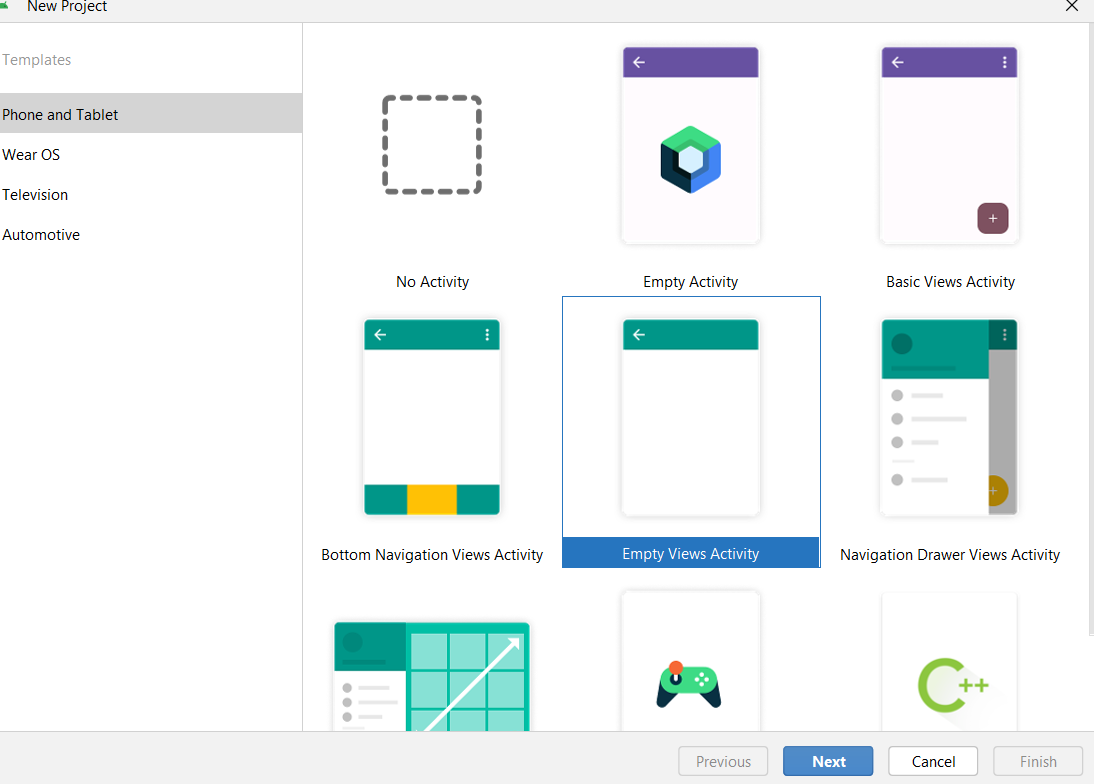
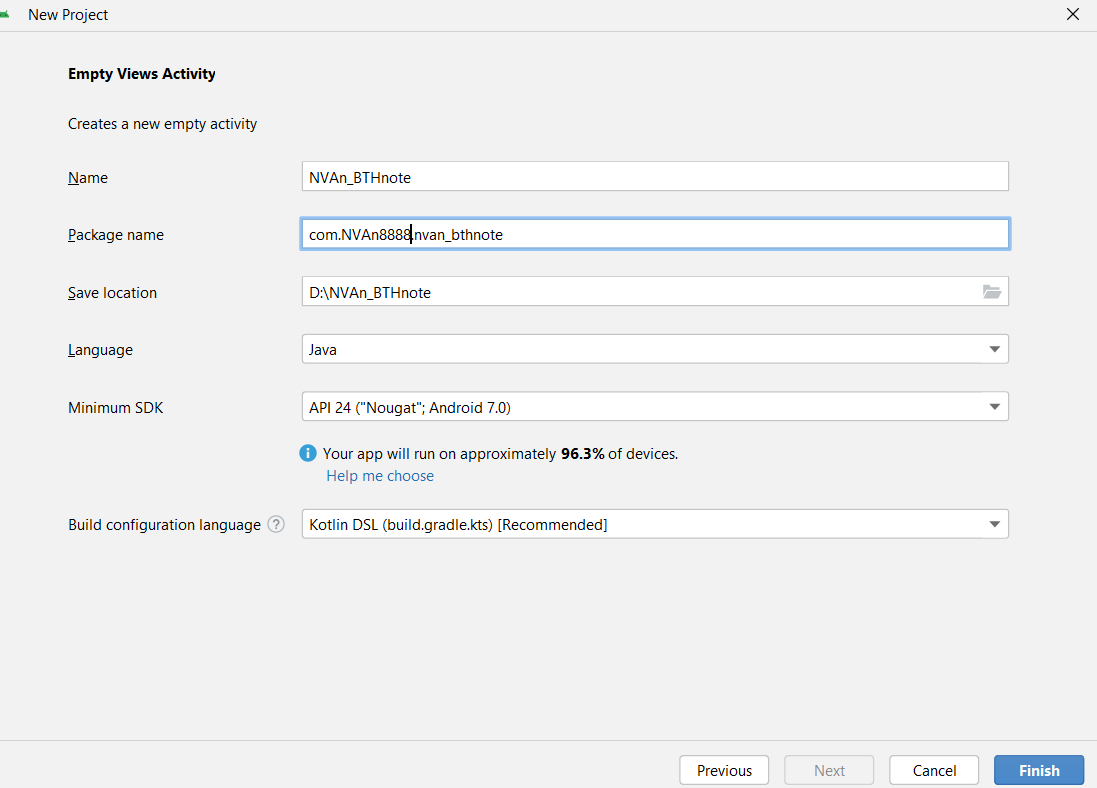
- Create a new project

