E-Banking

# Descriere proiect

Sa se realizeze o aplicație Java de gestiune a tranzacțiilor in sistemele de e-banking, care presupune lucrul cu fișiere (text/binare), vectori si colecții in Java. Se va asigura persistenta datelor, utilizând fișiere (si/sau baze de date), iar fișierele cu date inițiale vor fi actualizate in urma efectuării de prelucrări in cadrul aplicației.

# Ierarhie clase

Clasele au fost organizate in patru pachete: enums, exceptions, model, service.

1. **enums**

– conține obiectele tip Enum

* 1. **AccountType**

🡪 defineşte tipurile de conturi care pot fi create (SAVINGS | SALARY | FIXED\_DEPOSIT)

* 1. **Occupation**

🡪 defineste optiunile pentru selectarea ocupatiei utilizatorilor (STUDENT | EMPLOYEE | RETIRED)

1. **exceptions** – conține clasa cu exceptii user-defined
   1. **CustomExceptions**

🡪 clasa folosita pentru a genera exceptii definite, precum AccountNotFoundException, InsufficientFundsException, InvalidInput

1. **model** – defineste principalele entitati ale aplicatiei
   1. **User**

🡪 defineste utilizatorul

🡪 campuri: ID, firstName, lastName, occupation, age

🡪 contine validari pentru ID (obligatoriu 13 caractere, iar primul sa fie 1, 2, 5 sau 6) si varsta (min. 16 ani)

* 1. **Transaction**

🡪 defineste tranzactia

🡪 campuri:

tranzactionNo – numarul tranzactiei incrementat automat la initializarea unei tranzactii

type – obiect de tip enum, care reprezinta tipul tranzactiei,

amount – suma tranzactiei,

date – data la care a fost efectuata tranzactia

* 1. **Account**

🡪 defineste contul de utilizator

🡪 campuri:

user – obiect de tip user, reprezentand beneficiarul contului,

accountNo – numarul contului generat automat la crearea unui cont,

accountType – obiect de tip enum, care reprezinta tipul contului,

balance – sold disponibil,

transactions – lista de tranzactii efectuate

🡪 functii:

deposit – functie care defineste depozitul bancar. Primeste ca parametru suma de bani care urmeaza sa fie depozitata, verifica ca aceasta sa fie mai mare de 0 si efectueaza modificarile variabilei amount;

withdrawal – utilizata pentru extragerea de numerar. Verifica soldul disponibil si efectueaza tranzactia doar daca soldul disponibil este mai mare sau egal cu valoarea primita ca parametru. Altfel arunca o exceptie (InsufficientFundsException);

transfer – folosita pentru transferul bancar. Primeste ca parametru un obiect de tip Account care reprezinta destinatarul si suma care se doreste a fi transferata. Se verifica existenta destinatarului si daca expeditorul are suficient sold in cont pentru a transfera suma dorita. Altfel, se arunca exceptie;

viewAccountDetails – afiseaza detaliile contului si istoricul tranzactiilor.

1. **service** – contine clasele care furnizeaza functionalitatile aplicatiei
   1. **Bank**

🡪 furnizeaza functionalitatile de baza ale aplicatiei

🡪 campuri:

name – numele bancii,

List<Accounts> accounts – lista conturilor deschise la banca

🡪 functii:

getAccountByNumber – returneaza un cont in functie de numarul contului

createAccount – creeaza contul si il adauga in fisier

deleteAccount – sterge un cont existent si actualizeaza datele din fisier

saveDataToFile – functie de salvare a datelor in fisier

loadDataFromFile – functie de afisare a datelor in fisier

* 1. **Statistics**

🡪 clasa folosita pentru generarea statisticilor unei banci

🡪 contine o lista de conturi

🡪 functii:

calculateAverageBalance – returneaza soldul mediu al tuturor conturilor deschise

accountsByType – returneaza numarul conturilor existente de un anumit tip primit ca parametru

countUsersByOccupation – returneaza numarul utilizatorilor in functie de ocupatie (primita ca parametru)

countUsersByAgeRange - returneaza numarul utilizatorilor care au varsta cuprinsa intr-un anumit interval

1. **view** – contine clasele responsabile de interactiunea cu utilizatorul
   1. **Menu**

🡪 principala pagina care interactioneaza cu utilizatorul

🡪 contine un obiect de tip Bank

🡪 functii:

displayMenu – este functia principala care ofera utilizatorului optiunile disponibile ce pot fi efectuate in cadrul aplicatiei (Create account, View Account Details, Deposit, Withdraw, Transfer, Delete account, Statistics, Exit)

displayStatisticsMenu – afiseaza meniul cu optiunile disponibile pentru statistici

backMenu – functia care trimite utilizatorul spre pagina principala

backStatistics – functia care trimite utilizatorul spre pagina cu statistici

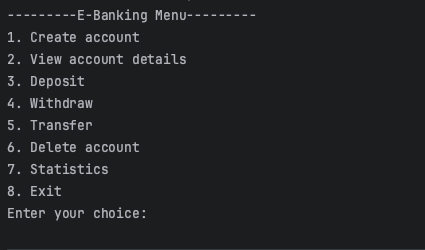
* 1. **Main**

🡪 in main este creata banca si un obiect de tip meniu care primeste ca parametru banca creata

