

# Android fejlesztés

Haladó felhasználói felületi lehetőségek, listák kezelése

[peter.ekler@aut.bme.hu](mailto:peter.ekler@aut.bme.hu)



Department of  
Automation and  
Applied Informatics

# Miről volt szó az előző alkalommal? 😊

- Kotlin emlékeztető
- Alkalmazás komponensek
- Felhasználói felület tervezés és eszközök
- Layout-ok
  - > LinearLayout
  - > RelativeLayout
  - > ConstraintLayout
- Egyszerű View elemek

# Tartalom

- Dinamikus felhasználói felület tervezés
- Stílusok és témák használata
- Animációk
- Listák kezelése – RecyclerView
- CardView – kompakt tartalom megjelenítés
- Dialógusok használata

# DINAMIKUS UI KEZELÉS

# Dinamikus UI kezelés - LayoutInflater

- LayoutInflater feladata:
  - > XML-ben összeállított felületi elemek példányosítása
- Használati mód:

```
val myView = getLayoutInflater().inflate(  
    R.layout.activity_main, null)
```

# STÍLUSOK, TÉMÁK

# Stílusok készítése

- Stílus file: res/values/styles.xml

```
<style name="ExampleStyle">
    <item name="android:textSize">22sp</item>
    <item name="android:textColor">#0000EE</item>
</style>
```

- Stílus használata:

```
<TextView
    android:id="@+id/tvHello"
    android:text="@string/hello_world"
    style="@style/ExampleStyle" />
```

# Saját témák használata

- Stílusokhoz hasonlóan definiálhatók:

```
<style name="CustomTheme" parent="android:Theme">  
    <item name="android:windowTitleSize">50dip</item>  
    <item name="android:textColor">#000000</item>  
    <item name=  
        "android:windowBackground">@color/white_color</item>  
</style>
```

- A témák öröklődhetnek egymásból
- Manifestben állítható be:

```
<activity android:label="" android:name=".MainActivity"  
    android:screenOrientation="portrait"  
    android:theme="@style/CustomTheme"/>
```



# ANIMÁCIÓK

# Animációk

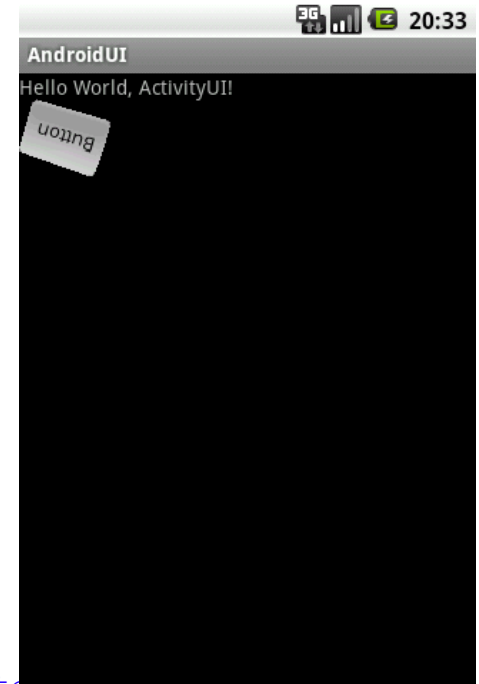
- Animációk támogatása
  - > XML erőforrás (*res/anim*)
  - > Programkód
- Layout animáció
  - > *Scale*
  - > *Rotate*
  - > *Translate*
  - > *Alpha*
- Három fő típus:
  - > Tween animáció
  - > Frame animáció
  - > Property animator

# Tween animáció erőforrás

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <scale
        android:interpolator=
            "@android:anim/accelerate_interpolator"
        android:fromXScale="0.0"
        android:toXScale="1.0"
        android:fromYScale="0.0"
        android:toYScale="1.0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="1000" />

    <alpha android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:duration="5000"/>

