

Mobil megoldások

Kocsis Gergely
2019.10.07.

SharedPreferences

Az android SDK natív módon lehetőséget ad a perzisztens adattárolásra a SharedPreferences fájl segítségével.

Ennek használata lehetővé teszi, az alkalmazás visszaállítását akár kilépés után is, viszont jóval erőforrásigényesebb, mint a savedInstanceState használata.

Töltsük le és alakítsuk át az előzőekben használt ShoppingList alkalmazást úgy, hogy a lista szövegét perzisztens módon tárolja.



SharedPreferences

Hozunk létre egy privát mezőt a SharedPreferences tárolására és egy String kulcsot.

```
private SharedPreferences sharedPref;  
private String sharedPrefFile = "shprf";
```

Az onCreate metódusban inicializáljuk sharedPref mezőt.

```
sharedPref = getSharedPreferences(sharedPrefFile, MODE_PRIVATE);
```



SharedPreferences

Hozzuk létre egy privát mezőt a sharedPreferences tárolására és egy String kulcsot.

```
private SharedPreferences sharedPref;  
private String sharedPrefFile = "shprf";
```

Az onCreate metódusban inicializáljuk sharedPref mezőt.

```
sharedPref = getSharedPreferences(sharedPrefFile, MODE_PRIVATE);
```

A mentés az onPause metódusban történik SharedPreferences.Editor segítségével:

```
SharedPreferences.Editor preferencesEditor = sharedPref.edit();  
preferencesEditor.putString("LIST", itemList.getText().toString());  
preferencesEditor.putBoolean("LIST_BOOL", isEmptyList);
```

```
preferencesEditor.apply();
```



SharedPreferences

Végül az onCreate metódusban tudjuk visszatölteni a sharedPreferences-ben tárolt adatokat.

```
itemsList.setText(sharedPref.getString("LIST", itemsList.getText().toString()));  
isEmpty = sharedPref.getBoolean("LIST_BOOL", isEmpty);
```

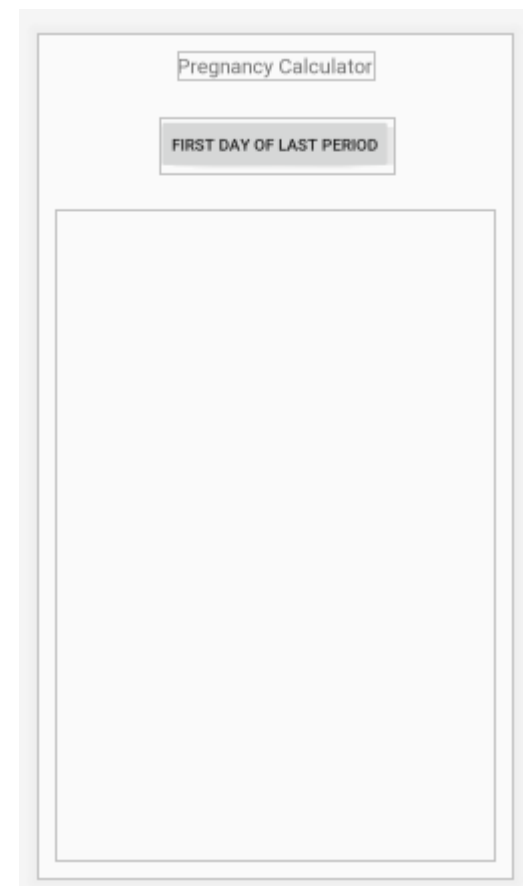
A második paraméter az alapértelmezett érték, mely akkor áll be, ha nem sikerül a perzisztens tárból betölteni az értéket.



Date picker

Készítsünk egyszerű terhességi kalkulátor alkalmazást, mely a megadott dátum alapján kiírja, hogy hány hetes és napos terhes az illető.

Az alkalmazás fő Activityje csak egy TextView-t és egy Button-t tartalmaz. A gombot megnyomva lehet megadni a dátumot, melyre a válasz egy újabb TextViewban jelenik meg.



