DEZVOLTAREA APLICAȚIILOR MOBILE

Conf.univ.dr. Ana Cristina DĂSCĂLESCU

Universitatea Titu Maiorescu

- **CardView** este o componentă grafică în Android care face parte din suita **`androidx.cardview.widget`**.

- Această bibliotecă oferă un container pentru a afișa informații într-un mod structurat, asemănător cu o carte de vizită sau un card fizic.

- CardView oferă o notă estetică și funcțională prin adăugarea umbrelor, a efectelor de ridicare și a altor detalii de design.

➢ Principalele caracteristici ale CardView

- *Umbră și efect de ridicare*: CardView adaugă automat umbre și un efect de ridicare elementelor sale, oferind o senzație de adâncime și separare față de restul interfeței.
  - ✓ `app:cardElevation`

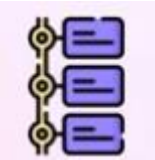
    - această proprietate determină adâncimea umbrei și a efectului de ridicare a cardului. Cu cât valoarea este mai mare, cu atât umbra și efectul de ridicare sunt mai pronunțate
    -

- *Rotunjirea colțurilor*: CardView oferă opțiuni pentru a rotunji colțurile cardului, permițând personalizarea aspectului vizual.
  - ✓ `app:cardCornerRadius`
    - această proprietate determină cât de rotunjite sunt colțurile cardului. Cu cât valoarea este mai mare, cu atât colțurile sunt mai rotunjite. De exemplu:

```
app:cardCornerRadius="40dp"
app:cardElevation="20dp"
```

➢ Definirea unui CardView

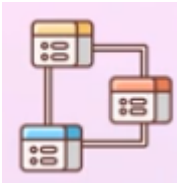
Se definește un **layout** pentru a stabili modalitatea de vizualizare a elemntelor încapsulate într-un card



Se definește o component 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 care extinde **RecyclerView.Adapter**.

- **RecyclerView.Adapter** conține metode pentru a define vizualizarea concretă a fiecărui item, precum gestionarea datelor.

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
```

- 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 lege 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 atribute
    ale elementului
    }
```

- **getItemCount():**

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

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

➢ Implementarea unui ascultător de evenimente pentru RecyclerView

- Definește actiunea care va avea loc la selecția unui element din recycler (asemănător evenimentului pentru ListView)
- Un ascultător recyclerView se definește printr-o interfață ce conține metoda/metodele care se apelează în urma lansării unui eveniment

```
public interface ItemClickListener {
    void onCLickL(View v,int pos);
}
```

- Se definește interfața ascultător în clasa adaptor

```
public class MyAdapter
        extends RecyclerView.Adapter<MyAdapter.MyViewHolder>
        implements View.OnClickListener {

    public ItemClickListener clickListener;
```

- Implementarea se realizează în clasa ViewHolder

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

   public MyViewHolder(@NonNull View itemView) {
     super(itemView);
     . . .

     itemView.setOnClickListener(this);
}
@Override
public void onClick(View v) {
     if (clickListener != null){
         clickListener.onClick(v,getAdapterPosition());
     }
}
```