    <rotate
        android:interpolator="@android:anim/accelerate_interpolator"
        android:fromDegrees="0.0"
        android:toDegrees="360"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="5000" />
</set>
```



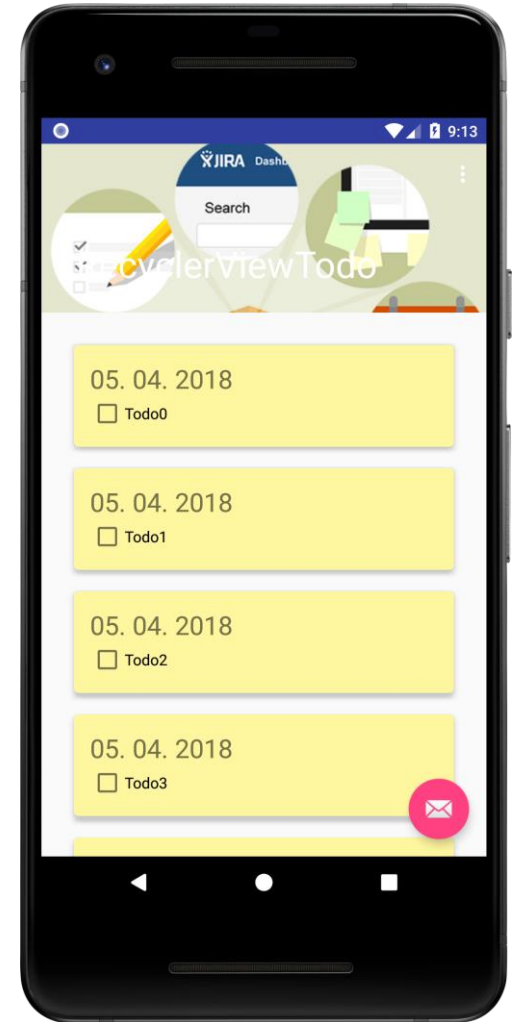
# Tween animáció lejáttszása

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        var demoAnim = AnimationUtils.loadAnimation(this,  
            R.anim.demo_anim)  
  
        btnAnim.setOnClickListener {  
            btnAnim.startAnimation(demoAnim)  
        }  
    }  
}
```

# LISTÁK KEZELÉSE

# RecyclerView

- Listák hatékony kezelése
- Gyors scrollozás
- Általános érintés gesztusok támogatása (swipe, move, stb.)
- *ViewHolder* minta a gyors működés érdekében
- Hatékony elem újrafelhasználás
- *Flexibilis*



# RecyclerView.Adapter<ViewHolder> 1/3

- Inicializálás, konstruktor

```
private val context: Context
private val items: MutableList<ShoppingItem> = mutableListOf<ShoppingItem>(
    ShoppingItem("milk", 200, false),
    ShoppingItem("car", 4000, false),
    ShoppingItem("beer", 1, false)
)

constructor(context: Context) : super() {
    this.context = context
}
```

- Egy sor nézetének beállítása: onCreateViewHolder

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
    val view = LayoutInflater.from(parent.context).inflate(
        R.layout.row_item, parent, false
    )
    return ViewHolder(view)
}
```

# ViewHolder implementáció

```
class ViewHolder(itemView: View?) : RecyclerView.ViewHolder(itemView) {  
    val tvName = itemView.tvName  
    val tvPrice = itemView.tvPrice  
    val cbBought = itemView.cbBought  
    val btnEdit = itemView.btnEdit  
}
```



# RecyclerView.Adapter<ViewHolder> 2/3

- Sorban levő elemek értékeinek beállítása
- Eseménykezelők beállítása
- ViewHolder binding

```
override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
    val (name, price, bought) = items[holder.adapterPosition]  
    holder.tvName.text = name  
    holder.tvPrice.text = price.toString()  
    holder.cbBought.isChecked = bought  
  
    holder.btnEdit.setOnClickListener{  
        (context as MainActivity).showEditTodoDialog(items[holder.adapterPosition])  
    }  
}
```

# RecyclerView.Adapter<ViewHolder> 3/3

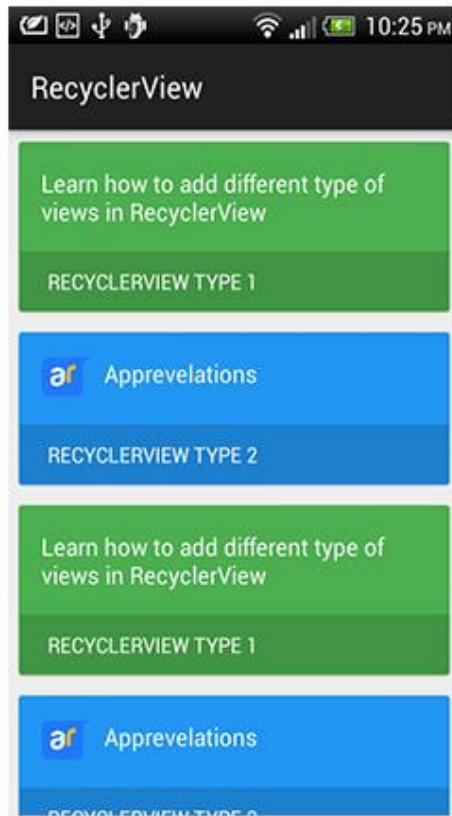
- Elemek száma, hozzáadás, törlés

