

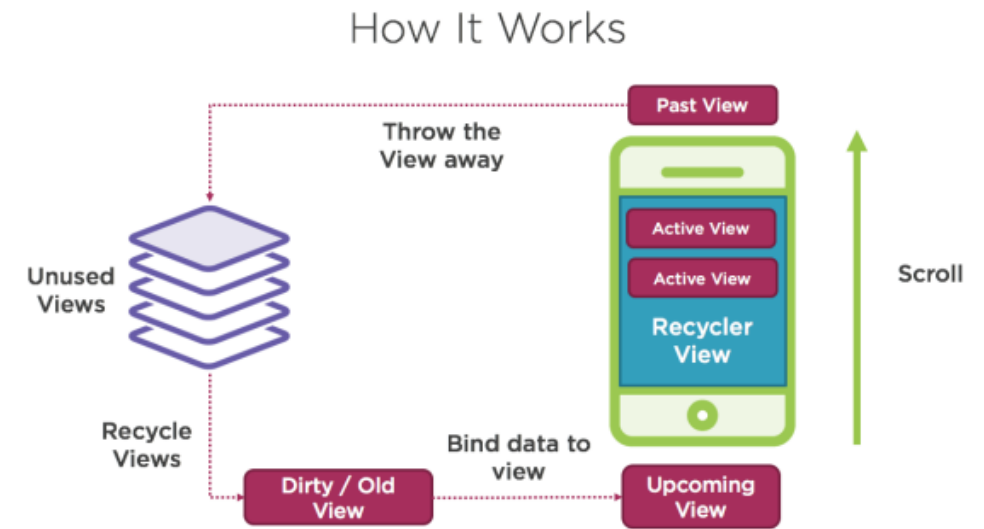
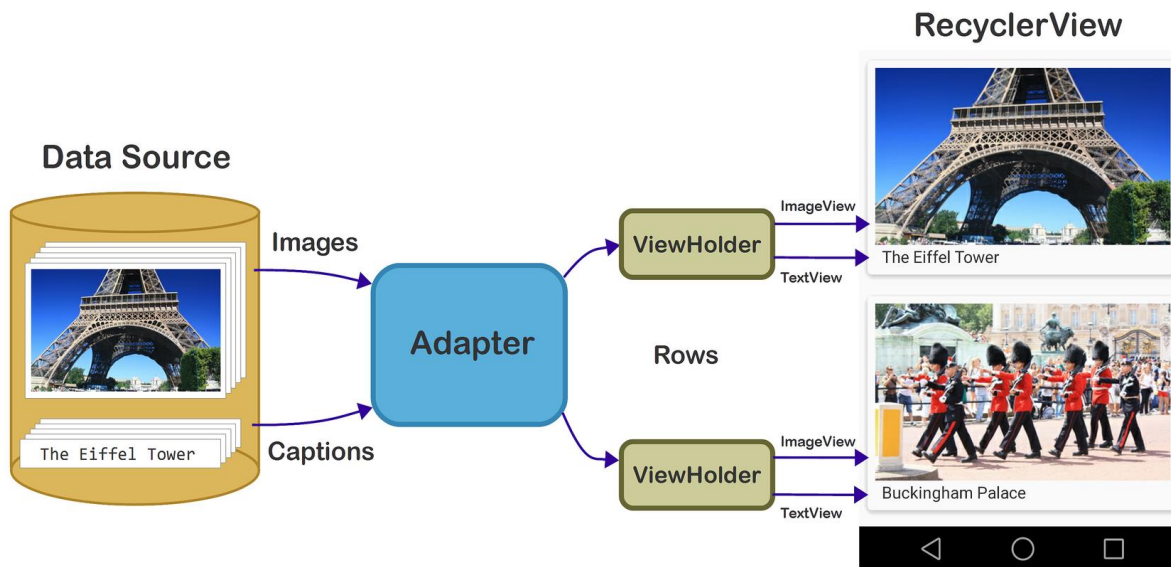


