The name of the project is Stupid Views :D Why stupid views? Because they are just simple and primitive imitations of real views that we will have in Android. So everywhere you will see word “View” in this doc, think about it as an imitation of view

Goal: Practice creating and using interfaces and inheritance. Practice creating anonymous inner classes. Also a little bit introduction to some view class name that we will meet in Android

We should be able to create some “Views” and add them to a “ViewGroup” (sort of container for views). We should be able to add click listeners to every view and imitate the click on the view in terminal.

We will have a class called **View** which will be the parent of all Views. We will have the following types of Views: **Button**, **Checkbox** and **Spinner** (drop down menu)

We will have a class called **ViewGroup** which will be a sort of panel to hold different views and will have a method called **draw()** which will call **toString()** on each view (Views must override the toString() method to show their state, for example Button will print it’s title, Checkbox will print if it is checked or not, Spinner will print if it is opened or closed and if it is opened it will print the String items in it)

The parent View class will have a method **performClick()** and a nested interface called **OnClickListener** which will have one method - **onClick(View v).** In our code we should set different OnClickListeners to different views.

Checkbox will also have another nested interface called OnCheckChangedListener which will have 1 method **onCheckChanged(View v, boolean isChecked).** If this listener is set to checkbox any time the check state is changed this *event will be triggered* and will be passed the view and checked state of the checkbox.

We will have a class called **TestStupidViews** in which we will create 2 Buttons, 2 Checkboxes and 1 spinner in its init() method.
Click actions:

- Button 1 will only print it’s name - Button Clicked: Button name
- Button 2 will print its name like Button 1 and also increment a **counter** instance variable of class **TestStupidViews** and print the counter number.
- CheckBox 1 will change its checked state and print it like (Checkbox clicked: isChecked = true)
- Checkbox 2 will do the same as checkbox 1, but it will have a OnCheckChangedListener attached to it, and from onCheckChanged(View v, boolean isChecked) *callback* we will change the boolean instance variable of **TestStupidViews** class called canExitProgram

- Spinner will have a state - opened or closed. It will hold some strings in it, and when it is clicked it will change it’s state and print either “Spinner is opened” or “Spinner is closed”. And when it is opened it will also print the items in it.

The imitation of the clicks will happen in the following way: When the program starts it will ask to type an index of a view in the ViewGroup to click on it or 0 to draw the ViewGroup.
It will ask this again and again in a loop until:

- The counter is >= 10
- The canExitProgram = true

When the above two conditions are true then the program will exit.

New terminology used here:

- Event will be triggered
- Callback