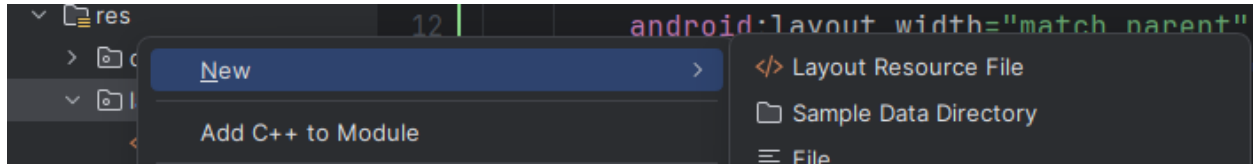
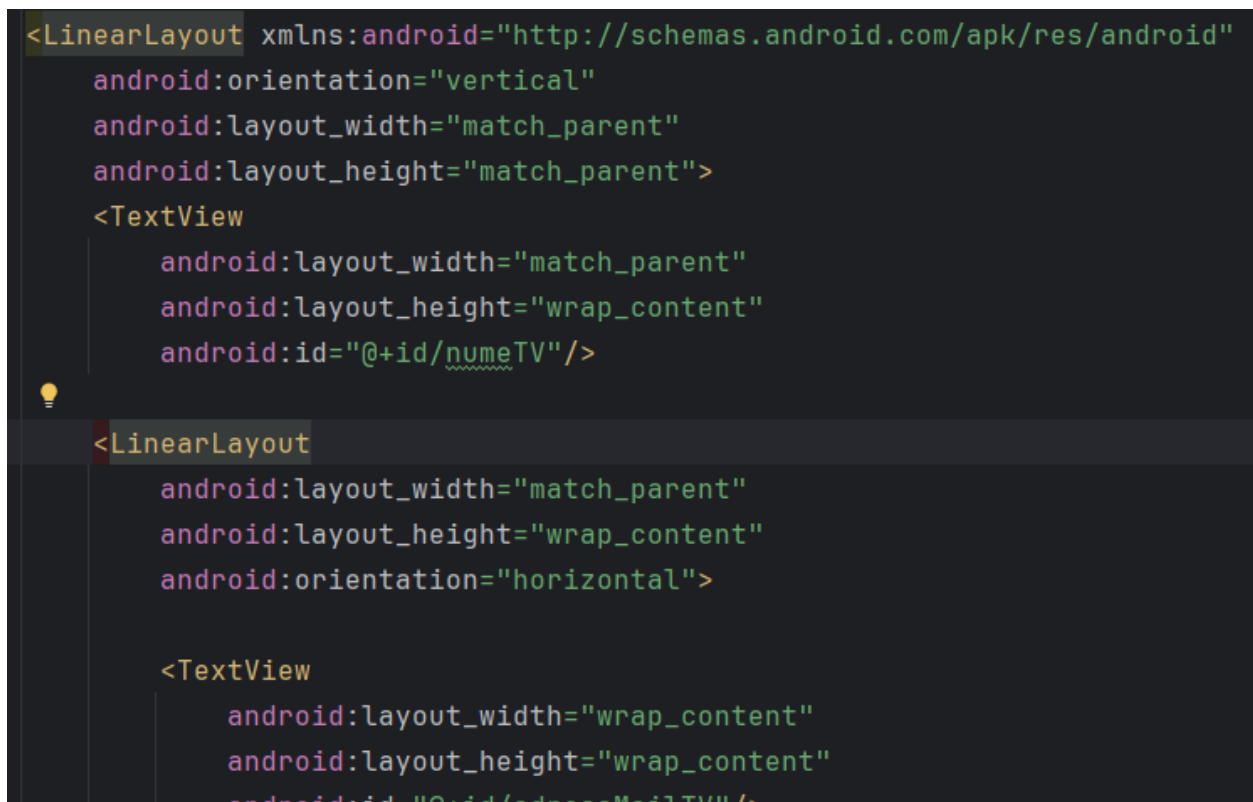


