Cont bancar

O familie formata din **n** membri gestioneaza in comun datele unui cont cu 2 valute (RON si EUR). Fiecare membru al familiei poate depune si retrage orice suma, in orice valuta, tranzac tiile pe valute diferite putandu-se face in paralel. Pentru retragere e necesar sa existe bani in cont. Toate tranzactiile, precum si soldurile curente sunt adaugate intr-o lista (Nod: index-user, tip\_valuta, tip\_tranzactie, suma\_tranzactionata, sold current).

Scrieti un program concurent care simuleaza comportamentul descris mai sus. Un alt thread itereaza lista de fiecare data dupa ce au avut loc 5 tranzactii (depuneri sau/si retrageri) si afiseaza pe ecran datele citite din lista.

Indicatii:

Membrii familiei sunt modelati de obiecte ale unei clase de tip Thread.

Contul bancar este reprezentat de un obiect “partajat” de cele **n** thread-uri (modelarea contului se face folosind o clasa cu metodele DepuneLei(), RetrageLei(), DepuneEur(), RetrageEur()).

**Test:**

* **n** = 3;
* Thread-ul Main porneste cele **n** thread-uri corespunzatoare membrilor si un alt thread care itereaza si afiseaza lista de tranzactii.
* Fiecare thread, corepunzator unui membru din familie genereaza aleator 20 tranzactii de tip: [*index-user (ID thread), tip\_valuta (RON sau EUR), tip\_tranzactie (depunere sau retragere), suma\_tranzactionata (intre 1 si 1000*)], la un interval de 10ms si adauga fiecare tranzactie in lista de tranzactii daca aceasta s-a putut efctua; daca o tranzactie nu se poate efectua (retragere fara sa existe destui bani) se afiseaza pe ecran un mesaj de eroare corespunzator.
* Thread-ul care itereaza, realizeaza afisarea la fiecare 5 tranzactii efectuate. In timp de asteapta efecturarea a urmatoarelor 5 tranzactii se afla in starea “wait”. Acest thread isi incheie executia dupa ce toate cele **n** thread-uri si-au incheiat executia si dupa ce a afisat soldurile finale ale celor doua valute.