# LẬP TRÌNH THIẾT BỊ DI ĐỘNG

Mai Cường Thọ

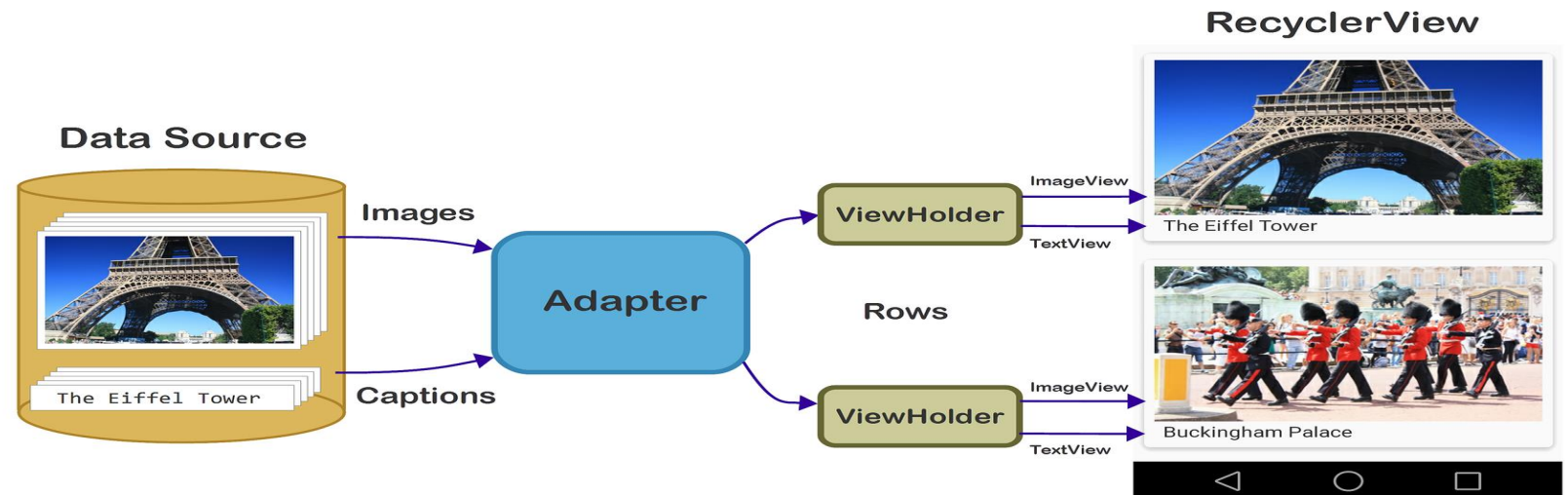
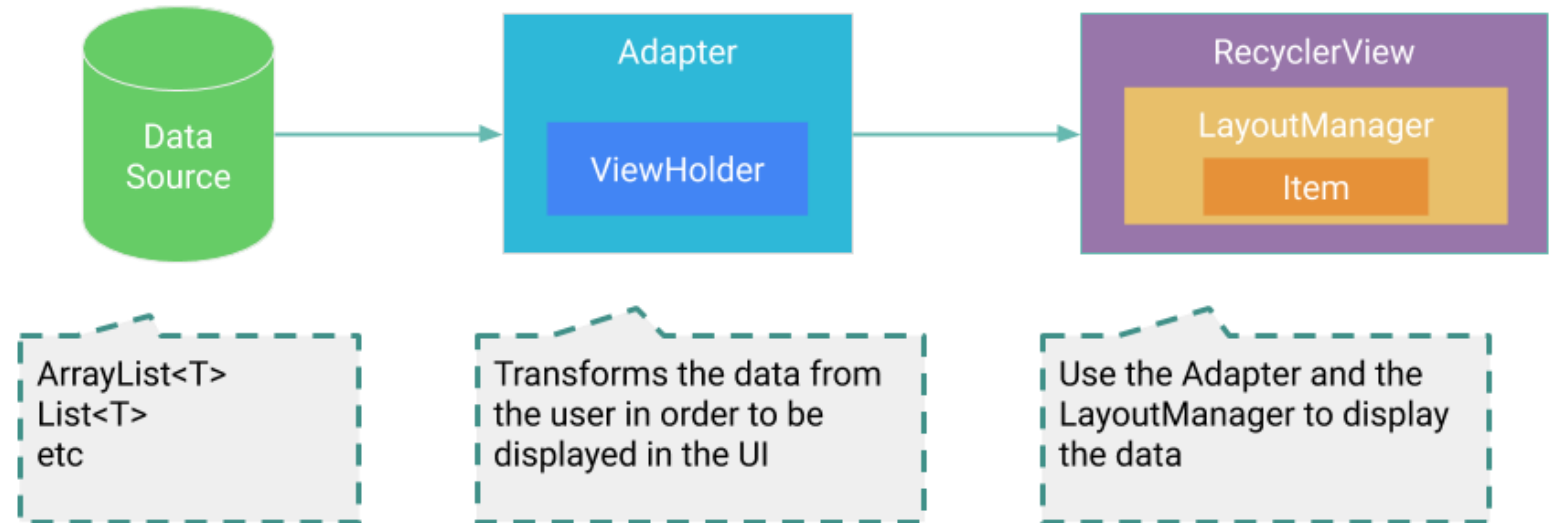
# RecyclerView