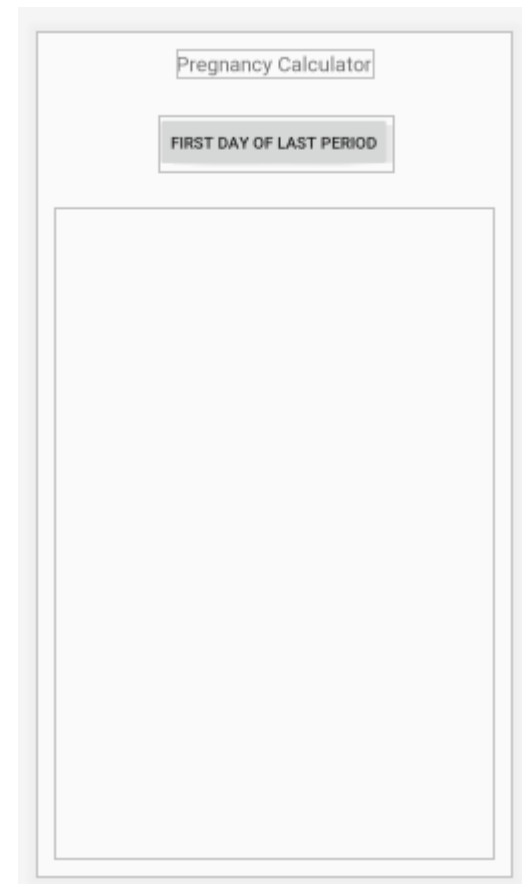
Date picker

Készítsünk egyszerű terhességi kalkulátor alkalmazást, mely a megadott dátum alapján kiírja, hogy hány hetes és napos terhes az illető.

Az alkalmazás fő Activityje csak egy TextView-t és egy Button-t tartalmaz. A gombot megnyomva lehet megadni a dátumot, melyre a válasz egy újabb TextViewban jelenik meg.

Ne felejtjük el generálni és beállítani a gomb megnyomásakor hívandó metódust!

```
public void showDatePicker(View view) {  
}
```



Date picker

Hozzuk létre egy új Fragment-et, mely a dátum választását implementálja.

A Fragment legyen a DialogFragment leszármazottja és implementálja a DatePickerDialog szükséges interfészét.

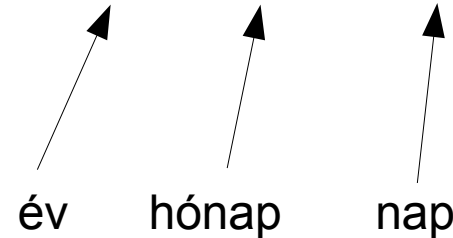
Létrehozáskor figyeljünk oda, hogy mindhárom checkbox-ot hagyjuk állítsuk kijelöletlenre!

```
import androidx.fragment.app.DialogFragment;
import android.widget.DatePicker;

public class DatePickerFragment extends DialogFragment
    implements DatePickerDialog.OnDateSetListener {

    @Override
    public void onDateSet(DatePicker datePicker, int i, int i1, int i2) {
    }
}
```

év hónap nap



Date picker

Az onCreate helyett ezúttal az onCreateDialog metódus definiáljuk felül.

```
@NonNull
@Override
public Dialog onCreateDialog(@Nullable Bundle savedInstanceState) {
    // Use the current date as the default date in the picker.
    final Calendar c = Calendar.getInstance();
    int year = c.get(Calendar.YEAR);
    int month = c.get(Calendar.MONTH);
    int day = c.get(Calendar.DAY_OF_MONTH);

    // Create a new instance of DatePickerDialog and return it.
    return new DatePickerDialog(getActivity(), this, year, month, day);
}
```

A metódus eredményeként iduló dátumválasztó dialógusablak a ami napra állítva indul. (A LocalDate osztály sajnos csak API Level 26-tól használható).



Date picker

Most állítsuk be, hogy a MainActivity-ben a gombnyomást kezelő metódus indítások jelenjen meg a dátumválasztó.

```
DialogFragment newFragment = new DatePickerFragment();  
newFragment.show(getSupportFragmentManager(), "datePicker");
```

A dátumválasztóban megadott dátum kezelésére adjunk meg egy új metódust a MainActivity-ben:

```
public void processDatePickerResult(int year, int month, int day) {  
}
```

Hívjuk meg ezt a metódust a dátumválasztó `onDataSet` metódusából.

```
MainActivity activity = (MainActivity) getActivity();  
activity.processDatePickerResult(i, i1, i2);
```



Date picker

A `processDatePickerResult` metódusban számoljuk ki, hogy hány hetes és hány napos terhes a felhasználó.