```
override fun getItemCount() = items.size
```

```
fun addItem(item: ShoppingItem) {  
    items += item  
    notifyItemInserted(items.lastIndex)  
}
```

```
private fun deleteItemBasedOnPosition(position: Int) {  
    items.removeAt(position)  
    notifyItemRemoved(position)  
}
```

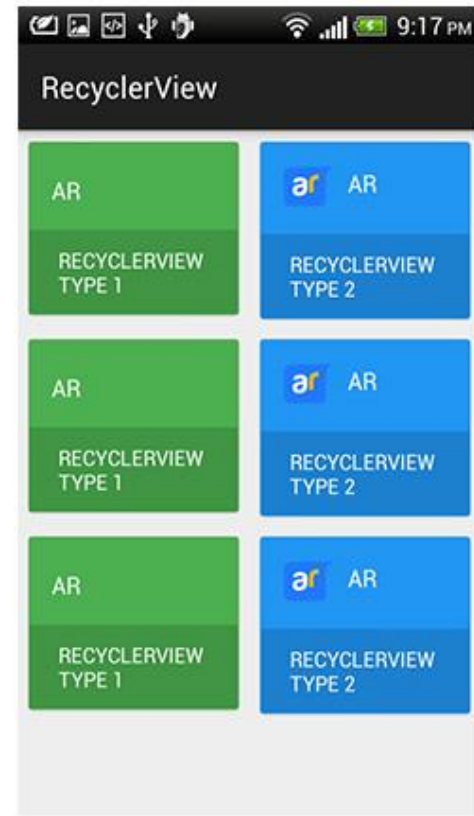
# RecyclerView LayoutManager-ek



Linear Layout View



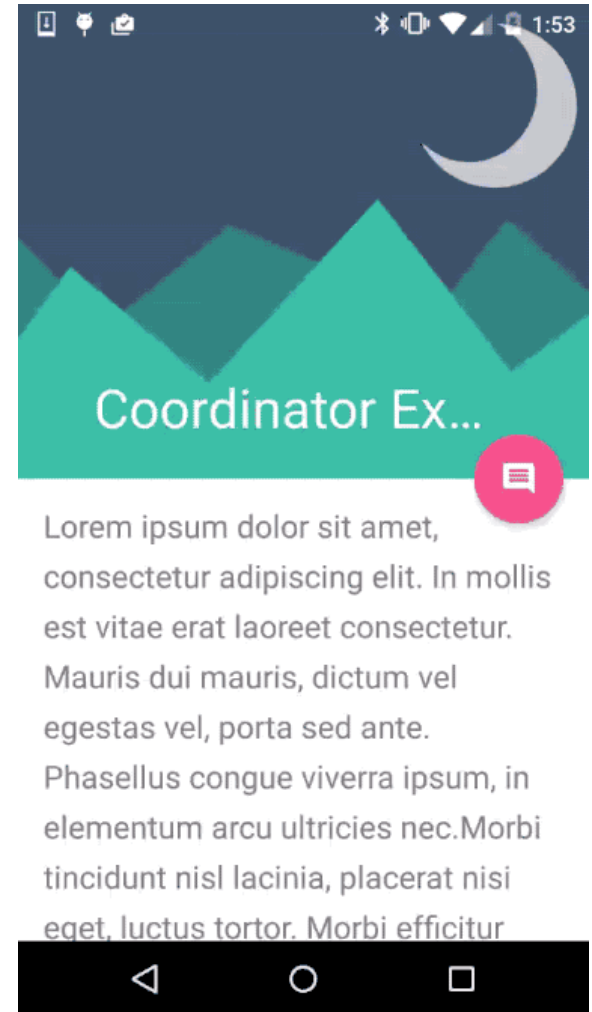
Staggered Grid View



Grid View

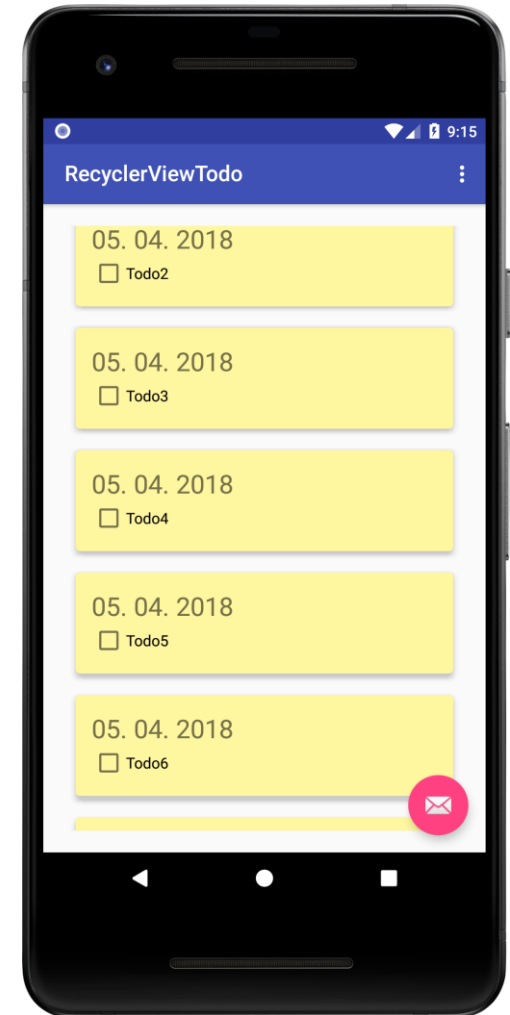
# CoordinatorLayout, AppBarLayout

- *CoordinatorLayout* is a super-powered *FrameLayout*.
- *CoordinatorLayout* is intended for two primary use cases:
  - > As a top-level application decor or chrome layout
  - > As a container for a specific interaction with one or more child views
