Requirements list for SetupPollTrackerController.java

1. Three integer private local variables to store the values of numberOfPolls, numberOfSeats, and numberOfParties. When you get the values of these text fields from the user, they will have to be converted from TextField objects into integers.

private int localNumberOfPolls = 0;

private int localNumberOfSeats = 0;

private int localNumberOfParties = 0;

1. A local version of the main PollList which is called polls. The idea here is that you will fill and/or make changes to the local version of polls, and then copy it back over to the main version of polls.

private PollList localPolls;

1. A local version of the main Factory which is called factory. Again, the idea here is that you will create a local factory, and then copy it back over to the main factory.

private Factory localFactory;

1. In order for your code to know what the types PollList and Factory mean, you will have to import those classes from the model package.
2. “Constructor” method: setupController
   1. Make a link between the local version of the PollTrackerApp (this.app) and the main version of the PollTrackerApp which is passed to this method.
   2. Create a new set of polls using some default values. Since you will obviously be creating new sets of polls whenever someone puts in new values, it makes sense to put the process of creating new polls into a local private helper method (I called mine createNewPolls()
3. refresh() method:

The only thing that needs to be done is this methos is to set the TextField values to the local integer values for the three appropriate integer variables defined in step 1 above. You might not think that you even need to do this, but depending on how other people write their portions of the code, it may be that these values get changed in the main app from another tab, and so we should always update them in the GUI each time we either come back to this tab, or do anything really.

1. handleClear() method:
   1. Set the values of the three local integer variables back to zero.
   2. Call the refresh() method, which will update things on the GUI.
2. handleSubmit() method:
   1. Set the values of the three local integer variables to whatever the user put into the TextField boxes.
   2. Call the createNewPolls() method.
   3. Call the refresh() method.
3. createNewPolls() method:
   1. The assumption here is that this tab is going to be clicked on FIRST when someone starts up the app, and then they will create a set of polls that has some default set of names for each party. In the requirements document, she has chosen to just call the parties 1,2,3, etc., which seems like a reasonable choice to me. So, the first step then is to create a local String[] array that contains the names of the parties:

String[] nameList = new String[localNumberOfParties];

Then, you will need to write a little for loop to set the various elements of nameList to be “1”, “2”, “3”, etc.

* 1. Create the local factory object and initialize it with nameList that you just created by calling the setPartyNames() method of the Factory class.
  2. Create the local polls object by calling the createRandomPollList() method of the PollList class.
  3. You now have created your local versions of the polls and factory, and now all you have to do is copy them into the main app by calling the setPolls() and setFactory() methods on the PollTrackerApp object which is called app.