

פרק 4 - תכנות דינאמי

מה זה בעצם תכנות דינאמי?

תכנות דינאמי הוא תכנות בו אנחנו יוצרים את האובייקטים דרך JAVA, ולא דרך ה XML.

למה זה טוב?

במידה ונרצה לשים 10 תמונות על המסך, לא ניצור ליצור בצורה ידנית 10 אובייקטים כאלה, אלא ליצור קוד שיעשה זאת בשבילנו.

כך נוסיף תמונה בעזרת תכנות דינאמי:

```
public class MainActivity extends AppCompatActivity {
```

```
    ImageView iv;  
    LinearLayout main;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        iv = new ImageView(this);
```

```
        LinearLayout.LayoutParams layoutParams = new LinearLayout.LayoutParams(100,100);
```

```
        iv.setLayoutParams(layoutParams);
```

```
        main = (LinearLayout)findViewById(R.id.ll1);
```

```
        iv.setImageResource(R.drawable.a0);
```

```
        main.addView(iv);
```

```
    }
```

הצהרות

יצירת תמונה חדשה

ניצור LayoutParams שניתן לו את ערכי ה width ו height שיהיו לתמונה שיצרנו

ניתן לתמונה את ה LayoutParams שיצרנו

נקבל הפניה ל LinearLayout שיצרנו ב XML ושובו תופיע התמונה. אל תשכחו לתת ID ל LinearLayout ב XML

נוסיף את התמונה ל LinearLayout

נקבע איזו תמונה תופיע (העלינו תמונה בשם a0 מהמחשב)

הדרך השנייה להוסיף תמונות בדרך דינאמית היא ליצור פונקציה שמקבלת כפרמטר את מספר התמונות שנרצה להוסיף.

שימו לב שניתן לתמונות את השם a1, a2 וכן הלאה.

נשתמש בפונקציה GetIdentifier בשביל לשלוף את התמונות מתיקיית drawable:

```
public void loadpics1(int n)
{
    for(int i=0;i<n&&i<=7;i++)
    {
        iv = new ImageView(this);
        LinearLayout.LayoutParams layoutParams = new LinearLayout.LayoutParams(100, 100);
        layoutParams.setMargins(0,20,0,0);

        iv.setLayoutParams(layoutParams);
        int imageKey = getResources().getIdentifier("a" + i, "drawable", getPackageName());
        iv.setImageResource(imageKey);
        main.addView(iv);
    }
}
```

העלינו בדוגמה שלנו 7 תמונות, לכן אי אפשר לשלוף יותר מ 7.

ניצור תמונה וניתן לה ערכי אורך ורוחב

קבענו שהשוליים של התמונה יהיו רחוקים מהאובייקט שמעל

נשלוף את התמונה שהעלינו לתיקיית drawable

בדוגמה הבאה נשים את התמונות בתוך HorizontalScrollView שגם אותו ניצור בדרך דינאמית.

שימו לב שכאשר ניצור LinearLayout בתוך HorizontalScrollView שיכיל את התמונות, נצטרך לתת לו את אותו ה LayoutParameters של ה HorizontalScrollView כדי שיתפרש על פני כל השטח שלו.

```
public void loadpics2(int n)
{
    //phase 2
    HorizontalScrollView s= new HorizontalScrollView(this);
    LinearLayout.LayoutParams layoutParams=new LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT);
    s.setLayoutParams(layoutParams);

    LinearLayout linearLayout = new LinearLayout(this);
    linearLayout.setLayoutParams(layoutParams);
}
```

ניצור HorizontalScrollView

ניצור LinearLayout שבו יהיו כל התמונות. בהמשך נכניס אותו ל HorizontalScrollView (למדנו שב scrollView אי אפשר להכניס הרבה אובייקטים, לכן אנחנו יוצרים אובייקט אחד כללי ב ScrollView ובתוכו מכניסים את כל האובייקטים.

```

for(int i=0;i<n;i++)
{
    iv = new ImageView(this);
    LinearLayout.LayoutParams ivparams = new LinearLayout.LayoutParams(100, 100);
    ivparams.setMargins(20,0,0,0);
    iv.setLayoutParams(ivparams);

    int imageKey = getResources().getIdentifier("a"+i, "drawable", getPackageName());
    iv.setImageResource(imageKey);
    linearLayout.addView(iv);
}
s.addView(linearLayout);
main.addView(s);
}

```

בכל פעם ניצור תמונה חדשה, ניתן לה אחת מהתמונות שהעלינו, ונכניס אותה ל LinearLayout.

לבסוף נכניס את ה LinearLayout ל HorizontalScrollView שיצרנו

נכניס את ה HorizontalScrollView ל LinearLayout הראשי