Google Maps and Advanced Widgets

Google Maps

- After completing the Family Map Login assignment, the next step on the project could be to implement your MapFragment class (which can then be embedded in your MainActivity and Event Activity)

Google Maps

- I'm not a fan of the Google Maps chapter in the textbook. You might choose to ignore it, and instead use the Google tutorial and the code examples on the CS 240 web site
- Google tutorial on Google Maps
  - https://developers.google.com/maps/documentation/android-sdk/start
  - Shows how to embed a Google Map fragment in an Activity (i.e., fragment inside an activity)
- MapFragmentExample project in the lecture notes
  - Shows how to embed a Google Map fragment in a Fragment (i.e., a fragment inside another fragment)

Useful Map Related Methods

- GoogleMap.clear()
- GoogleMap.addMarker(…)
- GoogleMap.addPolyLine(…)
- GoogleMap.moveCamera(…)
- GoogleMap.animateCamera(…)
- GoogleMap.setMapType(…)
- GoogleMap.setOnMarkerClickListener(Marker)
- Marker.setTag(…)
- Marker.getTag(…)

Advanced Widgets

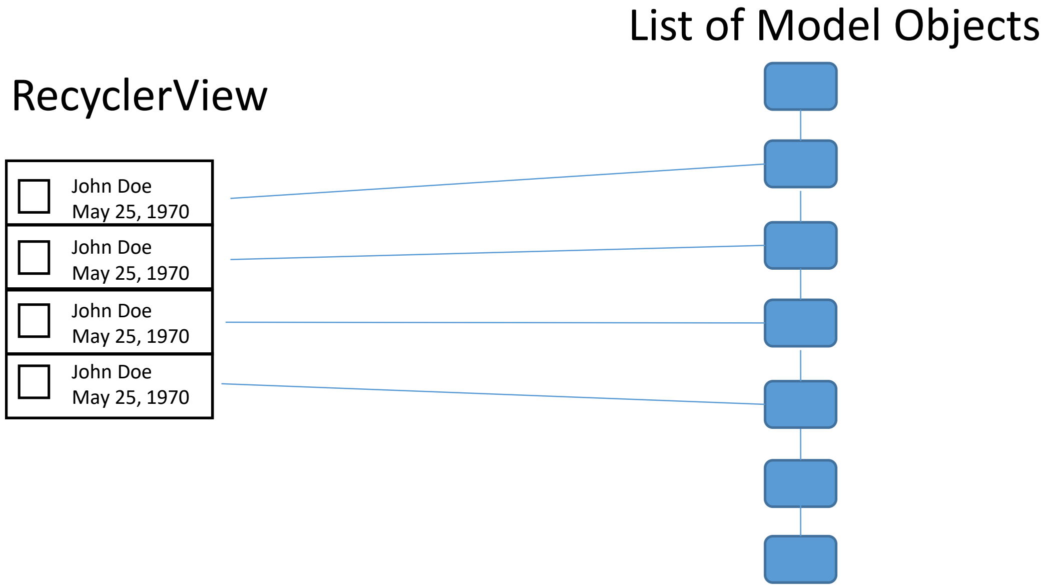
- Simple
  - Button, TextView, EditText, Switch, RadioButton, etc.
  - Display simple values
- Complex
  - Display complex data or functionality
  - Lists, Tables, Trees
  - How do you feed your app’s data into a complex widget when the widget doesn’t know anything about your data?
  - How can a widget display your data and let the user edit your data when it doesn’t understand your data?

