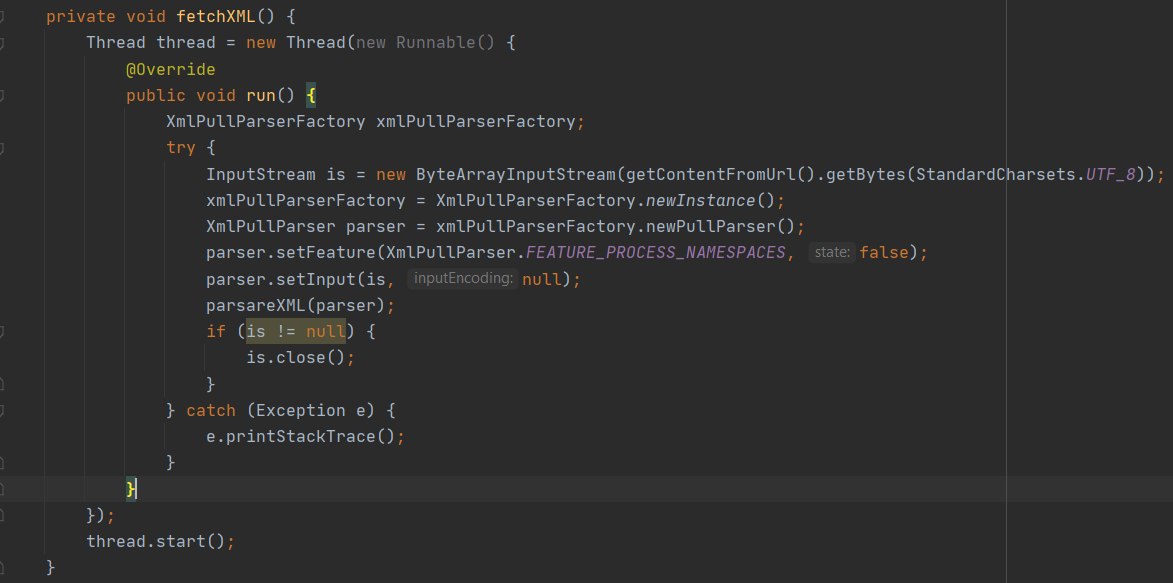
***Document tehnic***

**~Curs BNR - Inventar~**

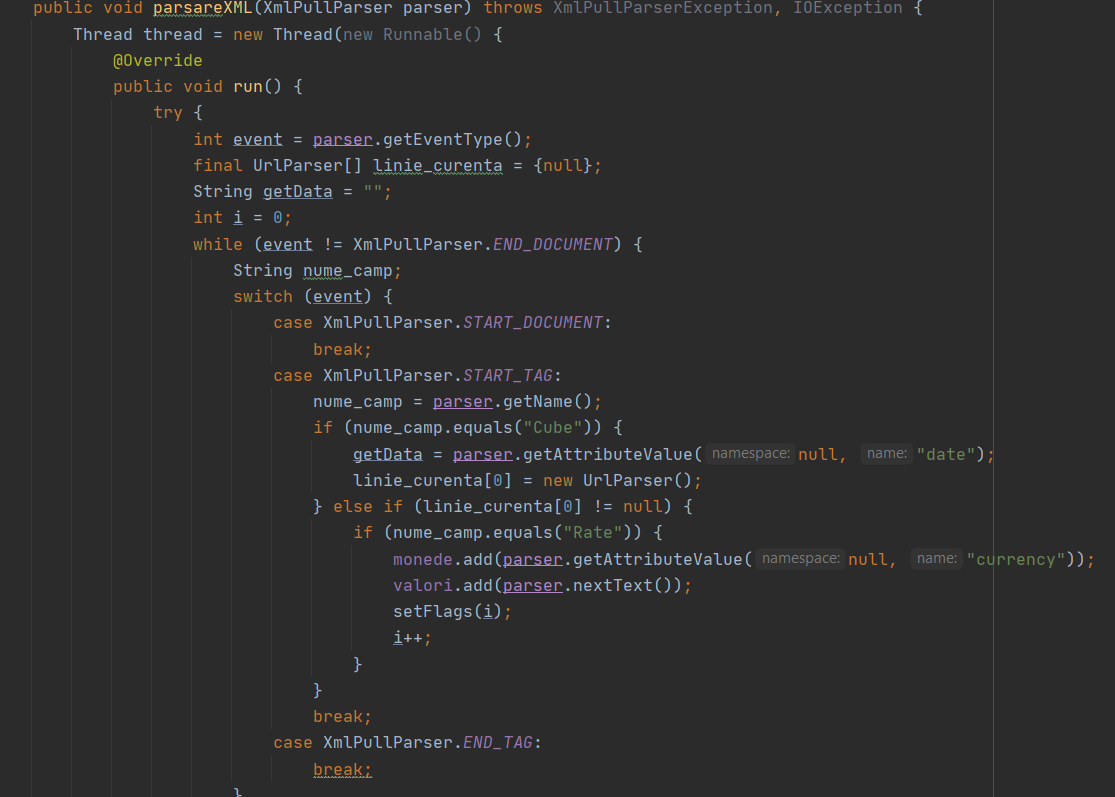
* getContentFromURL(){} -> functie care preia continutul unui fisier de tip XML din URL-ul atribuit si il transmite ca si input catre parser(URL-ul specific ultimei actualizari a cursului).



* fetchXML(){} -> functie care creeaza o noua instanta de parser, caruia ii atribui input-ul dorit si care apeleaza functia de parsare necesara.