- Phiên bản nâng cấp của ListView và GridView:
  - Hỗ trợ các kiểu layout linh hoạt: LinearLayout (Horizontal hoặc Vertical), GridLayout, StaggerdGridLayout, hoặc kiểu layout do người lập trình tự thiết kế.
  - Áp dụng kiểu thiết kế **ViewHolder** để nâng cao hiệu năng.
  - Hỗ trợ các hiệu ứng animation



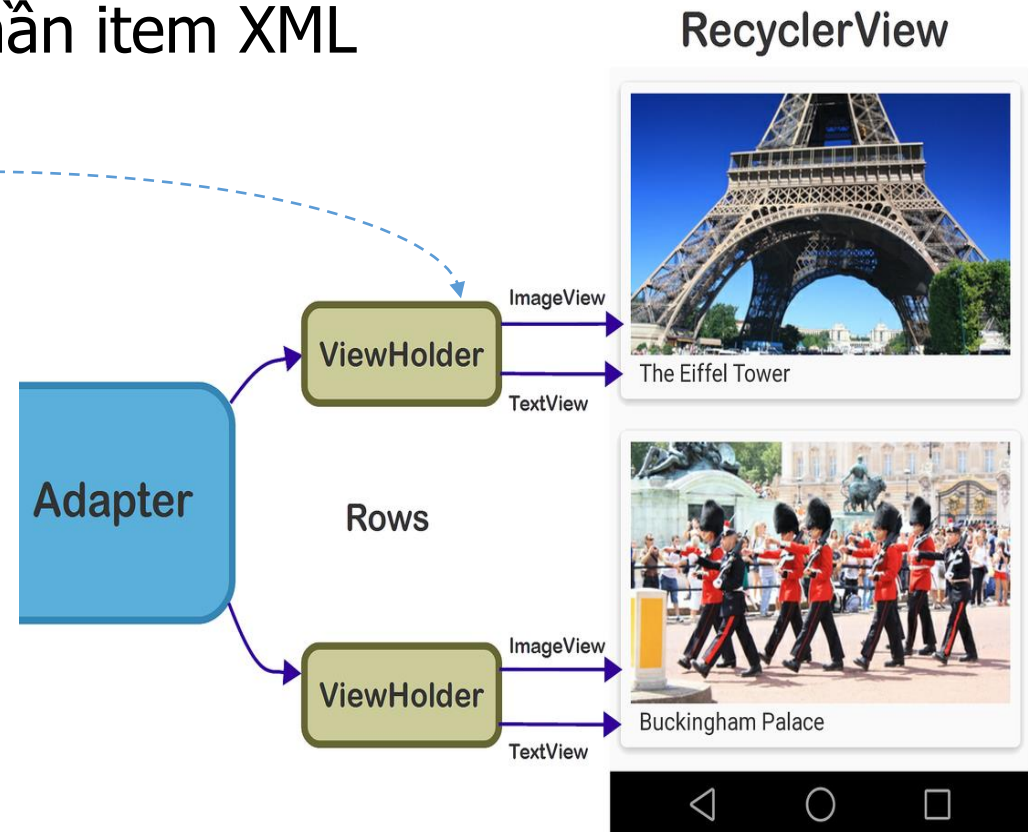
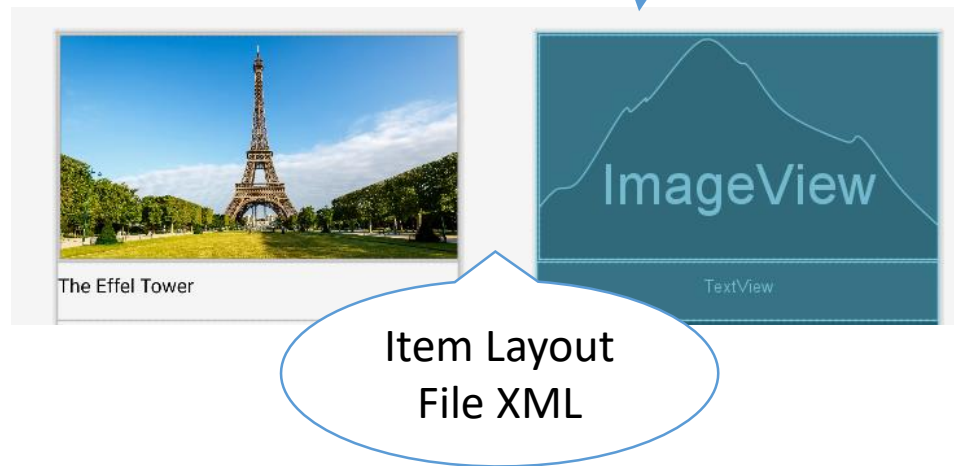
# Các thành phần chính