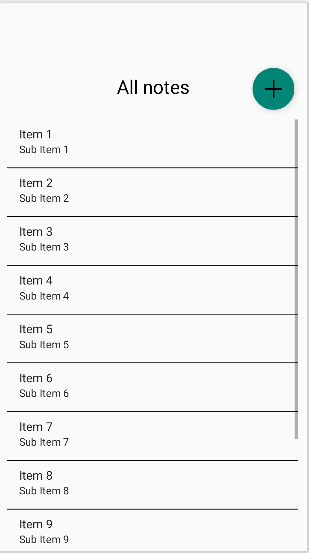


* Create project name and package name according to regulations:

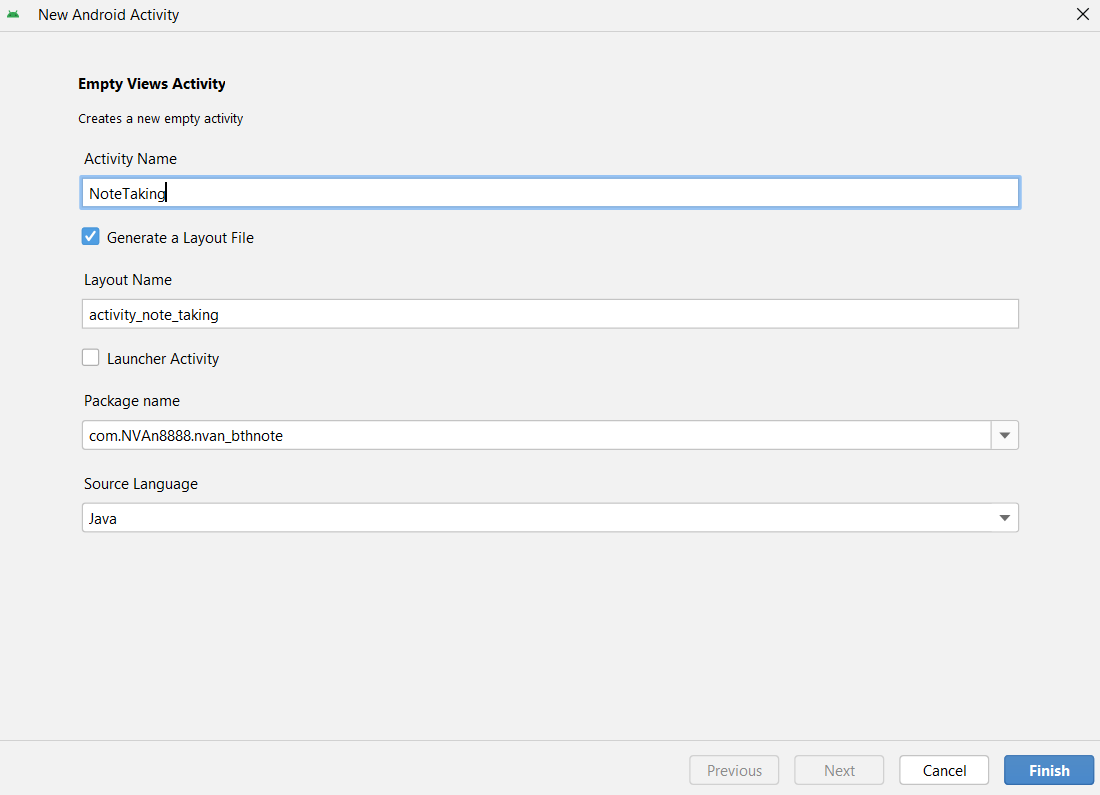


* Open activity\_layout.xml and design the interface including 1 listview, 1 *FloatingActionButton*