Advanced Widgets

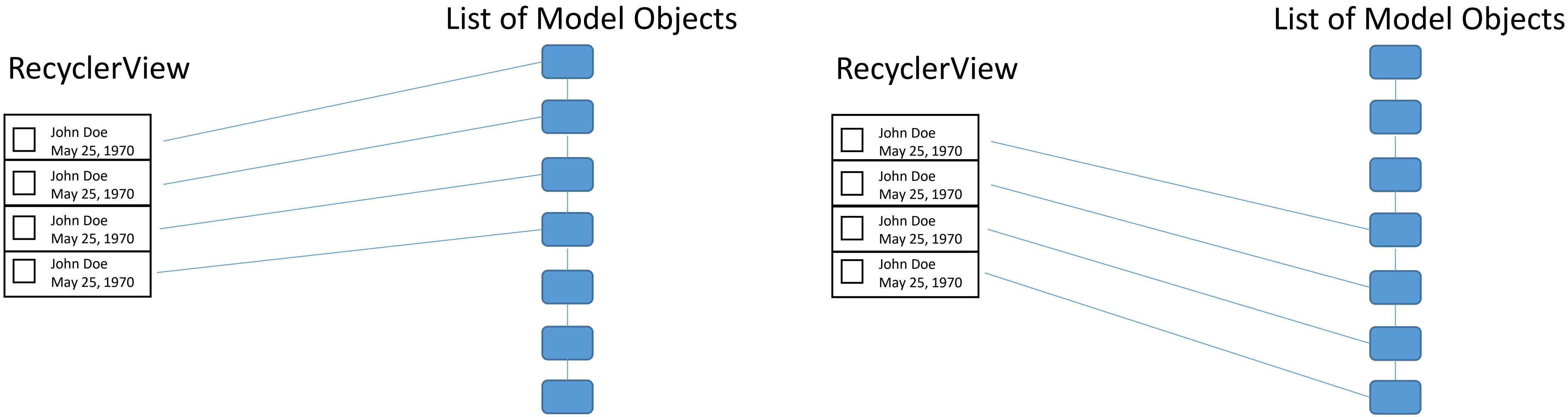
- For this discussion we will focus on List widgets, because that’s what you need for the FMC project
- However, we could have a similar discussion about Tree or Table widgets
- Example: Family Map Search Activity
  - Dynamic list of search results containing both people and events
  - RecylerView is the widget of choice for this
  - See RecyclerViewExample_MultipleViewTypes project on the web site
- Example: Family Map Person Activity
  - Displays list of family members and events in list with expandable sections
  - ExpandableListView is the widget of choice for this
  - See ExpandableListViewExample project on the web site

RecyclerView

- PROBLEM: You have a list of objects you want to display on the screen. The list could be very long (hundreds of items). You might want to let the user edit the objects on the screen.
- RecyclerView can display an arbitrarily long repeating list of ViewGroups