- RecyclerView.Adapter
- RecyclerView.ViewHolder
- RecyclerView.LayoutManager



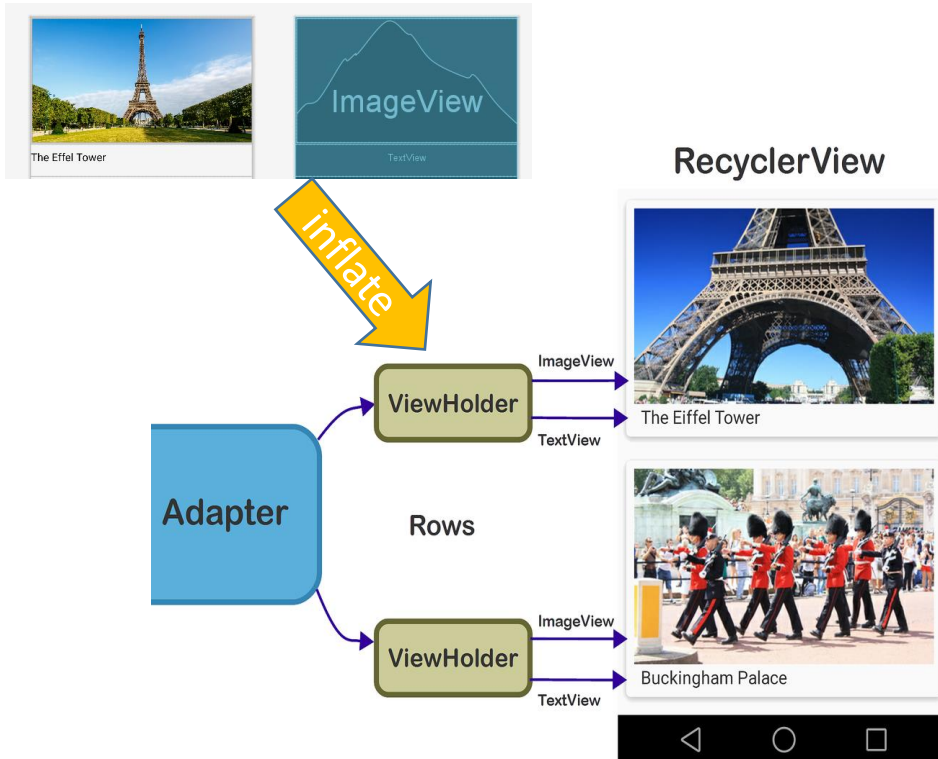
# RecyclerView.ViewHolder

- **ViewHolder** là một **lớp** java, **đại diện** cho mỗi **view item/row** của danh sách
- Mỗi thuộc tính của lớp – một thành phần item XML
- Lớp nội của adapter



# RecyclerView.Adapter

- Là một lớp trừu tượng
- Với các phương thức:



```
public RecyclerView.ViewHolder onCreateViewHolder(..) { ...}
```

- Thực hiện việc inflate bố cục thiết kế của item (trên file **item layout XML** sang → **lớp Viewholder**)
- Trả về một đối tượng của lớp Viewholder này

```
public void onBindViewHolder(...) {...}
```

- Đầu vào:
  - ✓ Đối tượng đã tạo được ở **onCreateViewHolder**
  - ✓ Vị trí phần tử
- Thực hiện gán giá trị cho các view con của item/viewholder đã tạo dựa vào vị trí