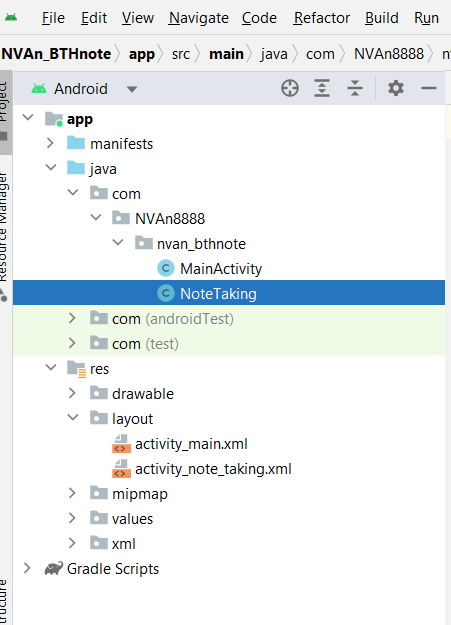
For example, listview has id **lvNote**, FloatingActionButton has id **btNote**



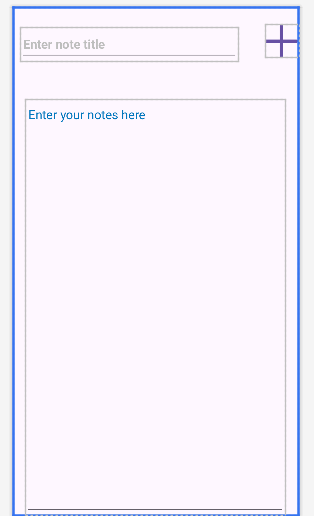
- Create a new activity named **NoteTaking.java** and a layout named **activity\_note\_taking.xml**



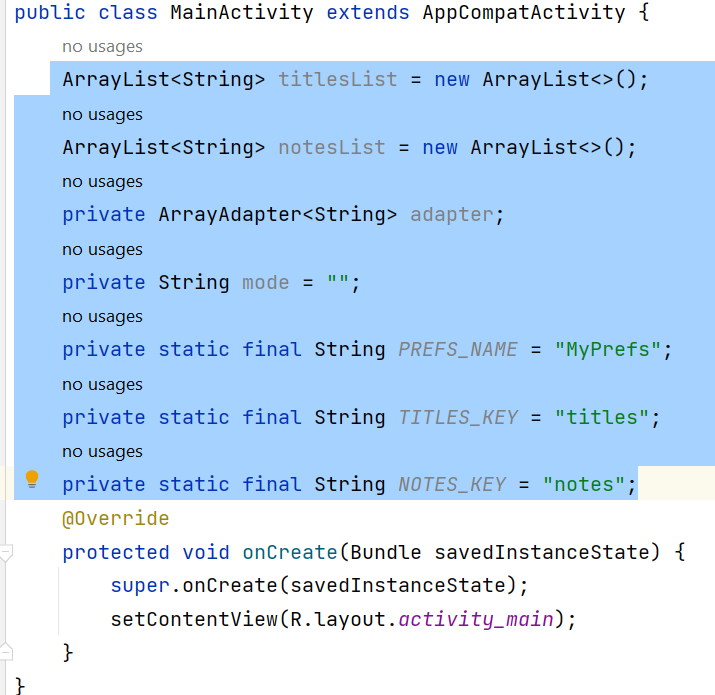
The results are as shown:



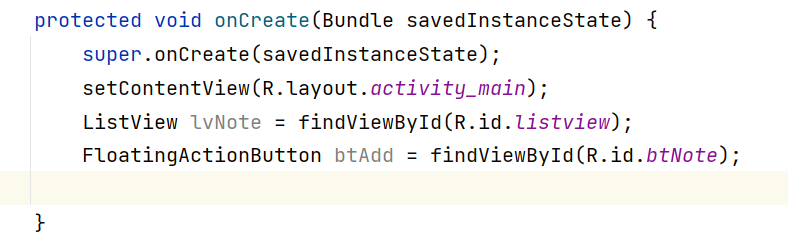
- Design layout **activity\_note\_taking.xml** with 2 edittexts with ids **edNContents** and **edNTilte**, 1 button with id **btAddNote**



- Open MainActivity.java, programming: Declare list to store title and notelist, declare string containing information to store notes in phone memory



