Токенизация на банкови карти с RMI

В проекта си имам 8 класа: TokenizationServer, TokenizationServerImpl, TokenizationClient, Client, BankClients, Tuple, CardDatabase, UserDatabase.

TokenizationServer представлява интерфейс. Методите му са следните: tokenization (извършва токенизацията, като приема параметър номер на банкова карта), isValid (проверява дали номерът на банковата карта, подаден като параметър, е валиден), isValidUser (проверява дали името и паролата са коректни, задават се като параметри на функцията), getCardNumber (връща номера на банковата карта по даден токен), canRegisterToken (проверява дали потребителят може да регистрира токен), canGetCardNumber (проверява дали потребителят може да извлече номера на картата). canRegisterToken и canGetCardNumber приемат като аргумент име на потребител.

TokenizationServerImlp имплементира TokenizationServer. В този клас реализирам описаните горе функции. Освен тях има метод algLuhn(String str), която по даден стринг str проверява дали той удовлетворява формулата на Luhn.

Също така създава server обект и го регистрира с RMI регистър. Чрез него се стартира сървъра.

Client е клас, който съхранява информация за потребител(име, парола и права). Има 4 член данни : name, password, canRegister и canGetCardNumber. Има конструктор по подразбиране и за общо ползване, set и get методи, също и toString метод.

BankClients е клас, който има за член данна списък от Client обекти. Има метод isCorrect(String name, String password), в който се извършва проверка дали името и паролата на дадения клиент са коректни; canRegisterToken(String name) - по зададеното име проверява дали постребителят има право да регистрира токен; canGetCard(String name) - по зададеното име проверява дали потребителят има право да изисква номер на карта.

Tuple е клас, който съхранява съотношението номер на карта - токен. Има 2 член данни - bankNumber, token. Има конструктор, set и get методи и toString метод. Освен тях има метод sort(List<Tuple> tuple), който връща сортиран списък от номера на карти и съответните им токени, сортиран по банковите карти. Има и метод isUnique(List<Tuple> tuple, String token), който проверява дали съществува токен в записаните досега токени.

UserDatabase е клас, който имплементира Serializable, записва във файл и чете от файл потребителите. Има методи serialize() и deserialize(), които сериализират и десериализират обекти от тип Client. Има метод add(Cllient client), който добавя клиент към списък от клиенти. Чрез writeToFile(String fileName) и readFromFile(Strinf fileName) съответно записва във файл и чете от файл, подаден като аргумент.

CardDatabase е клас, който имплементира Serializable, записва във файл и чете от файл информацията за номер на карт и токен. Има методи serialize() и deserialize(), които сериализират и десериализират обекти от тип Tuple. Има метод add(Tuple t), който добавя наредена двойка номер на карта - токен към списък от клиенти. Чрез writeToFile(String fileName) и readFromFile(Strinf fileName) съответно записва във файл и чете от файл, подаден като аргумент. Има метод getCard(String token), който по даден токен връща номера на картата.

TokenizationClient е JFrame Form, който играе ролята на клент. Има 3 бутона - Log in (за регистрация на потребител с име и парола), Register Token (за регистриране на токен), Get Card Number (за получаване номер на карта). Също така има 4 label-а и 4 текстови полета за име, парола, токен и номер на карта. Чрез Log in бутона се извършва регистрацията на потребител като се правят необходимите проверки. Чрез RegisterToken бутона става регистрирането на токен, след като се направят необходимите проверки. Чрез Get Card Number бутона се извежда номера на картата, след извършването на проверки. Чрез него се стартира клиента.