- By specifying Behaviors for child views of a *CoordinatorLayout* you can provide many different interactions within a single parent and those views can also interact with one another.
- *AppBarLayout* is a vertical *LinearLayout* which implements many of the features of material designs app bar concept, namely scrolling gestures.
  - > This view depends heavily on being used as a direct child within a *CoordinatorLayout*.



# Gyakoroljunk

- Készítsünk egy Todo alkalmazást RecyclerView-val
- Használjunk CheckBox-ot
- Készítsünk „Undo” funkciót, mely elérhet elem létrehozása után néhány másodpercig

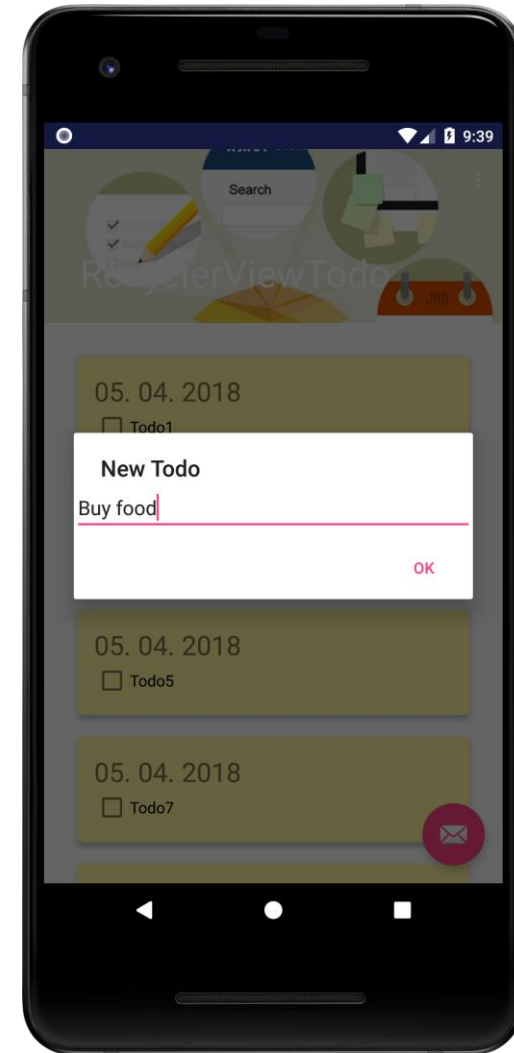


# Lista készítés – fő lépések

1. Data class
2. Egy sor layout-ja
3. RecyclerView – lista hol legyen
4. Adapter – megmondja hogy mi legyen a RecyclerView-ba

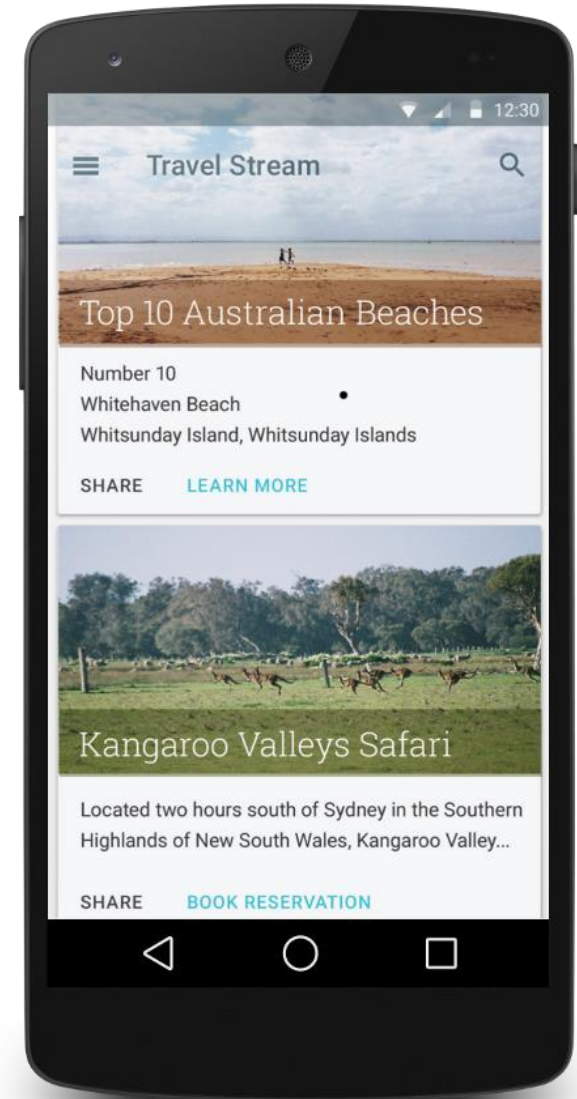
# AlertDialog használata DialogFragment-el – Új Todo dialógus

