Bibliotecă Online

**Lista JAR-urilor necesare rulării aplicației mele:** [**ojdbc11.jar**](https://download.oracle.com/otn-pub/otn_software/jdbc/232-DeveloperRel/ojdbc11.jar) **și** [**oraclepki.jar**](https://jar-download.com/artifact-search/oraclepki)

Am încercat să implementez un sistem de împrumut/returnare al cărților dintr-o/către bază de date localizată în Oracle Cloud (ce joacă rolul de server) spre/de la un client. Lista cărților poate fi găsită în fișierul ListăCărți.csv, iar scriptul cu care am generat baza de date poate fi găsit în fișierele bookParser.py și dbInserter.py. Astfel, server-ul realizează conexiunea către baza de date autonomă ce rulează pe serverele celor de la Oracle, iar apoi într-o buclă infinită primește conexiuni. Pentru baza de date am creat un user special, doar pentru acest proiect, care are acces de select și update doar asupra tabelei Books, ce are următoarea structură:

NAME VARCHAR2(100 BYTE)

AUTHOR VARCHAR2(100 BYTE)

STOCK NUMBER

PDF BLOB

unde NAME reprezintă numele cărții, AUTHOR reprezintă autorul, STOCK este un număr întreg generat aleatoriu, reprezentând stocul de cărți în biblioteca mea virtuală și PDF reprezintă un obiect binar ce va fi reconstruit sub formă de fișier .pdf în calculatorul clientului.

În cadrul server-ului am creat și un Map<String, SocketAddress> static, unde cheia reprezintă numele clientului, iar valoarea este adresa thread-ului clientului care face request-uri către server. Dacă alt client se conectează cu același nume, nu va putea face request-uri către server. Când clientul va scrie exit, adică se va deconecta de la server, înregistrarea aferentă conexiunii sale se va elimina din Map.

Comenzile pe care le pun la dispoziție la nivelul clientului sunt:

* exit – comanda care oprește execuția clientului și elimină înregistrarea din Map
* caut după <autor> – comanda care interoghează baza de date și returnează un String[] reprezentând stocul fiecărei cărți scrise de autorul dat ca parametru sau un mesaj corespunzător dacă autorul nu este prezent în baza de date
* caut după <titlu> – comanda care interoghează baza de date și returnează un String reprezentând stocul cărții date ca parametru sau un mesaj corespunzător dacă titlul nu se găsește în baza de date
* cumpar <titlu> - comanda care interoghează baza de date și returnează un byte[] dacă titlul dat ca parametru se regăsește în baza de date și are stock-ul peste 0. În caz contrar, afișează un mesaj corespunzător.
* returnez <titlu> - comanda care verifică dacă pe sistemul local se află titlul dat ca parametru, iar apoi dacă acesta există în baza de date; dacă ambele condiții sunt îndeplinite, se șterge fișierul .pdf de pe sistemul local și stocul cărții crește cu 1 în baza de date

Viitoare îmbunătățiri:

* implementarea unui sistem de login mai solid decât cel cu Map
* criptarea datelor de conectare
* separarea părții de Client și Server pe mașini diferite