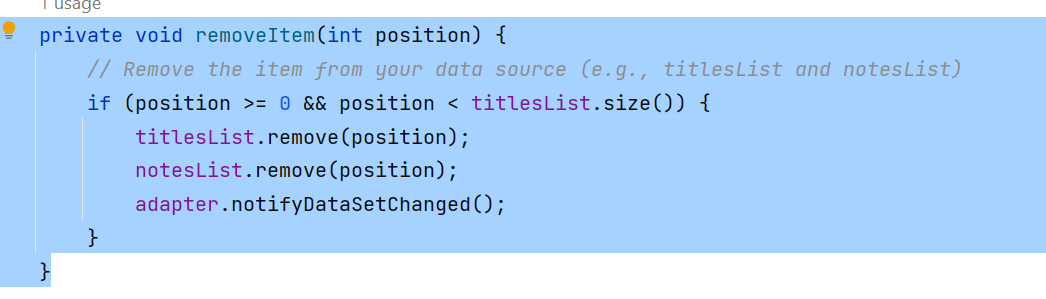
* Continue in the main **OnCreate** flow declaring and assigning id:



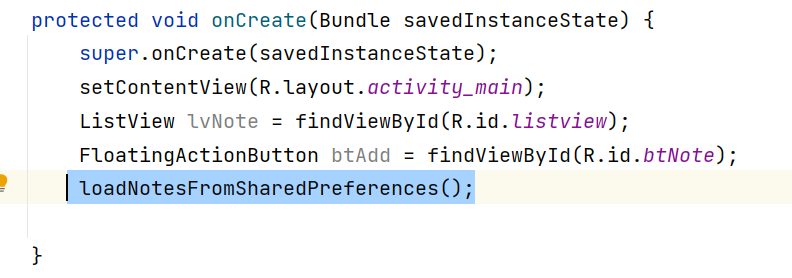
* Outside the main thread, Program a subfunction to load data (title and content) stored notes from memory to store in a list.



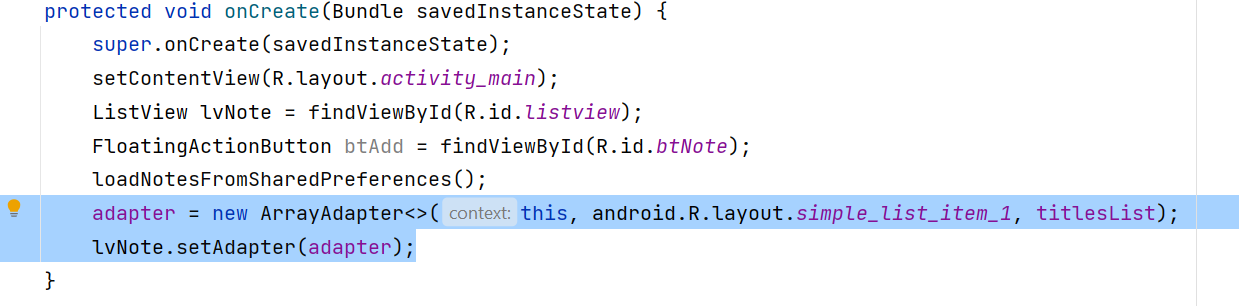
- Still outside the main thread, program a subfunction to delete the note saved in memory according to the selected location:



- Call the function to load note data in the main thread:



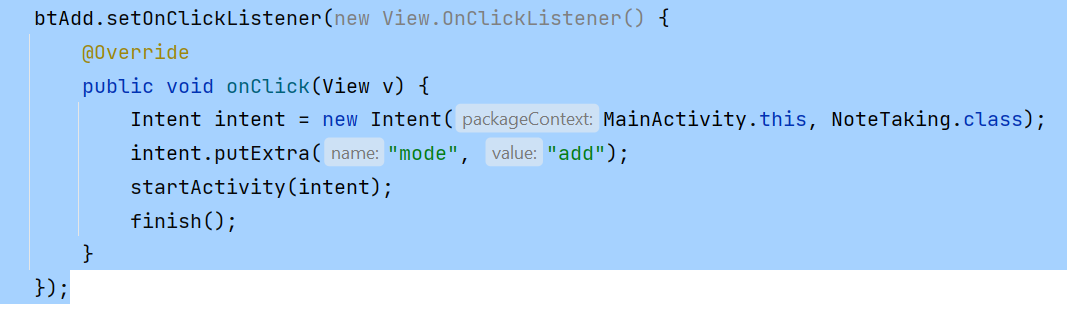
- Still in the main thread, continue pouring data to load the listview (only load the title) and display the saved notes on the listview.