```
val builder = AlertDialog.Builder(this@MainActivity)
builder.setTitle("Enter Todo")
val input = EditText(this@MainActivity)
builder.setView(input)
builder.setPositiveButton("OK", { dialog, which ->
    todoRecyclerAdapter.addToDo(
        Todo(input.text.toString(), false))
    recyclerTodo.scrollToPosition(0)
})
builder.setNegativeButton("Cancel") {
    dialog, which -> dialog.cancel()
}
builder.show()
```



# CardView bevezetés

- Adat megjelenítés kompakt konténerben
- Gyakran előfordul listákban, de külön is használható

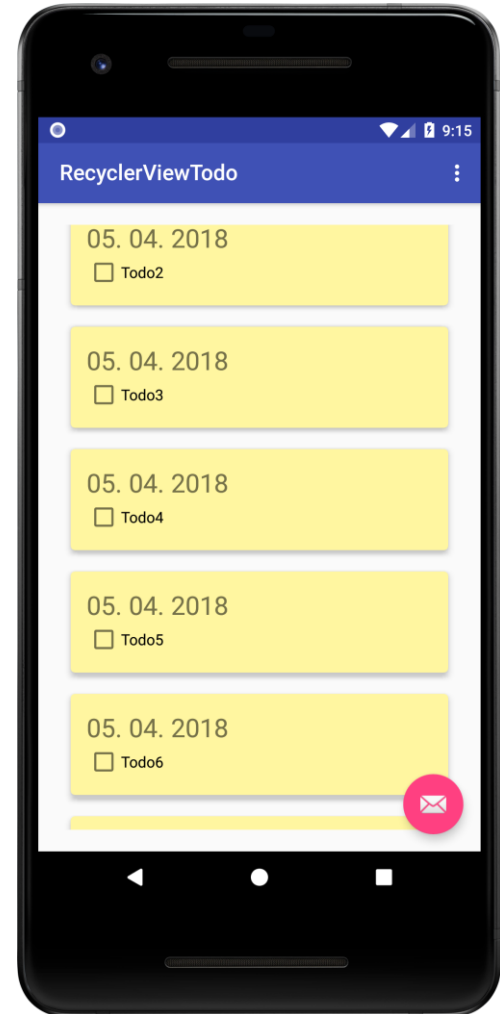




# CardView példa

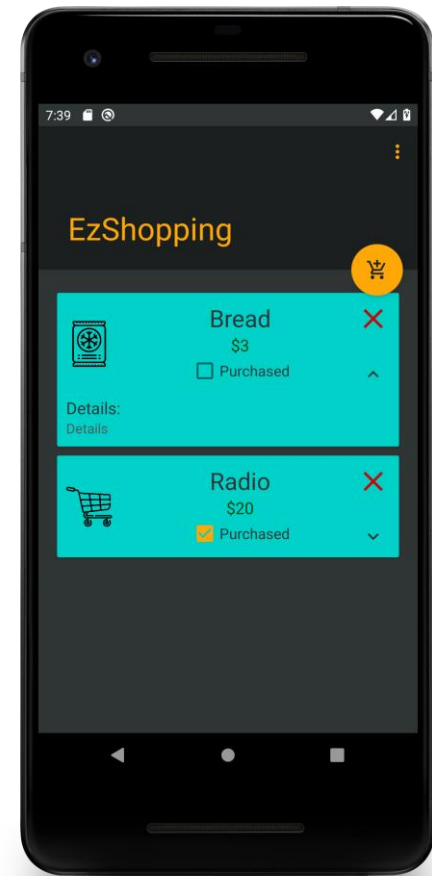
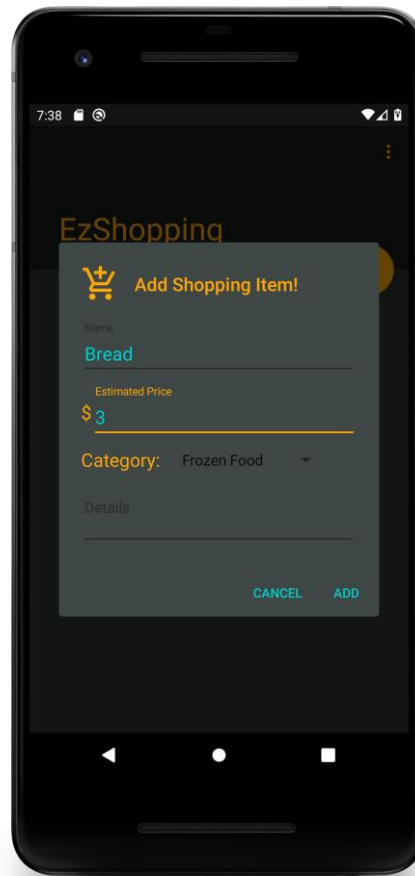
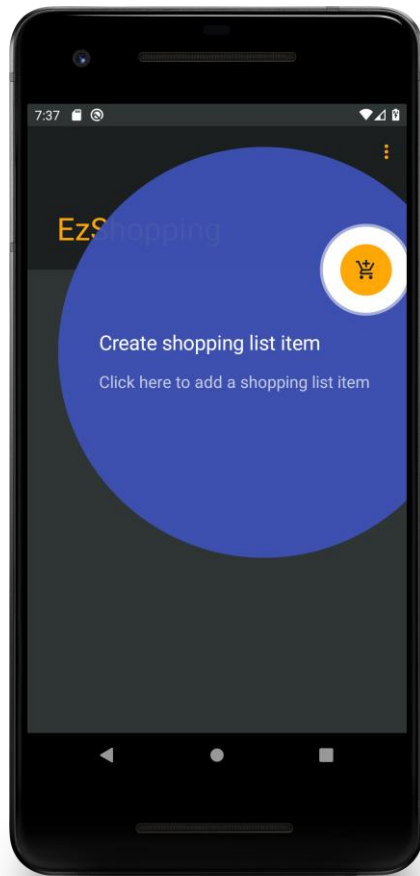
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:id="@+id/card_view"
    android:layout_margin="10dp"
    card_view:cardBackgroundColor="#fff6a0"
    card_view:cardCornerRadius="4dp"
    card_view:cardElevation="4dp">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:id="@+id/tvDate"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="date"
            android:textSize="24sp"/>

        ...
    </LinearLayout>
</android.support.v7.widget.CardView>
```



# Házi feladat

- Bevásárló lista alkalmazás



# Összefoglalás

- Dinamikus felhasználói felület tervezés
- Stílusok és témák használata
- Animációk
- Listák kezelése – RecyclerView
- CardView – kompakt tartalom megjelenítés
- Dialógusok használata

# Köszönöm a figyelmet!



[peter.ekler@aut.bme.hu](mailto:peter.ekler@aut.bme.hu)