* **Quy trình làm việc với listview:**

Xác định nguồn dữ liệu 🡺 Tạo Adapter cùng kiểu dữ liệu với nguồn – gán nguồn 🡺 Gán Adapter cho list view.

* **Tạo layout Item**: Chứa row cho mỗi item list được custom
* **Tạo model**: Khởi tạo các thuộc tính phục vụ cho việc lấy dữ liệu – Kế thừa Serializable
* Khởi tạo các thuộc tính: get, set, constructor
* **Tạo adapter** kế thừa ArrayAdapter<model>
* Nguồn dữ liệu có kiểu gì thì Adapter có kiểu đó
* Khởi tạo constructor (resource = item)
  + Hiệu chỉnh hàm getView() @Overide
  + Custom lại adapter

public class **CustomAdapter** extends ArrayAdapter<[*Kiểu đối tượng*]>{

public **CustomAdapter**(Context context, int resource, List<[*Kiểu đối tượng*]> objects) {

super(context, resource, objects);

}

public **View getView**(int position, View convertView, ViewGroup parent) {

LayoutInflater inflater=**this.**context.getLayoutInflater();*//một lớp dùng để build một layout bt=> một code java android có thể sử dụng được*

View item=inflater.inflate(**this**.resource,null);

//…. Phai lấy hết tất cả giao diện trong row (item) 🡺 lấy txt, btn

TextView txtMuaTM = (TextView) row.findViewById(R.id.***txtMuaTM***);

…

**final** TiGia db = **this**.**objects**.get(position);*//tra ve ti gia hien tai*

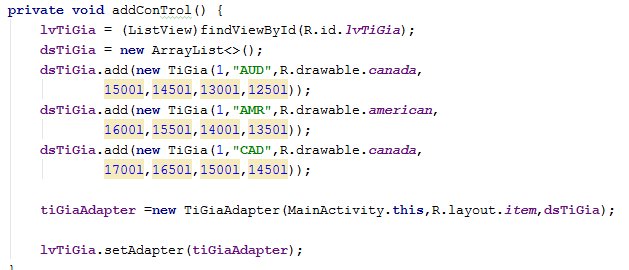
*muon ve*

*…*txtNuoc.setText(db.getNuoc().toString());

**return** item;//row 🡺 quy định giao diện

}

* Khai báo bộ ba listview 🡺 ArrayList(data) 🡺 Adapter 🡺 tiến hành thêm control và event



* Gridview
* Tương tự ListView (Hiển thị dạng ống) 🡺 tạo layout riêng 🡺 lấy hình(mảng dữ liệu kiểu int), dữ liệu…
* Activity.xml🡺 thêm thuộc tính numColumn
* Custom + Overide : 