Laborator ANDROID - Custom Adapter pentru ListView

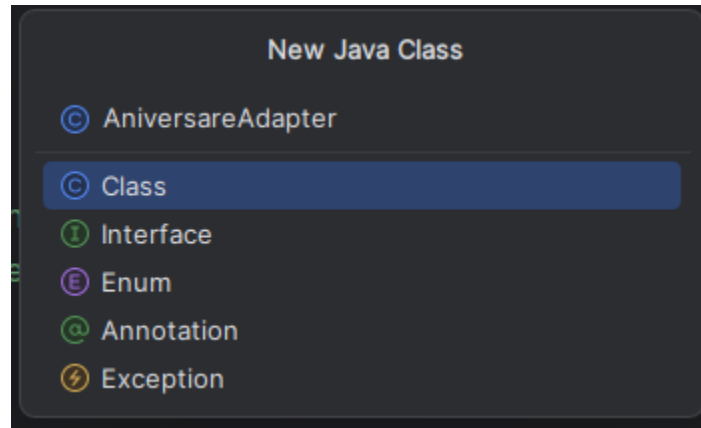
Trebuie adăugat în directorul de resurse Layout un fișier de tip "Layout Resource File".



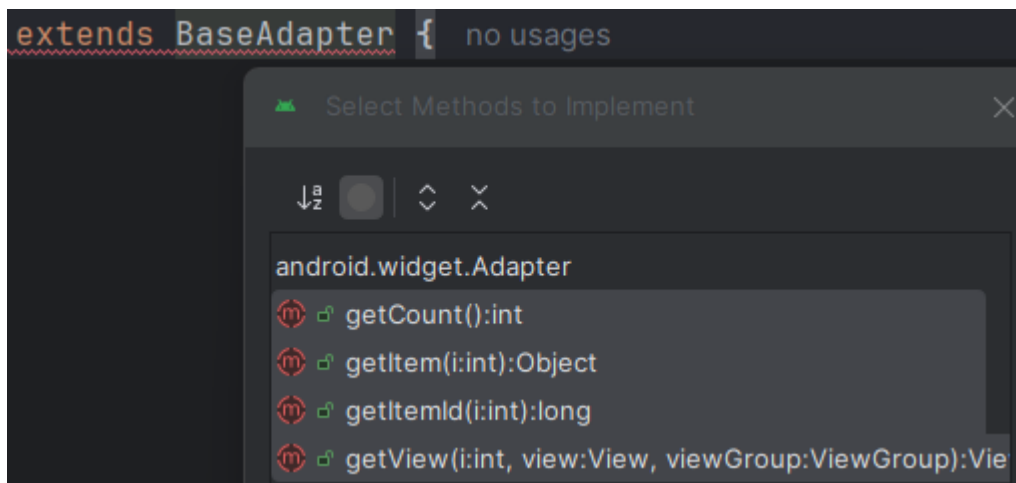
În acest fișier se creează layoutul pentru o linie din listView



Apoi este adăugată o clasă Java pentru adapterul personalizat:



Aceasta implementeaza interfața BaseAdapter și sunt adăugate metodele specifice.



În cadrul clasei sunt adăugate 3 atribute: o listă de obiecte pe care trebuie să o afișăm în ListView, contextul pe care îl vom folosi și un integer pentru layoutul pe care îl vom folosi pentru afișare. Pentru aceste atribute ne creăm un constructor cu 3 parametri.

```
private List<Aniversare> aniversari = null; 1 usage
private Context ctx; 1 usage
private int resursaLayout; 1 usage

public AniversareAdapter(List<Aniversare> aniversari, Context ctx, int resursaLayout)
{
    this.aniversari = aniversari;
    this.ctx = ctx;
    this.resursaLayout = resursaLayout;
}
```

Metoda getCount() trebuie sa returneze lungimea listei de obiecte.

```
@Override
public int getCount() {
    return aniversari.size();
}
```

Metoda getItem() trebuie să returneze elementul solicitat din cadrul listei.

```
@Override
public Object getItem(int i) {
    return aniversari.get(i);
}
```

Metoda getView este cea care trebuie sa faca maparea obiectului pe o linie din ListView. Aceasta metoda va fi apelata pentru fiecare obiect din lista noastra de obiecte (getCount() ori).