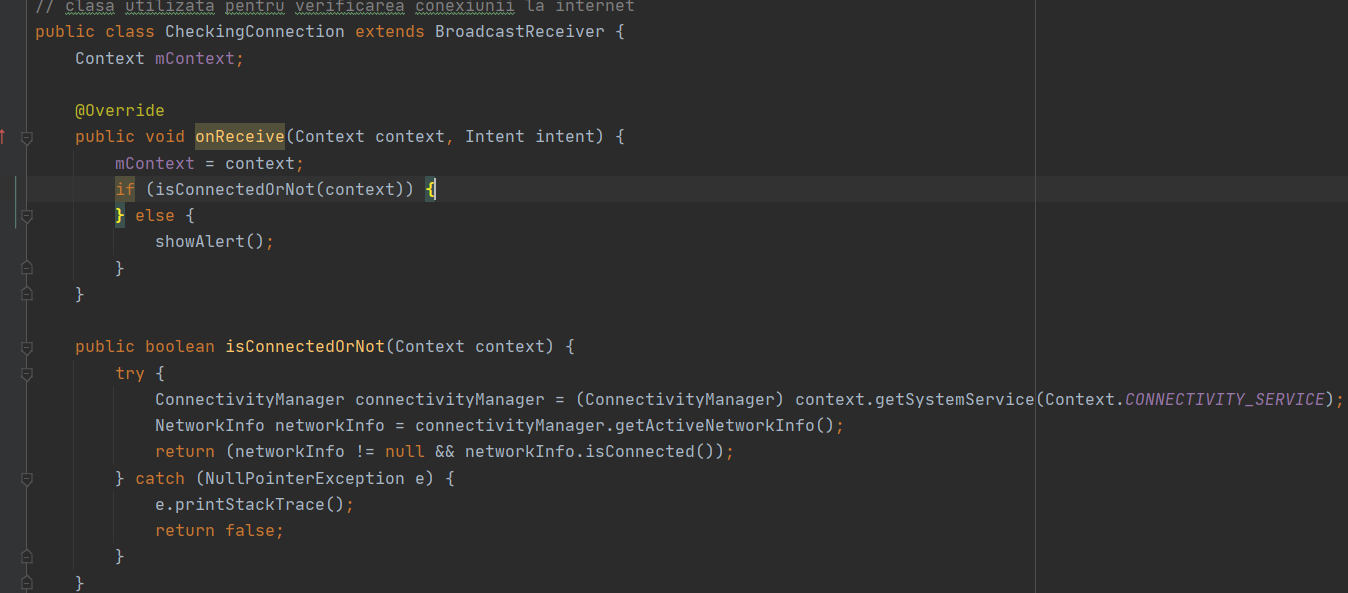


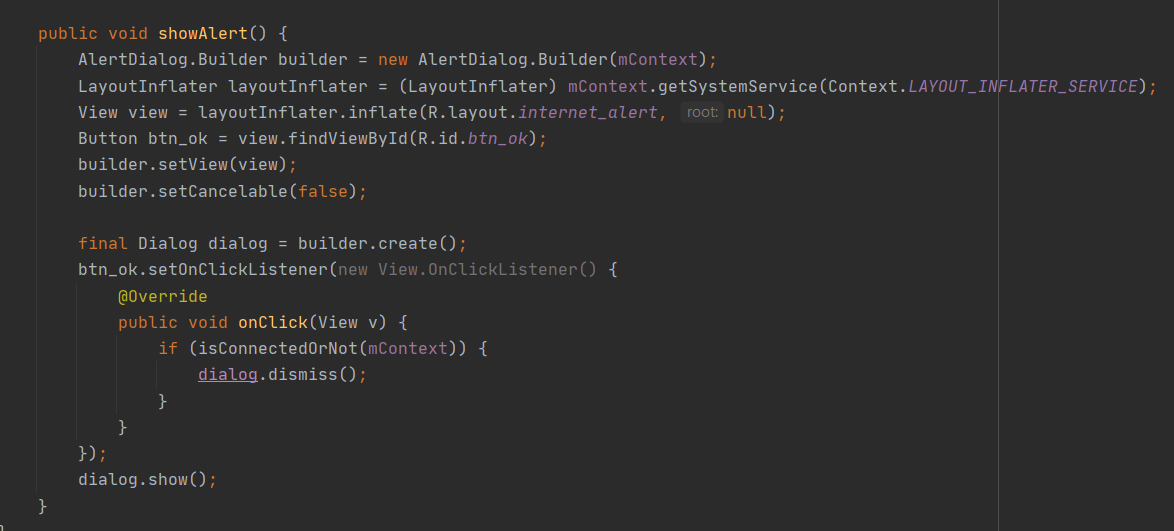
* parsareXML(){} -> functie care realizeaza parsarea propriu-zisa si adaugarea datelor necesare (monele si valoarea corespunzatoare lor) in lista ce urmeaza sa fie afisata.



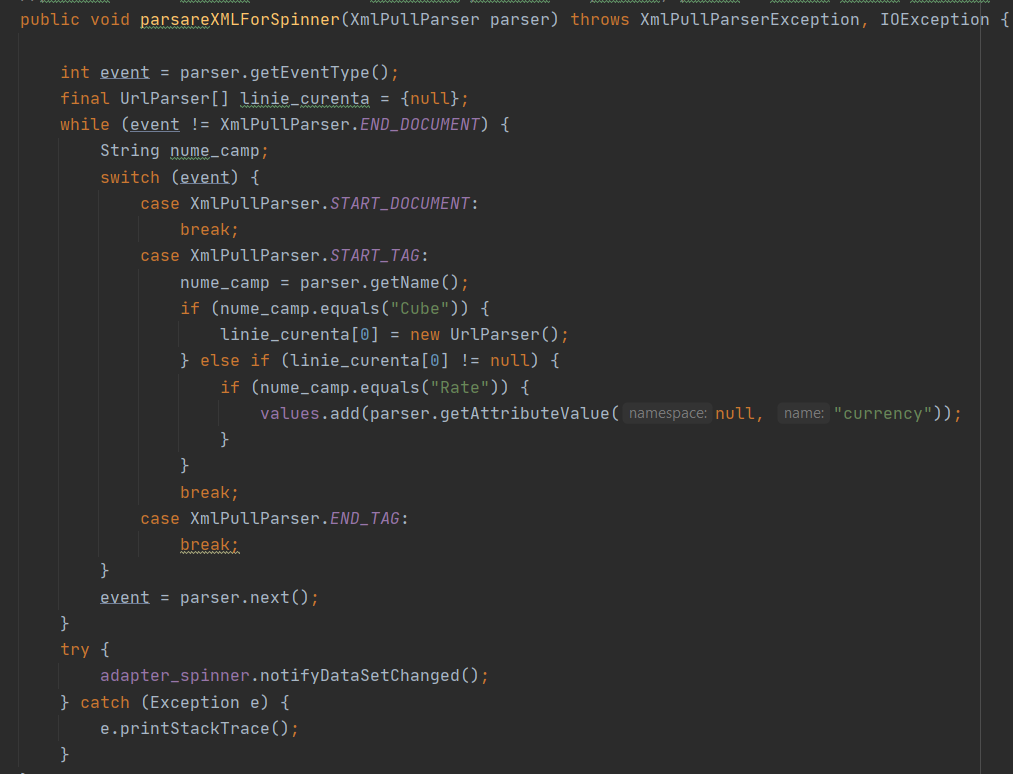


* CheckingConnection -> clasa utilizata pentru verificarea conexiunii dispozitivului la internet.