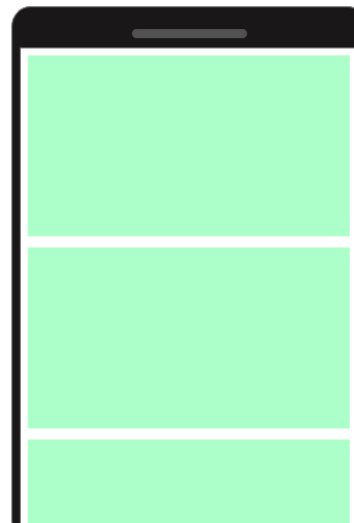
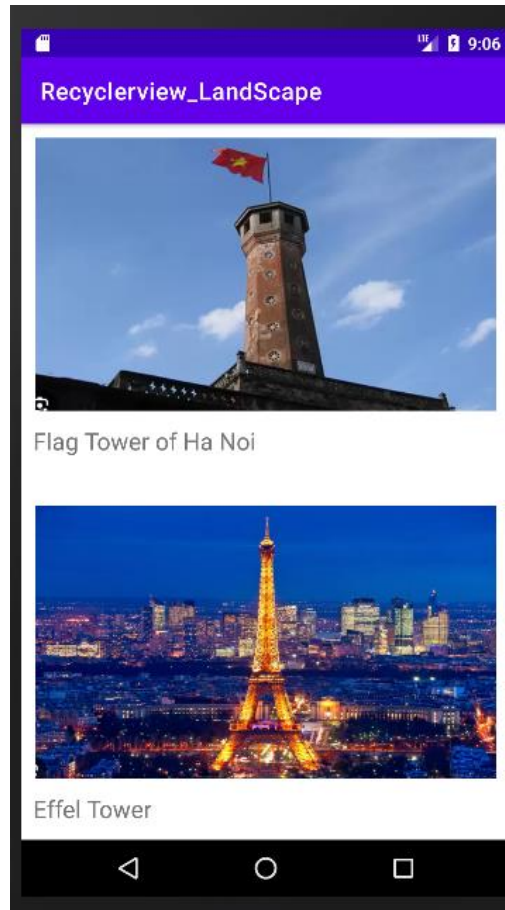
```
public int getItemCount() {}: trả về số lượng phần tử của danh sách
```



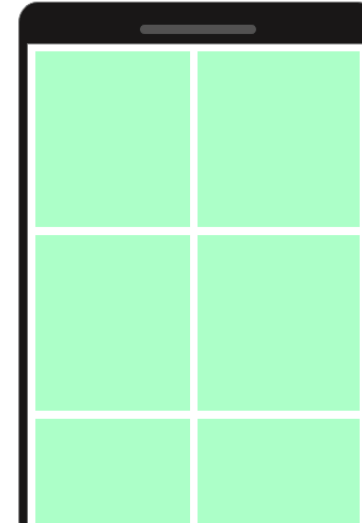
# LayoutManager



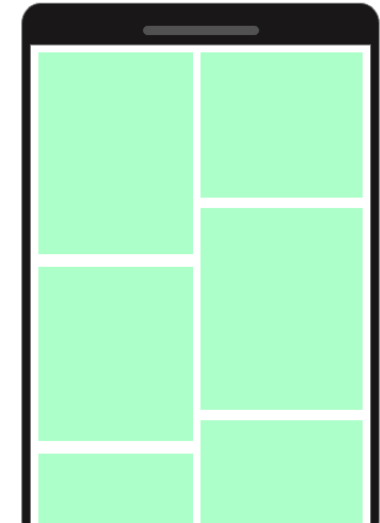
- Lớp cho phép ta qui định bố cục hiển thị cụ thể cho RecyclerView



LinearLayout



GridLayout



StaggerdGridLayout

# Làm việc với RecyclerView



## ■ Bước 1:

- Xây dựng model class (cấu trúc dữ liệu) cho mỗi item của nguồn dữ liệu

```
public class Landscape {  
    private String landscapeName;  
    private String landscapeImage;  
    // Các getter/setter/constructor  
    // ..  
}
```

## ■ Bước 2: Res/Layout:

- Tạo XML layout file và **thiết kế** bố cục cho **một Item** của RecyclerView

Ví dụ: landscap\_item.xml



- Đặt RecyclerView vào activity hoặc fragment muốn hiển thị  
Ví dụ: đặt vào activity\_main.xml

