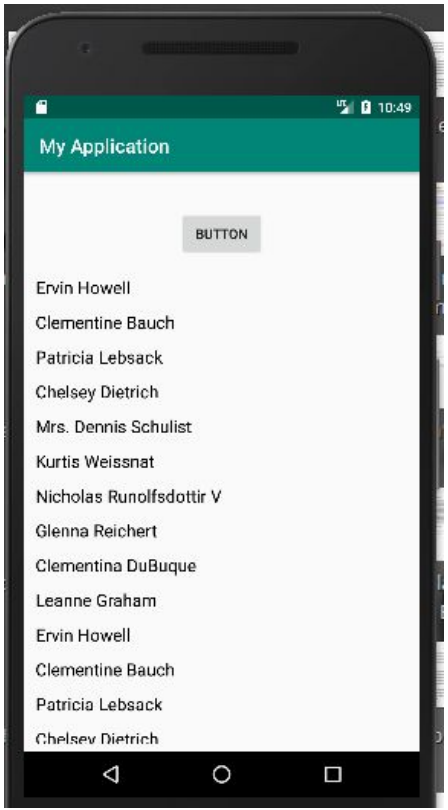
```
new AsyncTaskExample().execute("url");
```

- https://pastebin.com/rRWteAHw

AsyncTask

- Não esquecer de colocar a permissão de acesso a internet, se precisar.

Exercício


10:49
My Application
BUTTON
Ervin Howell
Clementine Bauch
Patricia Lebsack
Chelsey Dietrich
Mrs. Dennis Schulist
Kurtis Weissnat
Nicholas Runolfsdottir V
Glenna Reichert
Clementina DuBuque
Leanne Graham
Ervin Howell
Clementine Bauch
Patricia Lebsack
Chelsey Dietrich

Exercício

- Liste o nome dos usuários que estão no seguinte link
  - **https://jsonplaceholder.typicode.com/users**

- O seguinte método convert Inputstream em String
  - https://pastebin.com/1fBigFxp

Exercício

- Você pode converter uma string em JSONObject ou JSONArray (depende do que de fato ele é)
  - JSONArray jsonArray = **new** JSONArray(String) ou
  - JSONObject json = **new** JSONObject(String)

Exercício

- Com o jsonArray é possível retornar um jsonObject a partir da posição
  - JSONObject json = jsonArray.getJSONObject(i);

- Com o objeto jsonObject pode-se utilizar o get para retornar o objeto que quiser
  - json.get("**name**");

Link importante

- https://developer.android.com/guide/components/processes-and-threads?hl=pt-br

- https://developer.android.com/reference/android/os/AsyncTask

Dúvidas??

E-mail: almada@crateus.ufc.br