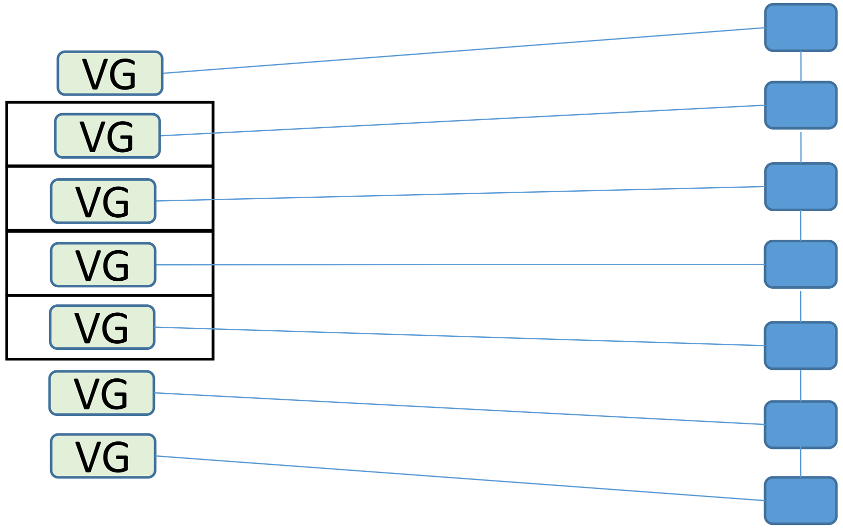


RecyclerView - Scrolling

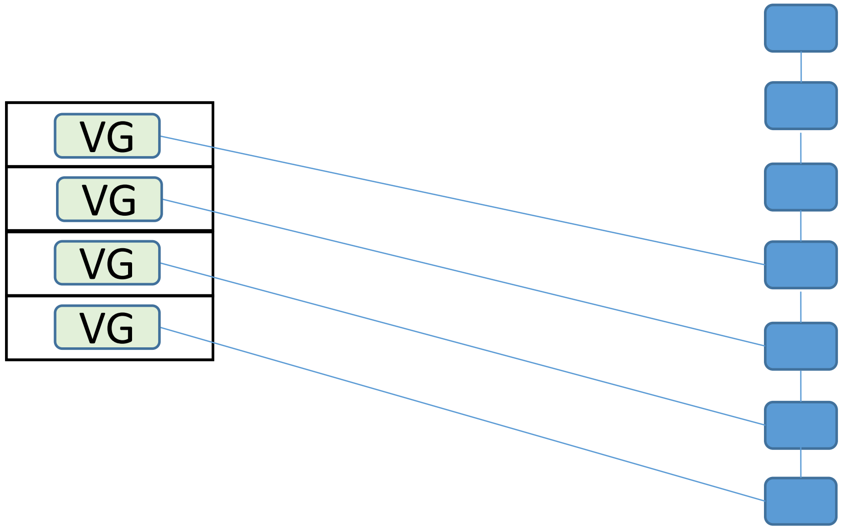


RecyclerView – Scrolling Efficiently

One ViewGroup Per List Object?
(doesn’t scale to large N)