# Làm việc với RecyclerView ..

- **Bước 3:** Tạo lớp adapter, kế thừa lớp RecyclerView.Adapter
  - **3.a.** Xây dựng lớp ViewHolder cụ thể kế thừa lớp RecyclerView.ViewHolder trong lớp này, trước khi implements các phương thức phải ghi đè của RecyclerView.Adapter.

```
public class ItemLandscapAdapter extends RecyclerView.Adapter<ItemLandscapAdapter.ItemLandscapViewHolder> {  
    @NonNull  
    @Override  
    public ItemLandscapViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) { return null; }  
    @Override  
    public void onBindViewHolder(@NonNull ItemLandscapViewHolder holder, int position) { }  
    @Override  
    public int getItemCount() { return 0; }  
    3 usages  
    static final class ItemLandscapViewHolder extends RecyclerView.ViewHolder {  
        TextView textViewLandName;  
        ImageView imageViewLandPicture;  
        public ItemLandscapViewHolder(@NonNull View itemView) { super(itemView); }  
    }  
}
```





# Làm việc với RecyclerView ..

## ■ Bước 3: ..

- **3.a.** tiếp: lập trình cho hàm tạo của lớp viewholder, nhằm tham chiếu đến các của các thành phần của item layout XML.

```
static final class ItemLandscapViewHolder extends RecyclerView.ViewHolder {  
    1 usage  
    TextView textViewLandName;  
    1 usage  
    ImageView imageViewLandPicture;  
    public ItemLandscapViewHolder(@NonNull View itemView) {  
        super(itemView);  
        imageViewLandPicture= itemView.findViewById(R.id.imLandscap);  
        textViewLandName = itemView.findViewById(R.id.tvLandscapName);  
    }  
}
```



item layout XML

# Làm việc với RecyclerView ..

## ■ Bước 3: ..

- **3.b.** Mỗi adapter cần có context và dữ liệu để hoạt động, do vậy ta khai báo thêm 2 tham số này của lớp adapter, và cài đặt hàm tạo cho nó

```
public class ItemLandscapAdapter extends RecyclerView.Adapter<ItemLandscapAdapter.Item
```

```
1 usage
```

```
Context context;
```

```
1 usage
```

```
ArrayList<LandScape> datas;
```

```
public ItemLandscapAdapter(Context _context, ArrayList<LandScape> _datas) {
```

```
    this.context = _context;
```

```
    this.datas = _datas;
```

```
}
```

```
@NonNull
```

```
@Override
```

# Làm việc với RecyclerView ..



## ■ Bước 3: ..

- **3.c.** lần lượt code các phương thức cần phải ghi đè của Adapter

```
public RecyclerView.ViewHolder onCreateViewHolder(..) { ...}
```

- Thực hiện việc inflate bố cục thiết kết của item (trên file **item layout XML** sang → **lớp Viewholder**)
- Trả về một đối tượng của lớp Viewholder này

```
public void onBindViewHolder(...) {...}
```

- Đầu vào:
  - ✓ Đối tượng đã tạo được ở **onCreateViewHolder**
  - ✓ Vị trí phần tử
- Thực hiện gán giá trị cho các view con của item/viewholder đã tạo dựa vào vị trí

```
public int getItemCount() {}: trả về số lượng phần tử của danh sách
```

# Làm việc với RecyclerView ..



## ■ Bước 3: ..

### ▪ 3.c\_1.

```
public RecyclerView.ViewHolder onCreateViewHolder(..) { ...}
```

- Thực hiện việc inflate bố cục thiết kế của item (trên file **item layout XML** sang → **lớp Viewholder**)
- Trả về một đối tượng của lớp ViewHolder này

@Override

```
public ItemLandscapeViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
    LayoutInflater inflater = LayoutInflater.from(context);  
    View viewItem = inflater.inflate(R.layout.landscape_item, parent, attachToRoot: false);  
    // Tạo đối tượng holder từ view inflate được, thông qua hàm khởi tạo của lớp ViewHolder  
    ItemLandscapeViewHolder viewHolder = new ItemLandscapeViewHolder(viewItem);  
    return viewHolder;  
}
```