- Still in the main flow, the Functional Programmer clicks on the listview to edit the saved note. Here, the information clicked on the notes including title, content, location is taken from the click location and the mode here will transmit the string "edit" to notify the target activity of the content that needs to be edited:



- Still in the main flow, continue programming events for FloatingActionButton **btAdd** to create a click event that requires adding a note. At this time, the sending mode has the content "add"

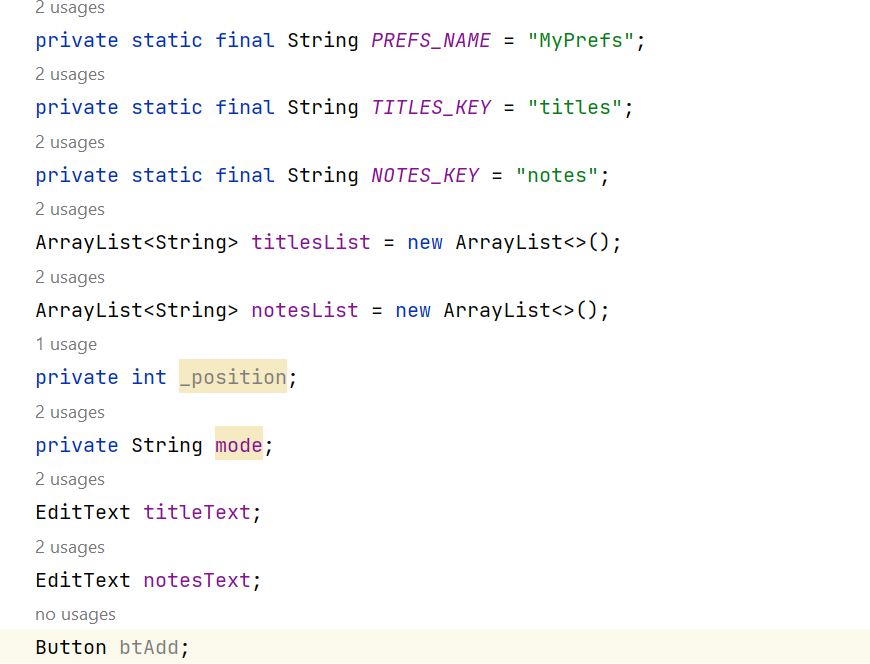


- Still in the main flow, Create a listview hold click event to delete the selected note:

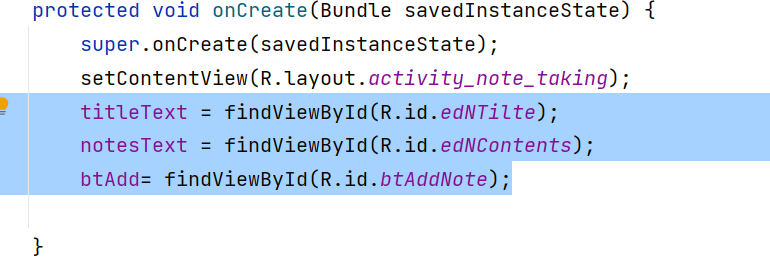


- Compile and test the functions. If there are no errors, you can go to the next step

- Open **NoteTaking.java** and programmatically declare global objects before the main thread:



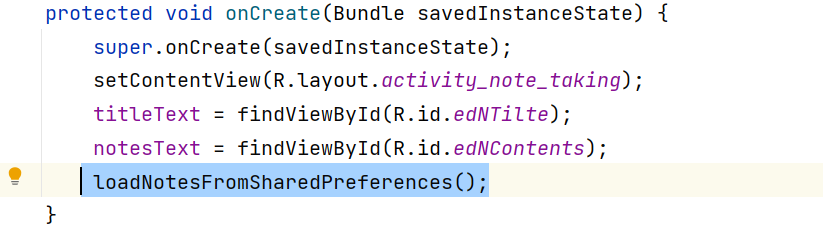
In the main thread, assign ids to objects:



- Outside the main thread, Program a subfunction to load data (title and content) stored notes from memory to store in a list.



- Call the function to load note data in the main thread:



* Still in the main thread, program the intent to receive data received from the main activity to determine what the mode is? If the mode is edit, receive the information that needs to be edited, including the title, content and location of the note that needs to be edited. At the same time, load the content to be edited into 2 edittext



- Still in the main thread, program the **btAdd** button event and get the input from the edittext



- Still in the main flow and programming events for the **btAdd** button, programming to check and handle the add or edit condition of notes. In case the **mode** is “add”:



- Stay in the main thread and program the event for the **btAdd** button, in case the **mode** is “edit”:



- Operate and test functions: add, edit, delete notes