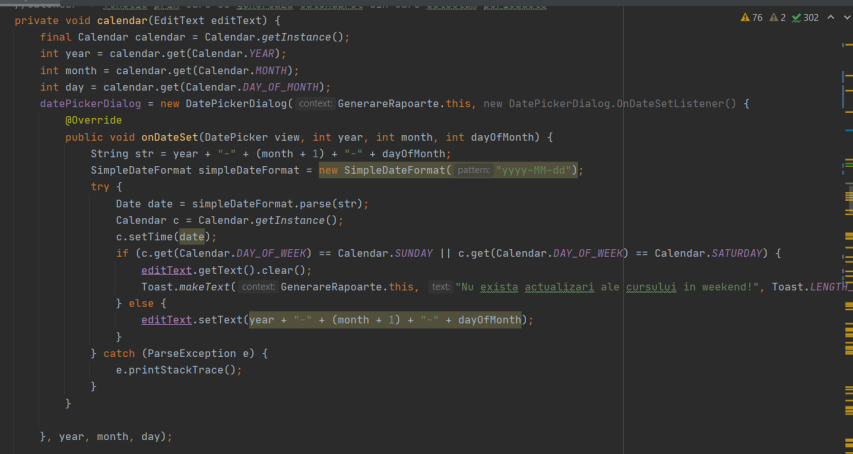


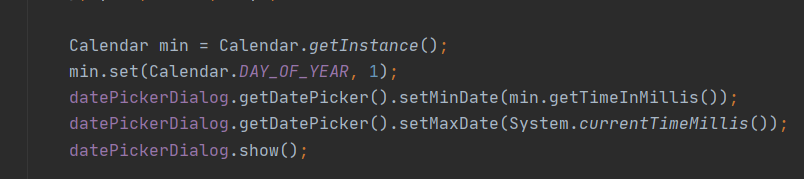


* parsareXMLForSpinner(){} -> functie ce realizeaza parsarea propriu-zisa a fisierului XML , preia din acesta doar numele monedelor si le adaugata in lista ce reprezinta elementele spinner-ului.

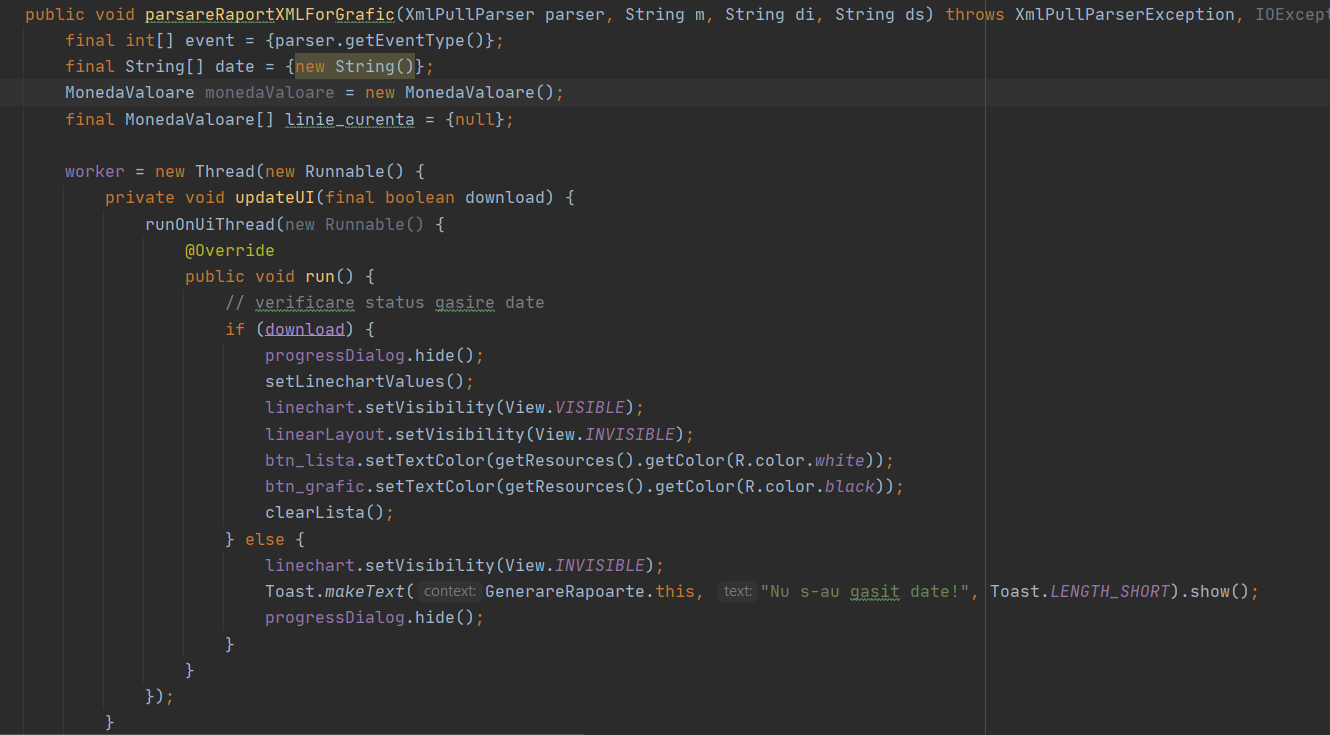


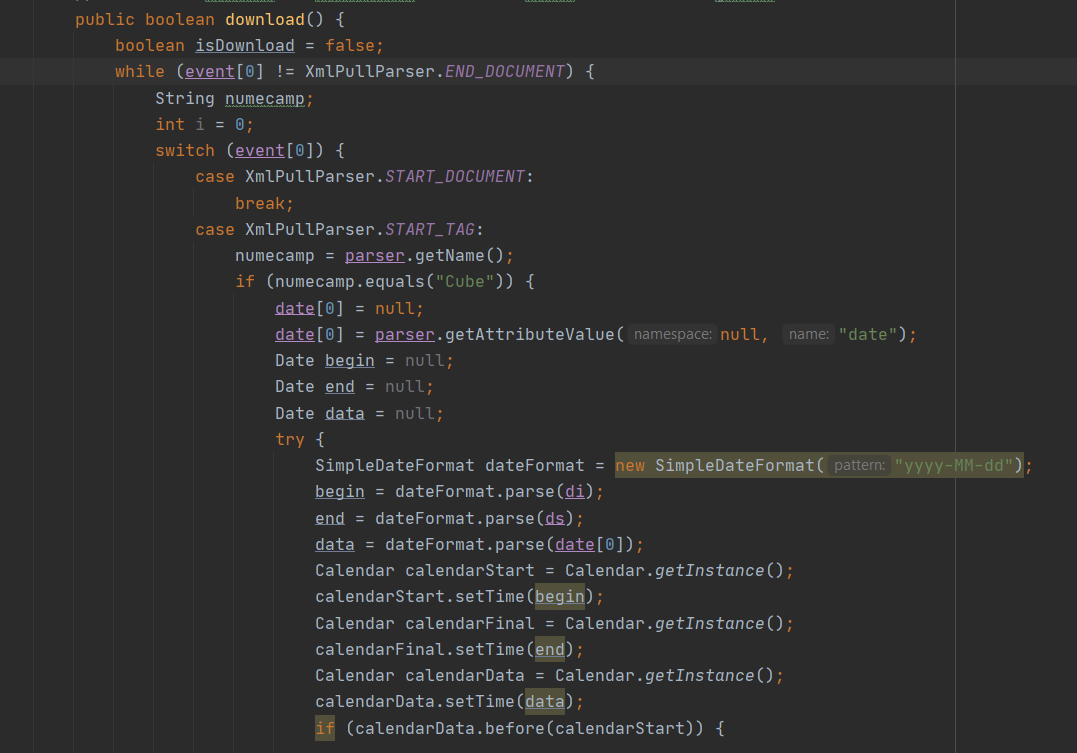
* calendar(){} -> functie care genereaza calendarul din care se aleg data de inceput si cea de sfarist.