# Làm việc với RecyclerView ..

## ■ Bước 3: ..

### ■ 3.c\_2.

```
public void onBindViewHolder(...) {...}
```

- Đầu vào:
  - ✓ Đối tượng đã tạo được ở `onCreateViewHolder`
  - ✓ Vị trí phần tử
- Thực hiện gán giá trị cho các view con của item/viewholder đã tạo dựa vào vị trí

```
@Override
```

```
public void onBindViewHolder(@NonNull ItemLandscapeViewHolder holder, int position) {
```

```
    // Lấy phần tử ở vị trí position của nguồn dữ liệu
```

```
    Landscape land = datas.get(position);
```

```
    // Đặt vào các thuộc tính hiển thị của view con
```

```
    // Đặt tên
```

```
    holder.textViewLandName.setText(land.getLandscapeName());
```

```
    // Đặt ảnh (ở đây ta giả sử các file ảnh đặt ở mục res/mipmap
```

```
        String packageName = holder.itemView.getContext().getPackageName();
```

```
    // lấy id cho ảnh thông qua tên
```

```
    String tenFileAnh=land.getLandscapeImage();
```

```
    int imageID = holder.itemView.getResources().getIdentifier(tenFileAnh, defType: "mipmap", packageName);
```

```
    holder.imageViewLandPicture.setImageResource(imageID);
```

```
}
```

```
@Override
```

```
public int getItemCount() {    return datas.size(); }
```



# Làm việc với RecyclerView ..



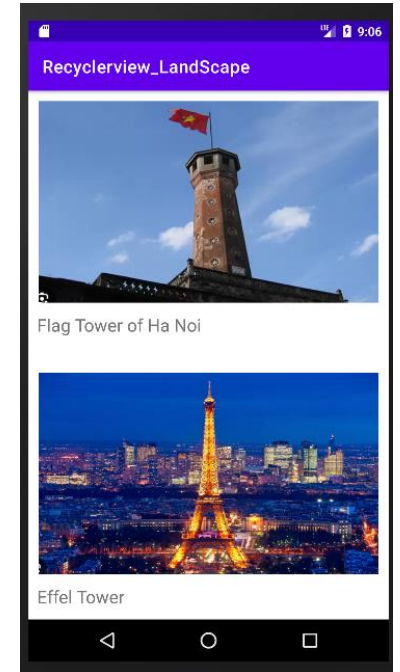
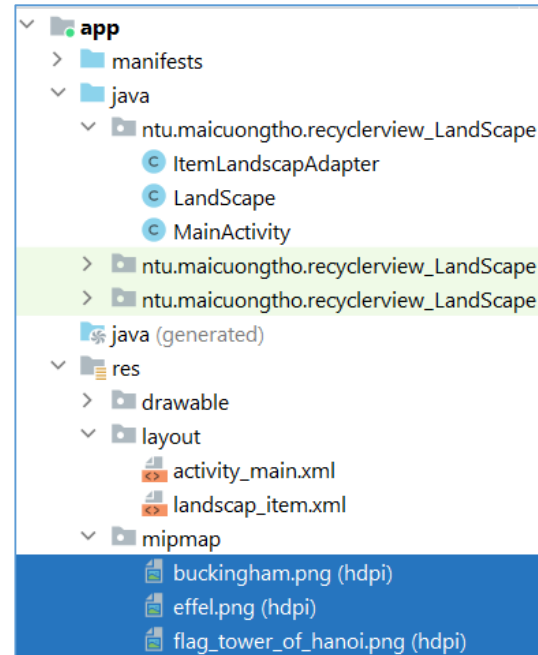
- **Bước 4:**  
khởi tạo và cài đặt  
tại  
activity hiển thị

```
public class MainActivity extends AppCompatActivity {  
    //1. Khai báo các biến truy cập toàn cục  
    ItemLandscapeAdapter adapter;           //adapter  
    ArrayList<Landscape> landscapeList; // nguồn dữ liệu  
    //2. Khai báo các biến đại diện cho các view cần tương tác  
    RecyclerView recyclerViewLand;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        //3. Chuẩn bị dữ liệu cho landscapeList  
        //4. Tìm điều khiển RecyclerView  
        //5. Tạo layout manager để đặt bố cục cho RecyclerView  
        //6. Tạo adapter gắn với nguồn dữ liệu  
        //7. Gắn adapter với RecyclerView  
    }  
}
```

