Class Hierarchy Notes

An activity: A core building block of Android apps. It is similar to a template or a webpage, can be started, paused, stopped, and resumed.

Model View Controller is a valid way of viewing this.

PodcastActivity.java **Central/Home page** associated with activity_podcast.xml

- onCreate(Bundle savedInstanceState)
    - Activated when the activity is started.
- Holds static variables
    - mlist is a ListView item.
        - It holds an adapter, which you can put strings into.
- Intent- When you want to start a new activity, you create an intent which holds an activity (the activity is a parameter when you create an intent).
    - Intent intent =new Intenet(getAppContext(), AddActivity.class). Use an activity as a param
    - StartActivity(intent) start the activity that is contained in the intent.
- onClick(View v) of our add button
    - Sets off the intent to start the addActivity.
    - Move over to addActivity.java and activity_add.xml

- onClick(View v) of our add button. The override sections makes it so that our button posts a toast as well as downloading.
- performDownloads()//does what it says.
    - Goes through all the most recent episodes for a given URL, up to five, and performs downloads with them.

Handy things:

1. SetContentView(<xmlFile>) posts/pastes the xml graphical layout to the screen when this activity is the active one.

AddActivity.java associated with activity_add.xml

The add-podcast button gets a listener (basically every button gets a listener).
When pushed, calls createPodcast.
createPodcast takes the URL and then calls the new GetRssFeed().execute().
Executes URL

GetRssFeed is a class that extends AsyncTask. AsyncTask performs background tasks that are not on the UI thread.

GetRssFeed.execute() results in performing doInBackground, which has been overrided. It creates a new podcast to put into the list of them. It populates the different fields of the podcast using the RssReader.java

Podcast.java class holds a handful of variables. RssItems are episodes.

Then we add the podcast title to the adapter that will go into our listView and we add the podcast itself to our list of podcasts.

GetRssFeed.onPostExecute()-----After finishing the doInBackground, set onClick listeners to the new items in mlist (each of those items is a podcast title and will be clickable)

Back to PodcastActivity.java
onItemClick()--If the episodes are being displayed and you just clicked on one of them!:

1. Put 2 key/value into a hash connected to our audio player intent.The values will be title and description.
2. Start audio player, now complete with title and description in the intent and therefore available in the activity. They will be displayed.

AudioPlayerActivity.java

Sets a large number of variables. Puts together the setup for the xml and audioplayer as a whole. Plays podcast.

The RssReader and the RssHandler work together to initialize the Podcast and RssItem objects, which are then stored in memory. RssItems are stored in the Podcast class using an ArrayList, while the Podcasts are stored in PodcastActivity.podcastList

PodcastActivity.java
-Our main class
-Home page
-Associated with activity_podcast.xml

-Allows us to go to AddActivity.java to add podcast feeds
-Allows us to download episodes
-Displays podcasts and episodes

RssHandler.java
RssReader.java

These two work together to parse the xml file.
They grab relevant data from the file to use in initializing the Podcast and its constituent RssItems.

AddActivity.java
-Associated with activity_add.xml
-Started when the user clicks the "add" button on activity_podcast.xml

-Allows us to take a string as URL.
-Uses that string to grab the xml podcast file.

-Has a private class called GetRssFeed. GetRssFeed, when executed, runs processes in the background.
-Specifically, GetRssFeed uses RssReader.java and RssHandler.java to parse it and initialize a Podcast object and several RssItem objects. It then stores them in appropriate areas and sets onClick listeners for the episodes that are soon to be visible in PodcastActivity.java.

Podcast.java
RssItem.java

Podcast.java holds variables relavent to each podcast, such as title, description, and an ArrayList of RssItem objects. Each RssItem corresponds to and contains data for an episode of the Podcast.

AudioPlayerActivity.java
-Associated with activity_audio_player.xml
-Plays the podcast
-Displays title,etc.