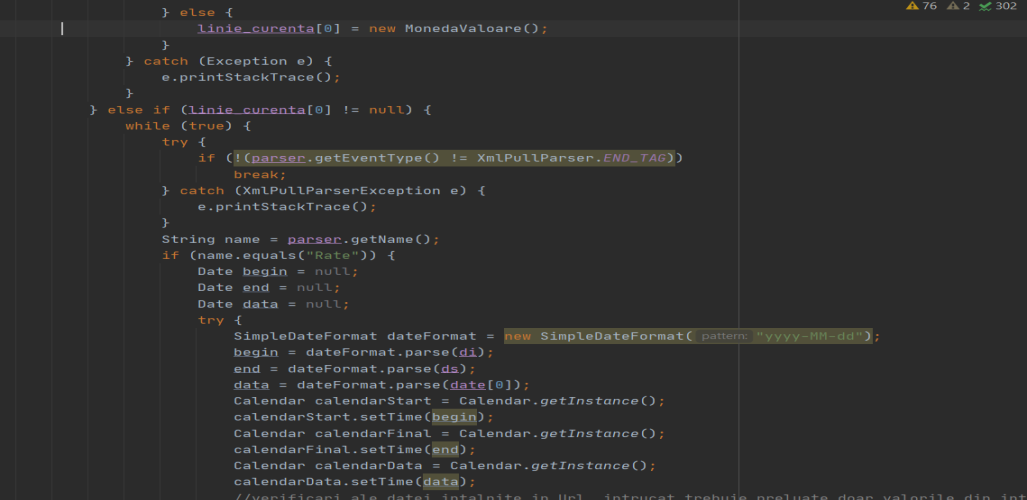


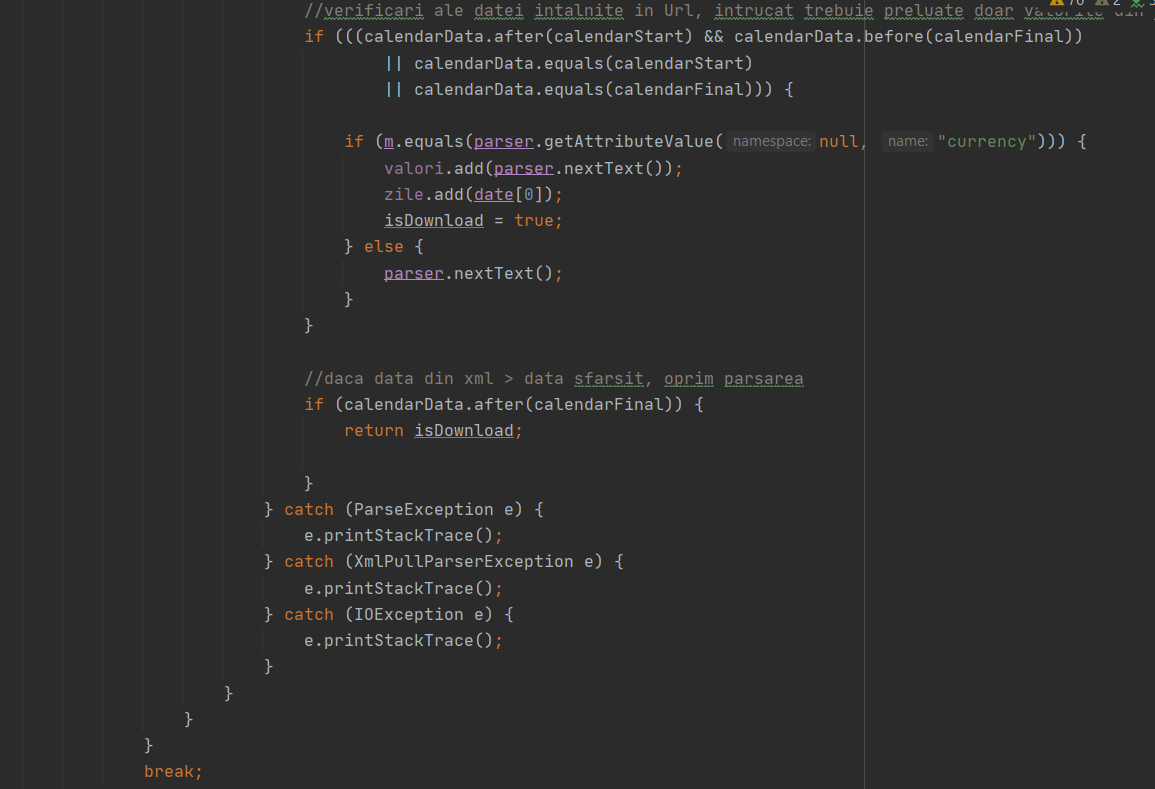


* parsareRaportXMLForGrafic(){} -> functie ce realizeaza parsarea propriu-zisa a fisierului XML, preia din acesta in functie de moneda selectata in spinner, toate valorile acesteia din intervalul ales. Valorile sunt atribuite unui vector care reprezinta valorile de pe grafic.



