

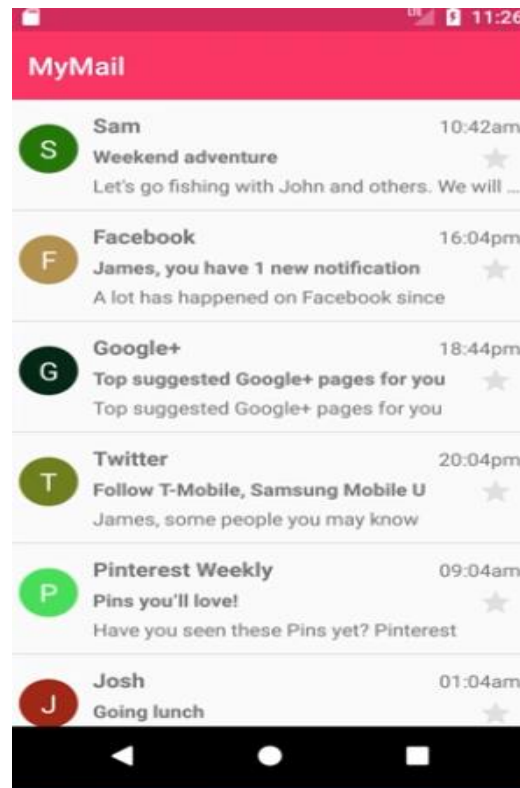
# **DEZVOLTAREA APLICAȚIILOR MOBILE**

**Conf.univ.dr. Ana Cristina DĂSCĂLESCU**

**Universitatea Titu Maiorescu**

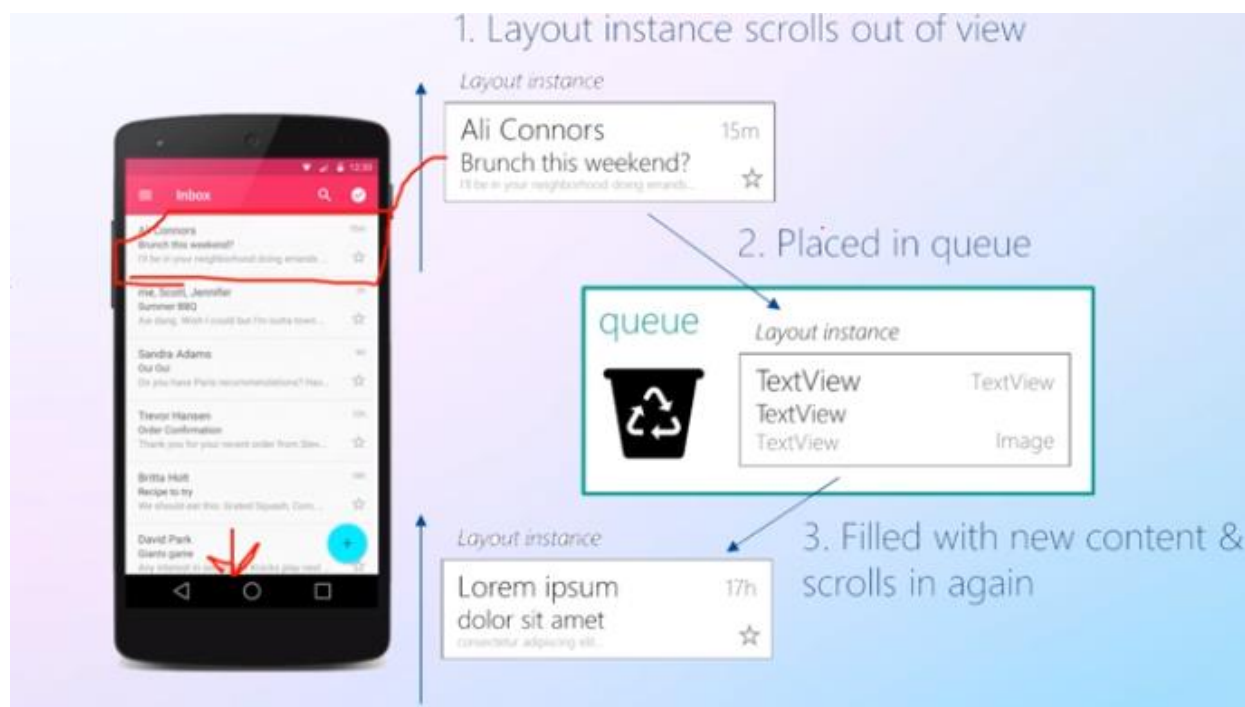
# Recycler VIEW

- **RecyclerView** este o componentă grafică în Android, proiectată pentru a afișa **liste(ListView)** sau **grile de date(Gridview)** în mod eficient.
- Utilitate RecyclerView:
  - Definește modul de vizualizare a unui item