Pentru a realiza această mapare ne folosim de un Inflater pentru încărcă layoutul realizat mai devreme.

```
LayoutInflater inflater=LayoutInflater.from(ctx);
View v=inflater.inflate(resursaLayout,viewGroup, attachToRoot: false);
```

În cadrul acestei metode trebuie să identificăm fiecare view din layoutul nostru si apoi să precizăm ce informație trebuie să afișeze.

```

    TextView numeTV=v.findViewById(R.id.numeTV);
    TextView adresaTV=v.findViewById(R.id.adresaMailTV);
    TextView telefonTV=v.findViewById(R.id.nrTelefonTV);
    TextView dataTV=v.findViewById(R.id.dataTV);
    CheckBox notificaCB=v.findViewById(R.id.notificaCB);

    Aniversare aniversare=(Aniversare)getItem(i);

    numeTV.setText(aniversare.getNum());
    adresaTV.setText(aniversare.getAdresaMail());
    telefonTV.setText(aniversare.getNrTelefon());
    dataTV.setText(aniversare.getData().toString());
    notificaCB.setChecked(aniversare.isNotifica());

    return v;

```

Acum in loc sa folosim acel ArrayAdapter vom folosi acest nou adapter creat. Putem sa il declaram ca atribut al clasei, astfel incat sa il putem folosi peste tot in cadrul clasei.

```

> public class ListaAniversariActivity extends AppCompatActivity {
    private List<Aniversare> aniversari=null; 6 usages
    private int idModificat=0; 2 usages
    private AniversareAdapter adapter=null; 4 usages

```

```

// ArrayAdapter<Aniversare> adapter=new ArrayAdapter<>(getApplicationContext(), android.R.layout.simple_list_item_1,aniversari);
// lv.setAdapter(adapter);
adapter=new AniversareAdapter(aniversari,getApplicationContext(),R.layout.row_item);
lv.setAdapter(adapter);

```

Acum trebuie sa modificam functionalitatea de la ItemClick, astfel incat sa deschidem activitatea de modificare/adaugare. Acestea ii vom da ca extra obiectul selectat din listview.

```

lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        Intent intentModifica=new Intent(getApplicationContext(),AdaugareActivity.class);
        intentModifica.putExtra( name: "aniversare",aniversari.get(i));
        idModificat=i;
        startActivityForResult(intentModifica, requestCode: 209);
        Toast.makeText(getApplicationContext(), aniversari.get(i).toString(), Toast.LENGTH_SHORT).show();
    }
});

```

In activitatea de adaugare pe onCreate trebuie sa verificam daca avem extra un obiect. In cazul in care avem, trebuie sa preluam acel obiect si cu ajutorul lui sa populam toate view-urile din activitate, astfel incat utilizatorul sa modifice ce doreste.

```

Intent it=getIntent();
if(it.hasExtra( name: "aniversare")){
    Aniversare aniversare=it.getParcelableExtra( name: "aniversare");
    EditText numeEt=findViewById(R.id.numeET);
    EditText mailEt=findViewById(R.id.emailET);
    EditText nrTelefonEt=findViewById(R.id.numarTelefon);
    DatePicker datePicker=findViewById(R.id.dataAniversare);
    CheckBox notificaCB=findViewById(R.id.notificaCB);

    numeEt.setText(aniversare.getNume());
    mailEt.setText(aniversare.getAdresaMail());
    nrTelefonEt.setText(aniversare.getNrTelefon());
    datePicker.init(aniversare.getData().getYear(),aniversare.getData().getMonth(),aniversare.getData().getDayOfMonth(),null);
    notificaCB.setChecked(aniversare.isNotifica());
}

```

La intoarcere, trebuie sa il modificam si in cadrul listei si sa notificam adapterul ca au fost realizate modificari in sursa de date

```

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode==RESULT_OK){
        if(requestCode==209){
            aniversari.set(idModificat,data.getParcelableExtra( name: "aniversare"));
            adapter.notifyDataSetChanged();
        }
    }
}

```