# Làm việc với RecyclerView ..

## ■ Bước 4://3

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    //3. Chuẩn bị dữ liệu cho landscapeList  
    landscapeList = getDataForRecycler();  
    //4. Tìm điều khiển Recycler  
    //5. Tạo layout manager để đặt bố cục cho Recycler  
    //6. Tạo adapter gắn với nguồn dữ liệu  
    //7. Gắn adapter với Recycler  
}
```

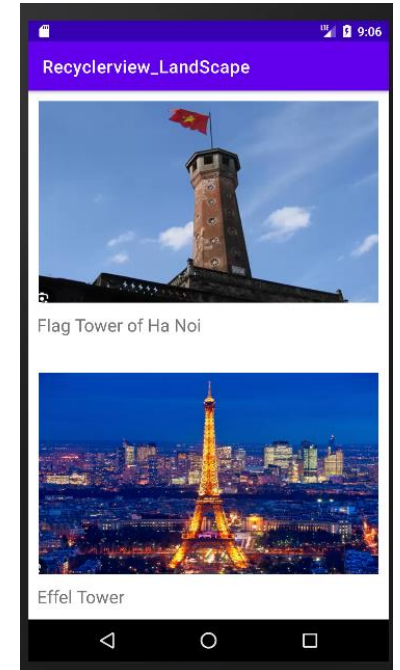


```
private ArrayList<LandScape> getDataForRecycler() {  
    ArrayList<LandScape> listData= new ArrayList<LandScape>();  
    listData.add( new LandScape( landscapeName: "Flag Tower of Ha Noi", landscapelImage: "flag_tower_of_hanoi"));  
    listData.add( new LandScape( landscapeName: "Eiffel Tower", landscapelImage: "effel"));  
    listData.add( new LandScape( landscapeName: "Buckingham Palace", landscapelImage: "buckingham"));  
    return listData;  
}
```

# Làm việc với RecyclerView ..

## ■ Bước 4://4-7;

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    //3. Chuẩn bị dữ liệu cho landScapeList  
    landScapeList = getDataForRecycler();  
    //4. Tìm điều khiển Recycler  
    recyclerViewLand = findViewById(R.id.recyclerLand);  
    //5. Tạo layout manager để đặt bố cục cho Recycler  
    RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(context, this);  
    recyclerViewLand.setLayoutManager(layoutManager);  
    //6. Tạo adapter gắn với nguồn dữ liệu  
    adapter = new ItemLandscapAdapter(_context, this, landScapeList);  
    //7. Gắn adapter với Recycler  
    recyclerViewLand.setAdapter(adapter);  
}
```



# Lập trình với RecyclerView .. Xử lý sự kiện



- **Bước 5: Xử lý sự kiện** khi item (viewholder) được nhấn
  - Có thể thực hiện qua việc **implements** bộ lắng nghe **OnClickListener**

```
final class ItemLandscapeViewHolder extends RecyclerView.ViewHolder
    implements View.OnClickListener{

    2 usages
    TextView textViewLandName; ImageView imageViewLandPicture;
    1 usage
    public ItemLandscapeViewHolder(@NonNull View itemView) {
        super(itemView);
        itemView.setOnClickListener(this);
        imageViewLandPicture= itemView.findViewById(R.id.imLandscape);
        textViewLandName = itemView.findViewById(R.id.tvLandscapeName);
    }
    @Override
    public void onClick(View v) {
```

# Lập trình với RecyclerView .. Xử lý sự kiện



- **Bước 5:** Xử lý sự kiện (tiếp)..

```
@Override
public void onClick(View v) {
    // Lấy vị trí item được click thông qua phương thức getAdapterPosition()
    int clickedPosition = getAdapterPosition();
    // Lấy dữ liệu tương ứng từ danh sách (theo vị trí)
    Landscape land = datas.get(clickedPosition);
    // Hiện thông báo, hoặc thực hiện các xử lý khác
    Toast.makeText(v.getContext(), text: "Clicked: " + land.getLandscapeName(),
        Toast.LENGTH_SHORT).show();
}
```



# Sử dụng thêm Cardview để trình bày Item layout

```
android:layout_height="wrap_content" >  
<androidx.cardview.widget.CardView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    app:cardBackgroundColor="#CCF3B8"  
    app:cardCornerRadius="10dp"  
    app:cardElevation="4dp"  
    app:cardUseCompatPadding="true"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent">  
  
    <androidx.constraintlayout.widget.ConstraintLayout  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content">  
        <ImageView...>  
        <TextView...>  
    </androidx.constraintlayout.widget.ConstraintLayout>  
</androidx.cardview.widget.CardView>  
</androidx.constraintlayout.widget.ConstraintLayout>
```