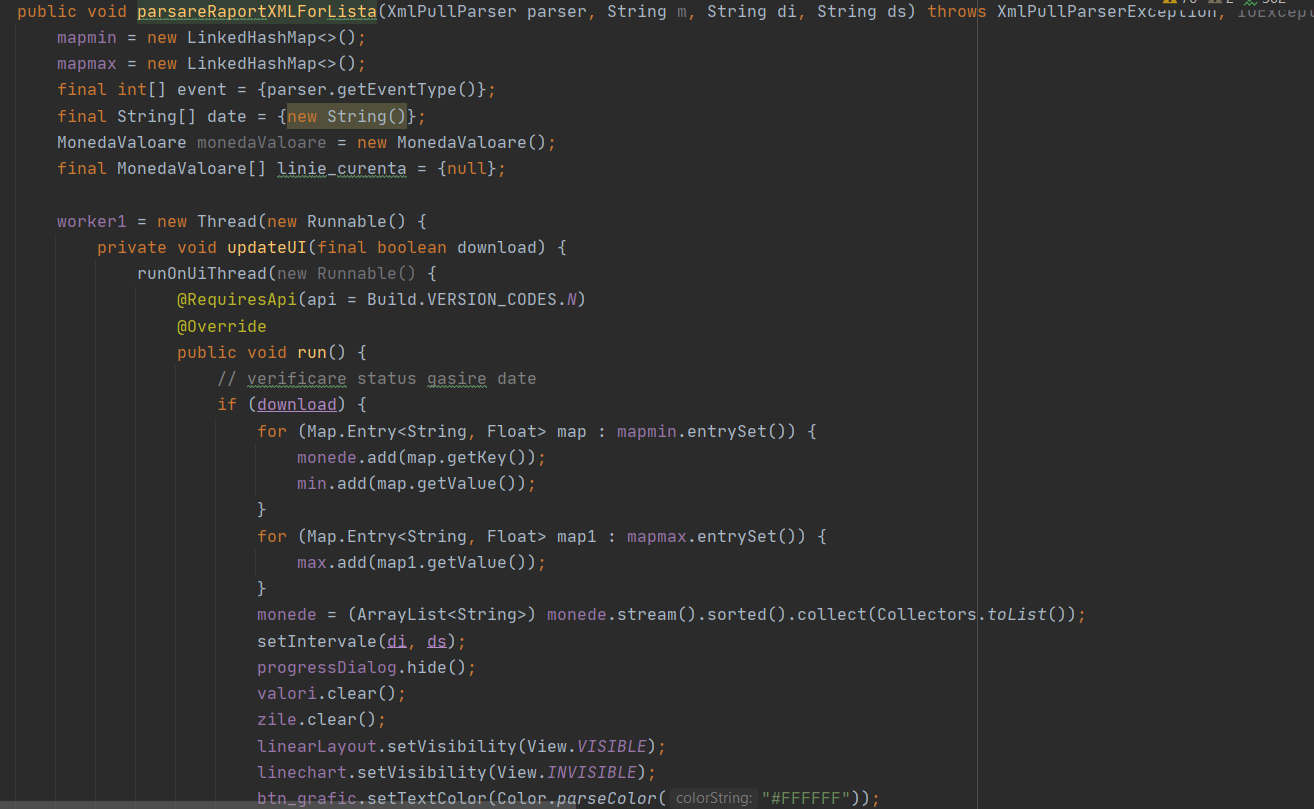


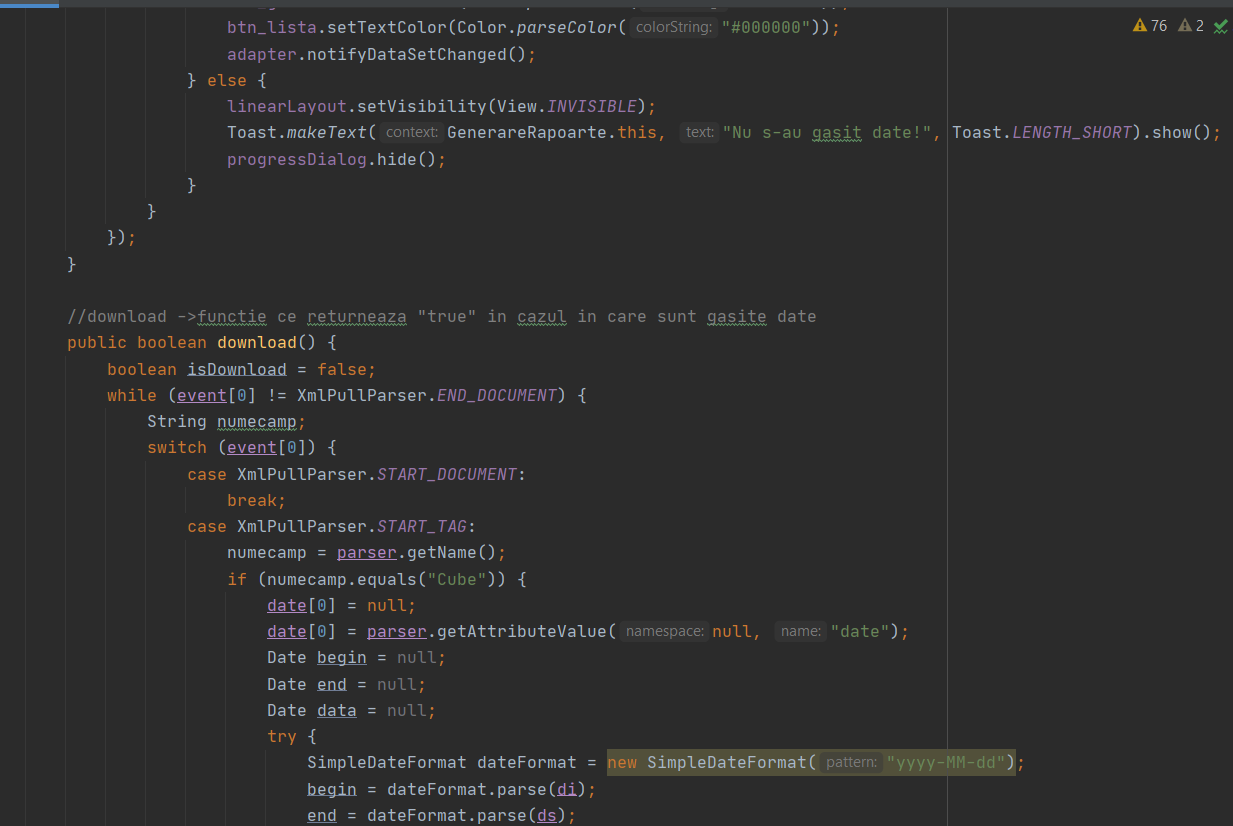


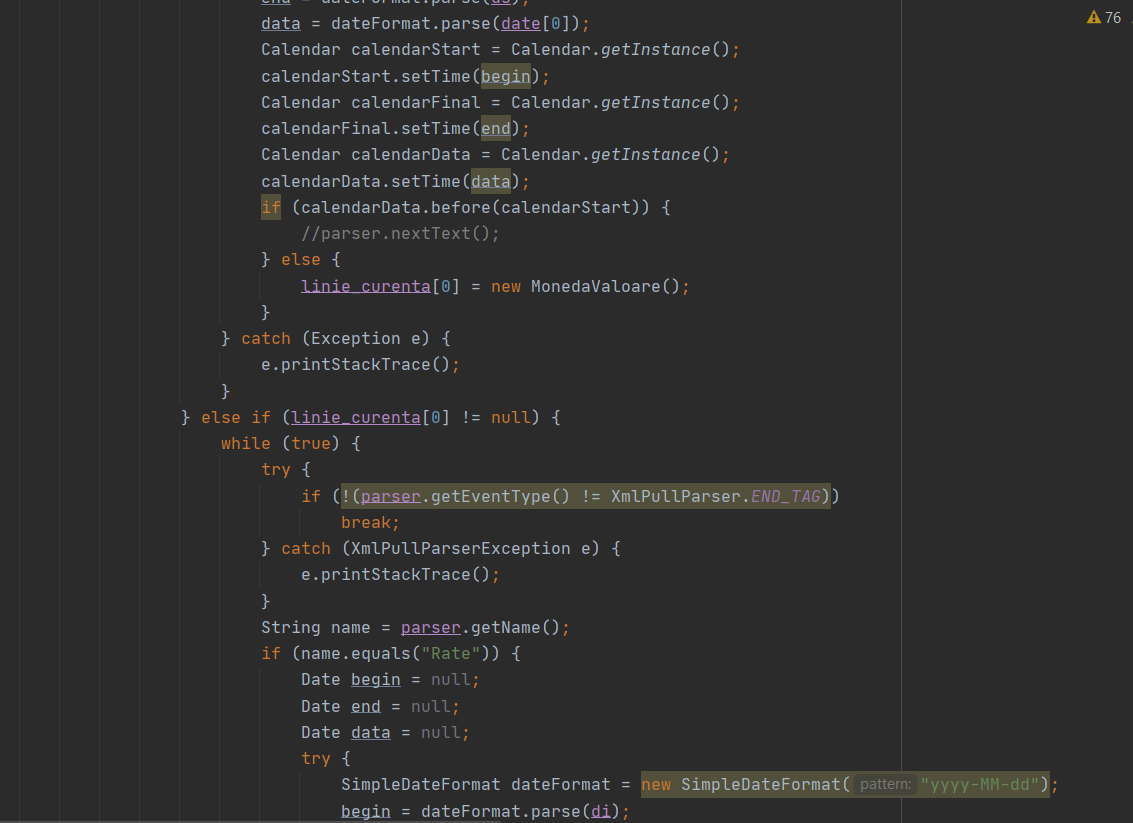


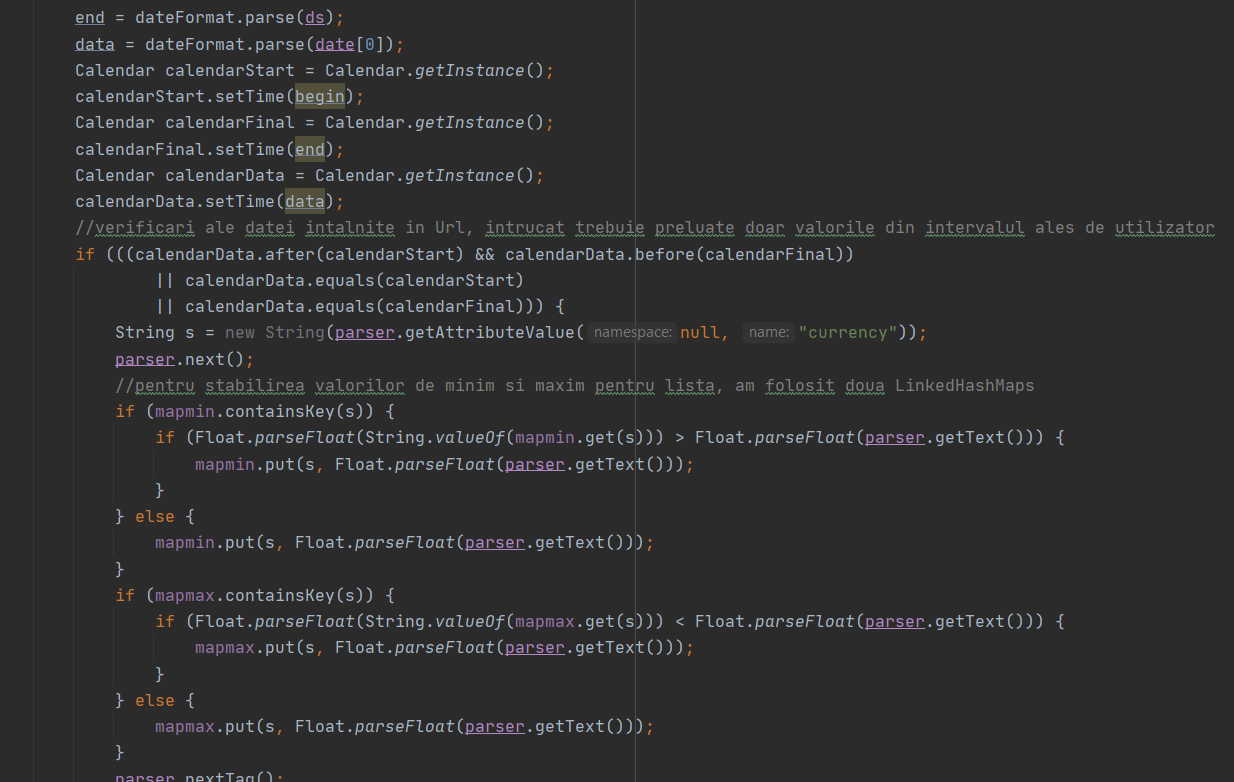


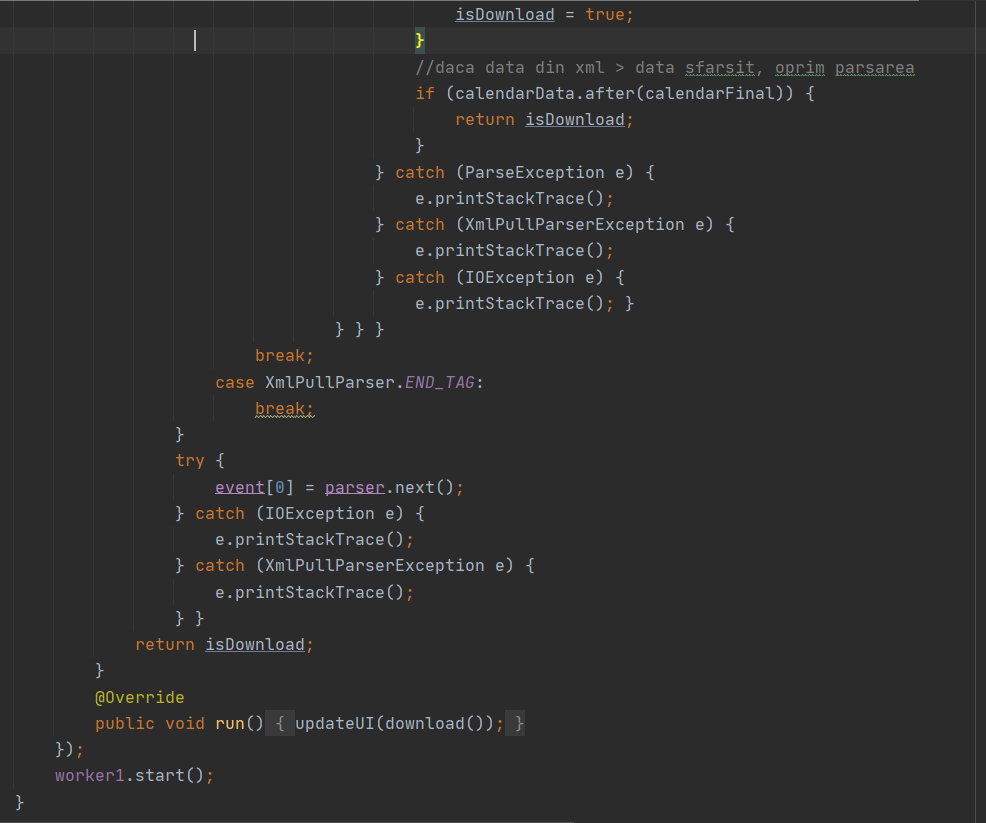
* parsareRaportXMLForLista(){} -> functie ce realizeaza parsarea propriu-zisa a fisierului XML, preia din acesta in functie de intervalul ales, valorile minime si maxime pentru fiecare moneda, valorile minime si maxime fiind decise cu ajutorul a doua LinkedHashMap-uri in care se stocheaza valorile respective.