Sajnos továbbra is a `Calendar` osztállyal kell dolgoznunk... :(

```
public void processDatePickerResult(int year, int month, int day) {
    Calendar c = Calendar.getInstance();
    c.set(year, month, day);

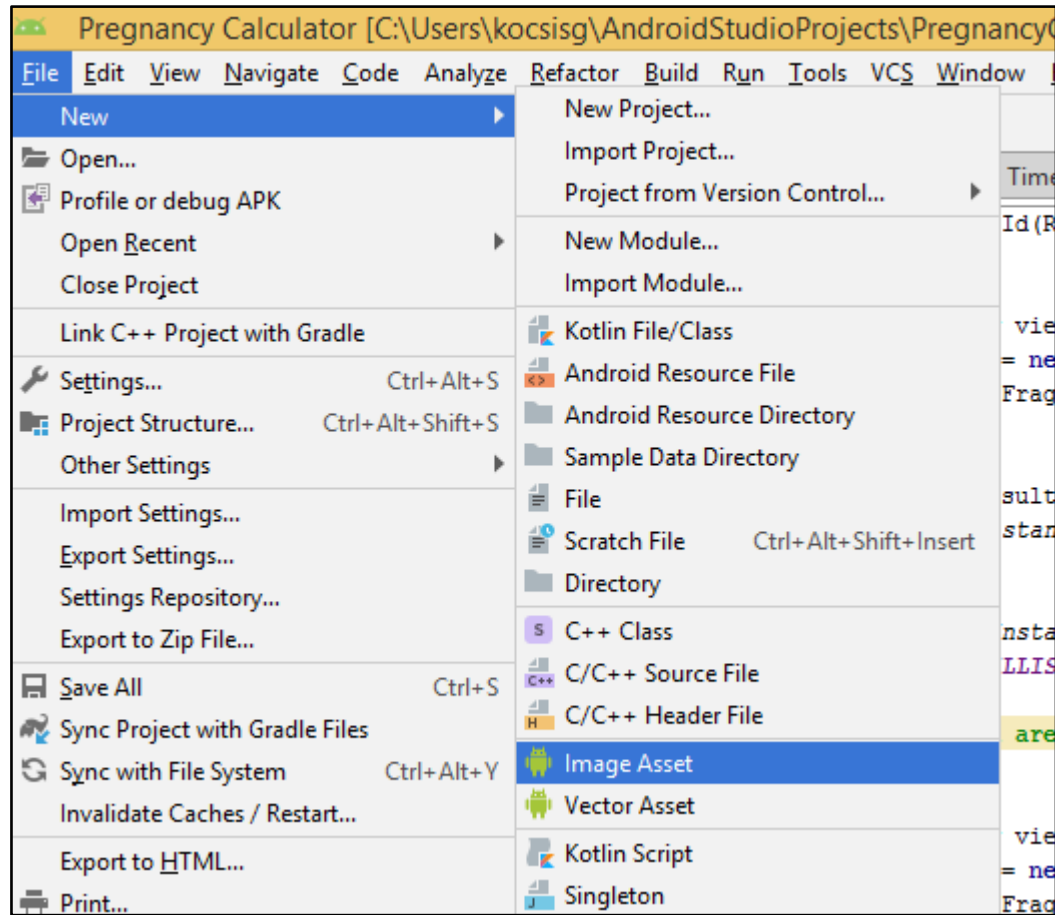
    long msDiff = Calendar.getInstance().getTimeInMillis() -
                  c.getTimeInMillis();
    long daysDiff = TimeUnit.MILLISECONDS.toDays(msDiff);

    answerTextView.setText("You are " + daysDiff/7 + " weeks\nand "
                           + daysDiff%7 + " days pregnant! ");
}
```



Date picker

Bónusz:



Önálló feladat – Time picker

Készíts egy olyan új alkalmazást, mely egyetlen gombot tartalmaz. A gomb megnyomása után egy időválasztó ablak jelenik meg. Az idő kiválasztása után a főablakon jelenjen meg egy Toast üzenetben a kiválasztott idő.

Segítség:

```
... implements TimePickerDialog.OnTimeSetListener
```

```
... Toast.makeText(this, "Time: " + i + ":" + i1, Toast.LENGTH_SHORT).show();
```