- **Reciclarea Vizualizărilor (View Recycling)**

- Oferă un mecanism inteligent de reciclare a vizualizărilor pentru a optimiza resursele și performanța aplicației.
- Vizualizările care nu mai sunt vizibile sunt reutilizate pentru a afișa elemente noi, minimizând astfel utilizarea memoriei.



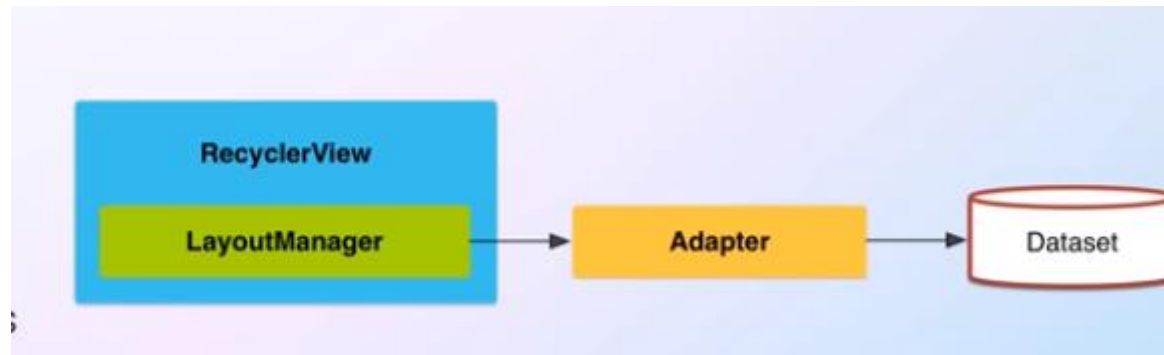
## ➤ Definirea unui RecyclerView

- Se definește un **layout** pentru a stabili modalitatea de vizualizare a elementelor dintr-un singur item
- Se definește o componentă grafică **RecyclerView** în cadrul activității

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/recyclerView"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

- Se definește o **clasă JAVA model** (Model class) cu rolul de structură pentru informațiile care sunt încapsulate într-un item (textView, image etc)

- Se definește un adaptor personalizat: o clasă de adaptor care extinde **RecyclerView.Adapter**.
- Această clasă gestionează datele și definește vizualizările pentru fiecare element din listă.



- Se definește clasa Java **ViewHolder**: are rolul de a reține referințele definite pentru fiecare item (TextView, ImageView etc)

```
public class MyViewHolder extends
    RecyclerView.ViewHolder implements View.OnClickListener
{

    ImageView imageView;
    TextView title;
    TextView description;

    //Constructor
}
```

- În cadrul unei clase de adaptor pentru **RecyclerView**, există câteva metode abstracte esențiale care sunt redefinite în clasa Adaptor. Aceste metode sunt responsabile pentru **crearea și gestionarea vizualizărilor în listă**

**onCreateViewHolder(ViewGroup parent, int viewType):**

- ✓ metodă este apelată atunci când RecyclerView are nevoie să creeze un nou obiect **ViewHolder**. De obicei, în această metodă se infla layout-ul pentru un element individual și se creează o instanță a clasei **ViewHolder**.

```
public MyViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {

View itemView =
LayoutInflater.from(parent.getContext()).inflate(R.layout.i
tem_layout, parent, false);

return new MyViewHolder(itemView);}
```

- `onBindViewHolder(MyViewHolder holder, int position):`

- Această metodă este apelată atunci când RecyclerView trebuie să legă un obiect ViewHolder existent la datele de la o anumită poziție. Aici, veți seta datele specifice pentru elementul din listă.

```
@Override  
  
public void onBindViewHolder(MyViewHolder holder, int  
position) {  
  
    ItemModel item = dataList.get(position);  
  
    holder.textView.setText(item.getItemName());  
  
    // Alte acțiuni pentru a seta imaginea sau alte attribute  
    ale elementului  
  
}
```



- `getItemCount()` :

- Această metodă returnează numărul total de elemente din setul de date al RecyclerView.

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