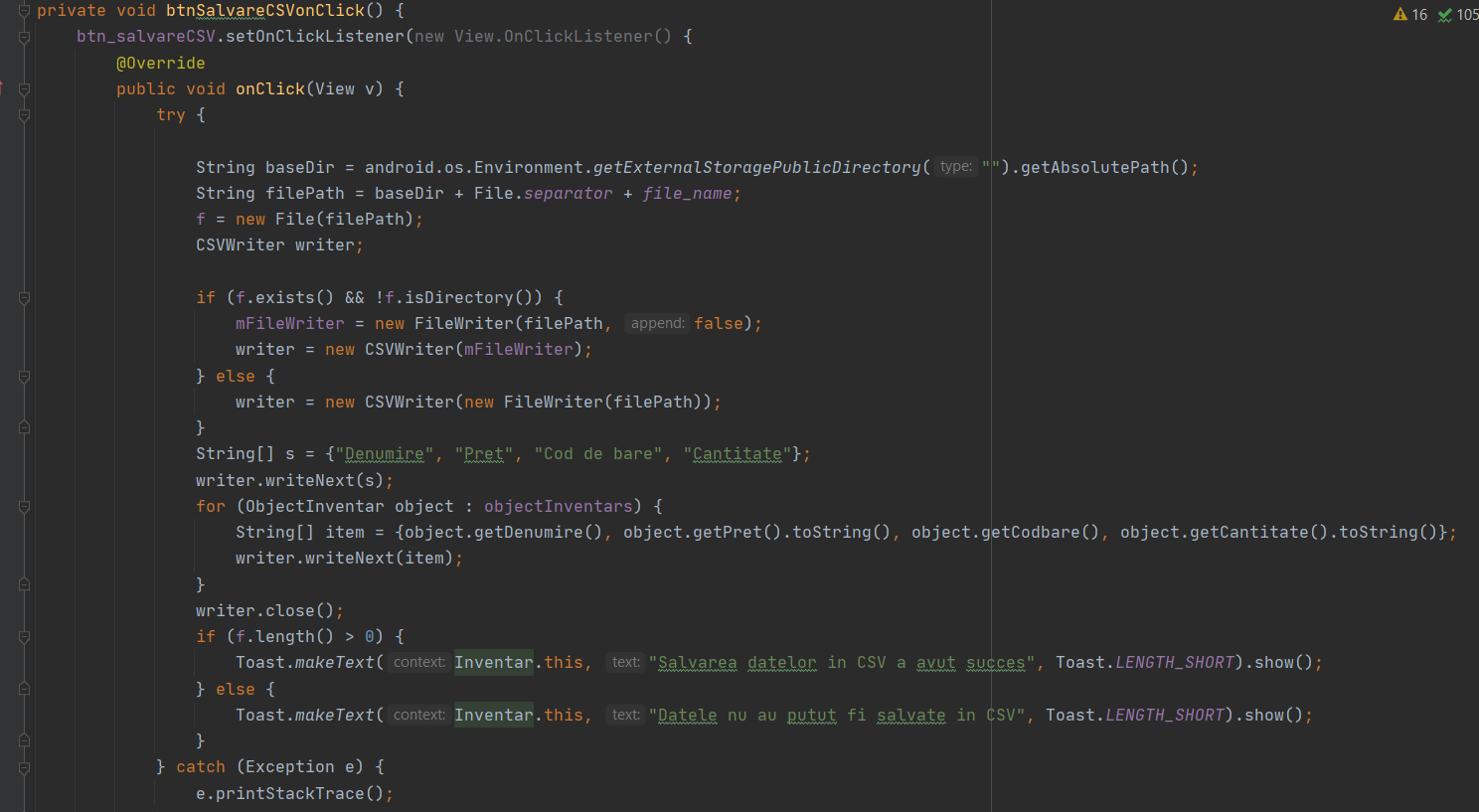




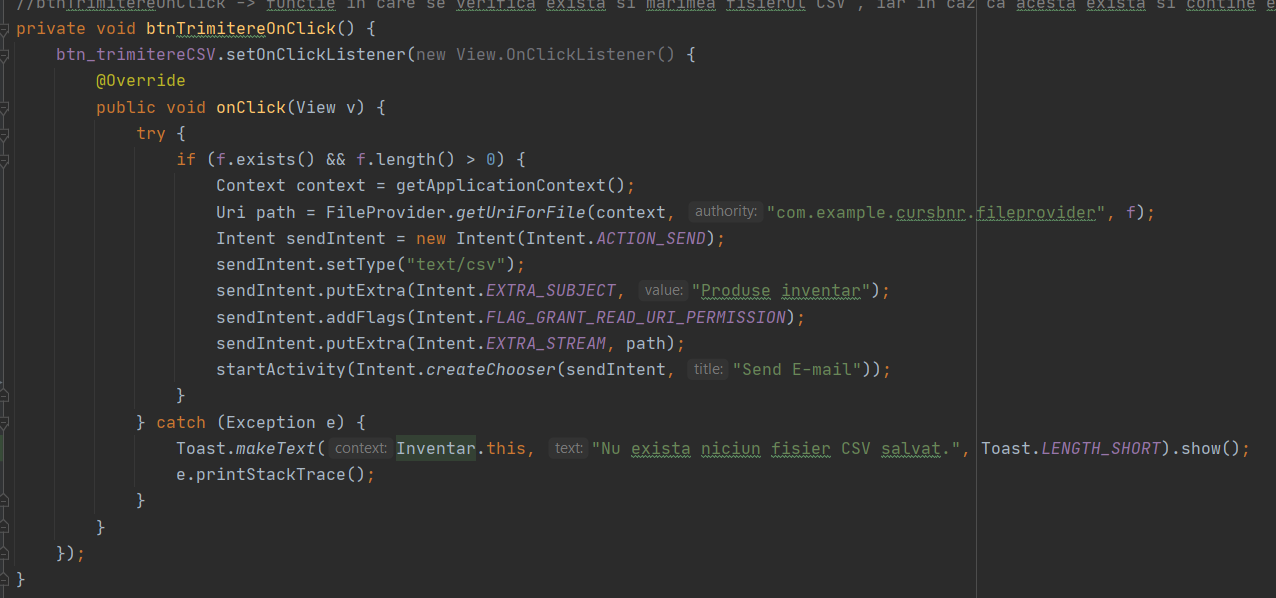




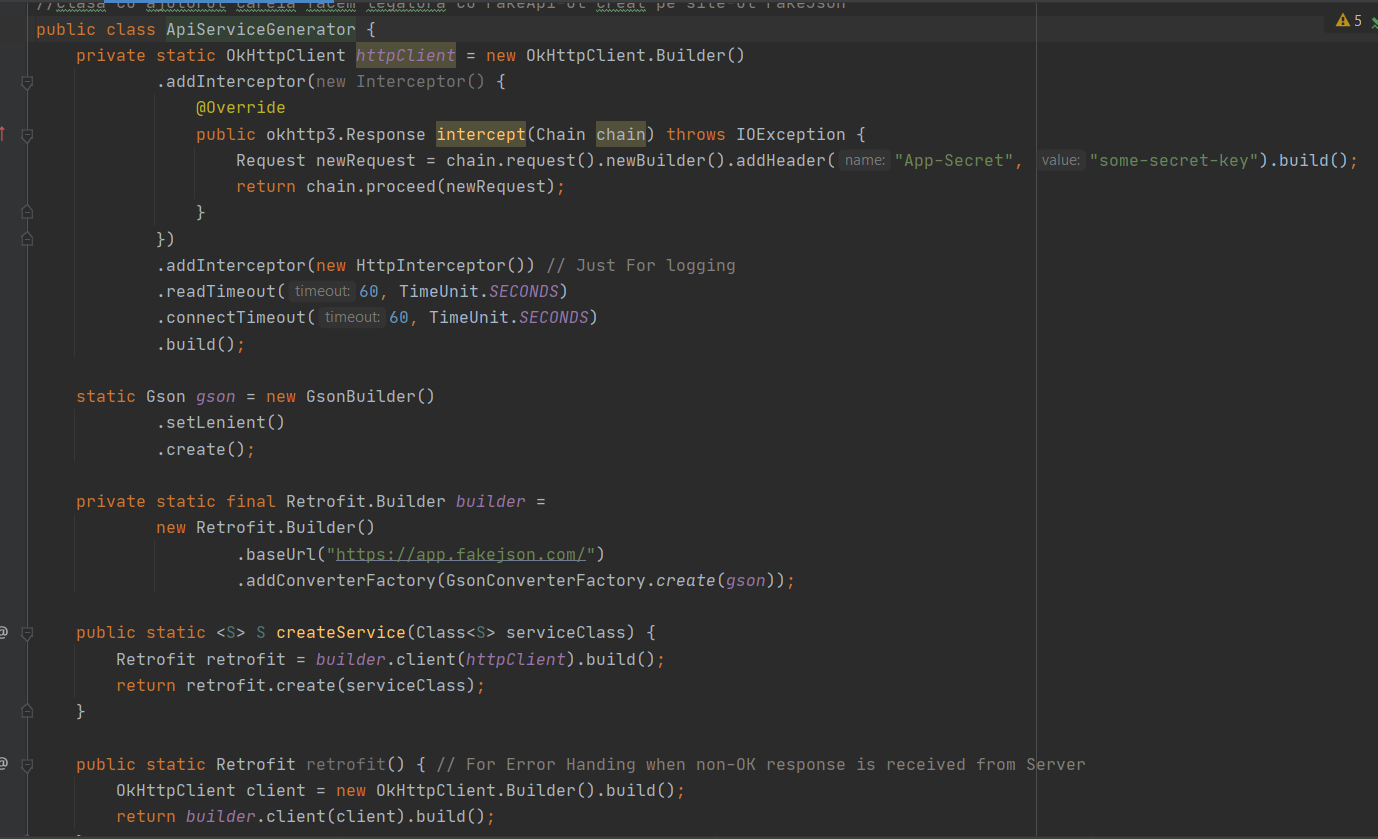
* btnSalvareCSVonClick(){} ->functie pe onClick care salveaza toate datele din lista din Inventar intr-un fisier de tip CSV , in memoria telefonului.



* btnTrimitereOnClick(){} -> functie pe onClick care verifica intai existenta fisierului de tip CSV, apoi daca acesta exista si contine date , afiseaza un popup cu variante prin care se poate trimite fisierul.



* ApiServiceGenerator() -> clasa folosita pentru generarea unui retrofit catre URL-ul ce contine fisierul Json care cuprinde datele necesare popularii listei din Inventar.



* retrofit() -> functie care verifica intai daca baza de date contine date din fisierul Json, daca nu, face apel catre acesta cu ajutorul clasei “ApiServiceGenerator”,clasei de tip obiect “FakeApiResponse” si a interfetei “JsonFakeApi”( care contine comanda de GET din fisier), iar daca apelul este realizat cu succes , raspunsul acestuia este inserat in baza de date de unde va fi adaugat in lista afisata in activitatea Inventar.