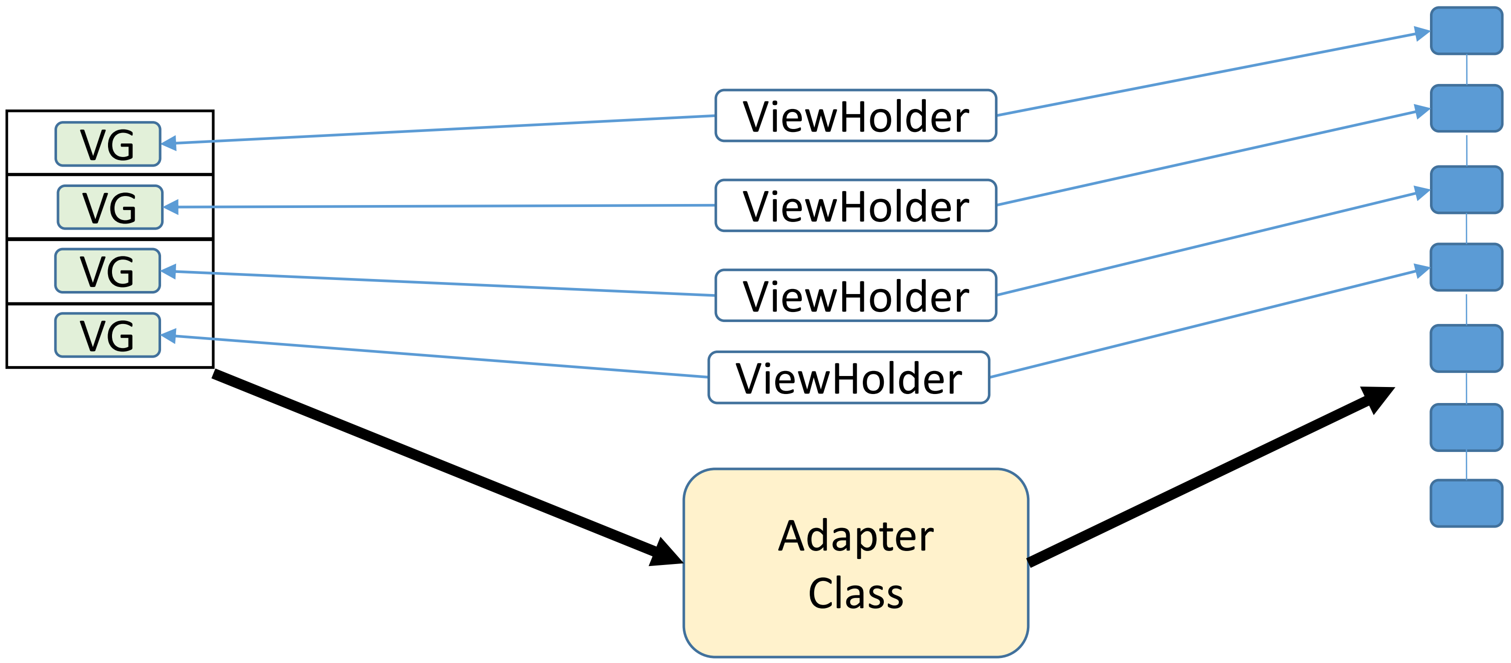


Create only the number of ViewGroups that can be displayed on the screen and recycle them as the user scrolls!

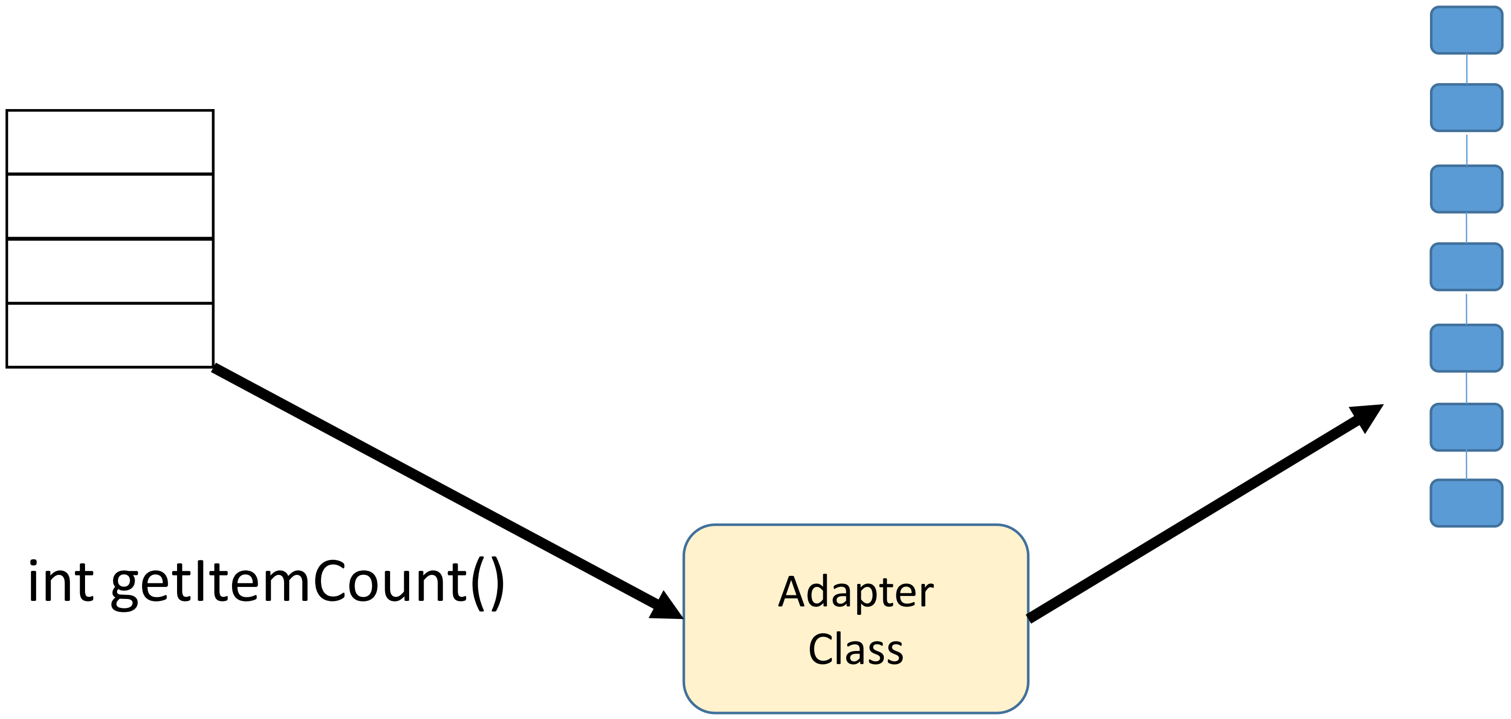


RecyclerView

- PROBLEM: How can RecyclerView display objects it doesn't understand and let the user edit them?
- SOLUTION: Create an Adapter class that translates model objects into a form that RecyclerView can understand