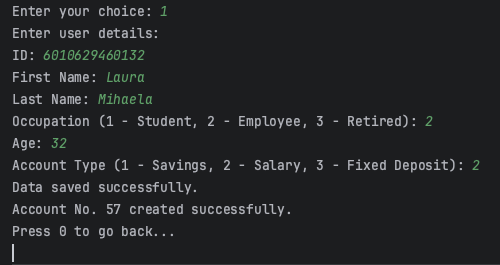
🡪 este apelata functia menu.displayMenu pentru afisarea meniului

# Descriere meniu

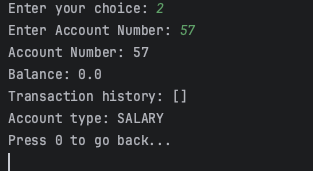
Meniul principal ofera utilizatorilor 8 optiuni disponibile. Se asteapta un raspuns de la utilizator, iar daca optiunea selectata nu exista, este afisat un mesaj.



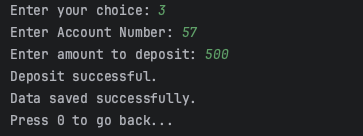
1. **Create account:** aceasta optiune permite crearea unui nou cont de utilizator, iar utilizatorul va treubi sa completeze datele necesare. ID-ul introdus, varsta si restul atributelor indeplinesc conditiile impuse, se genereaza un cont si este afisat un mesaj in care este specificat numarul contului. Utilizatorul poate efectua alte operatii doar prin numarul contului.

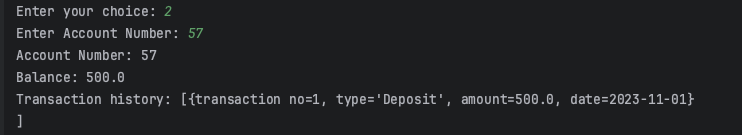


1. **View account details:** optiunea 2 afiseaza detaliile bancare ale contului, daca acesta exsita. Altfel, se afiseaza un mesaj de eroare.



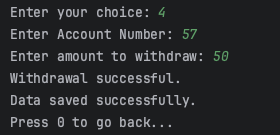
1. **Deposit:** se cer datele contului, iar daca acesta exista se introduce suma dorita pentru alimentarea contului si se afiseaza mesaj de succes. Altfel, se afiseaza un mesaj de eroare.

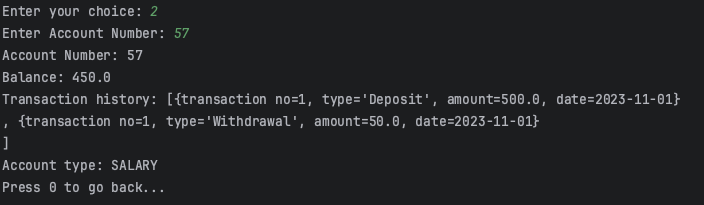




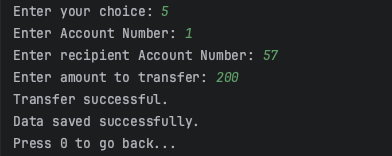
1. **Withdraw:** se cer datele contului, iar daca acesta exista se introduce suma

dorita pentru retragere si se afiseaza mesaj de succes, daca sunt indeplinite conditiile. Altfel, se afiseaza un mesaj de eroare.



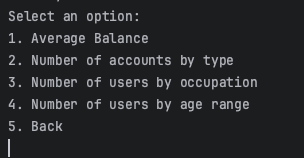


1. **Transfer:** se cer datele legate de contul destinatarului si suma dorita pentru a fi transferata. Daca transferul a avut loc cu succes, se afiseaza un mesaj de succes. Altfel, se afiseaza un mesaj de eroare.





1. **Delete account:** permite stergerea unui cont pe baza numarului de cont
2. **Statistics:** meniul pune la dispozitie 5 optiuni (media soldurilor, numarul de conturi in functie de un anumit tip, numarul de utilizatori in functie de o anumita ocupatie, numarul de utilizatori cu varsta cuprinsa intr-un anumit interval si optiunea de revenire la meniul principal)



1. **Exit**

# Concluzie

În concluzie, aplicația de E-Banking oferă o platformă pentru gestionarea și efectuarea tranzacțiilor financiare într-un mod eficient. Prin organizarea în pachete bine definite, cum ar fi enumerările, excepțiile, modelul, vizualizarea și serviciile, am creat o structură modulară care încurajează o dezvoltare ușoară și întreținere convenabilă. Această documentație vizează să faciliteze înțelegerea și dezvoltarea continuă a aplicației de E-Banking.

# Bibliografie

* <https://www.javatpoint.com/>
* <https://www.geeksforgeeks.org/>