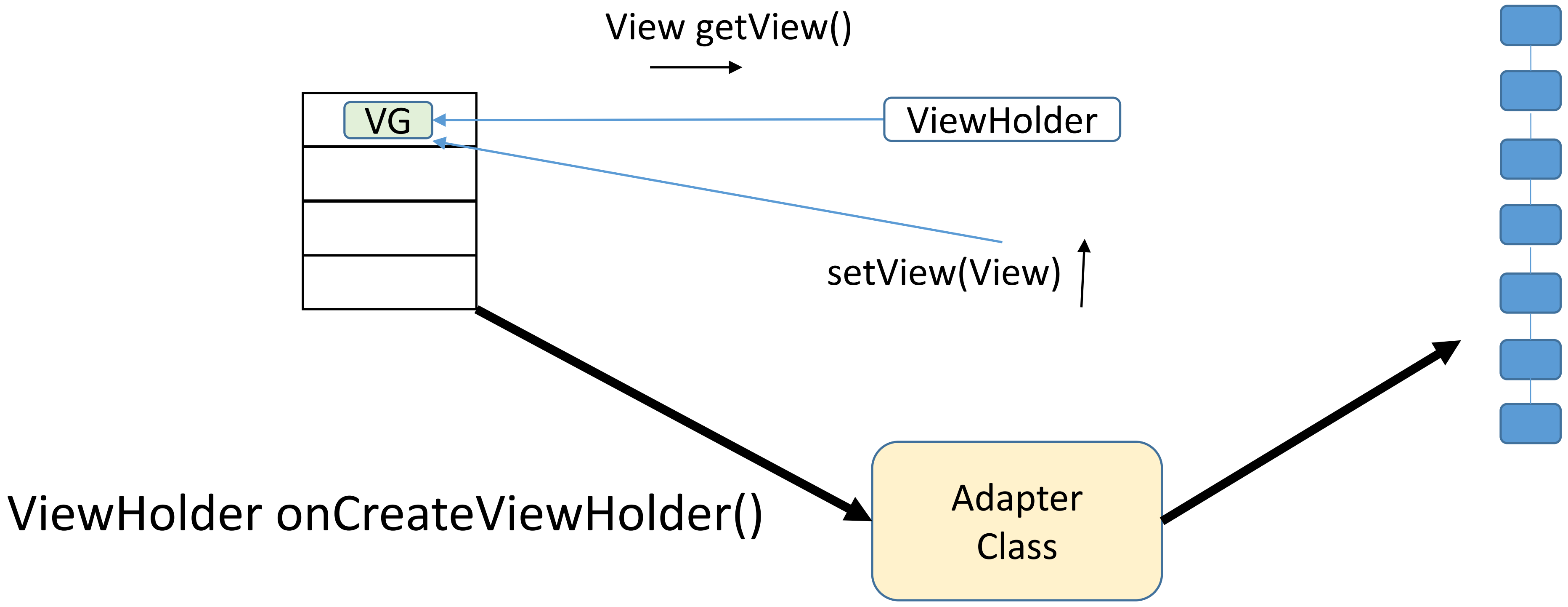
RecyclerViewAdapter Methods



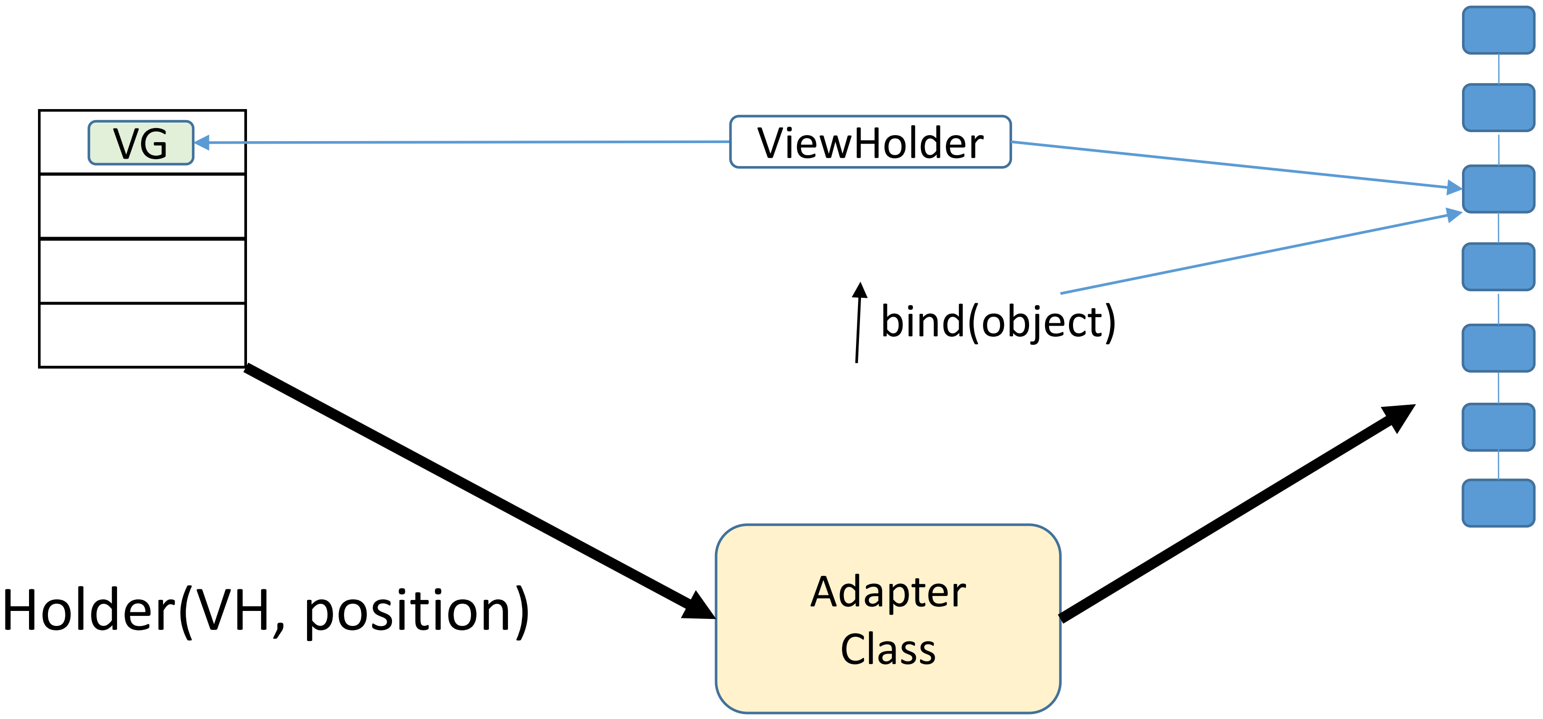
int getItemCount()

Adapter Class

RecyclerViewAdapter Methods



RecyclerViewAdapter Methods



onBindViewHolder(VH, position)

ViewHolders

- ViewHolders have the same structure as Activities and Fragments
- Consist of a Java class and an XML layout file
- Contains (or “holds”) a pointer to the root of its widget tree
- Wires up its widgets
  - Contains widget variables
  - Puts event listeners on widgets
  - Possibly modifies its bound model object in response to user inputs

RecyclerView

- RecyclerViewExample_MultipleViewTypes (on the web site)

ExpandableListView

- Displays a list with multiple expandable/contractable sections
  - Family Map Person Activity
- ExpandableListView is very similar to RecyclerView
  - Uses an Adapter class
  - However, it does not use ViewHolders
- ExpandableListViewExample (on the web site)