Софийски университет “Св.Климент Охридски”
Факултет по математика и информатика
Курсов проект по Обектно-ориентирано програмиране с JAVA

Тема:
Криптиране на кредитни карти с RMI

Изготвил:
Христо Георгиев Тодоров
Фак.номер: 61676
Специалност: Софтуерно инженерство

Ръководител:
доц.д-р Евгений Кръстев

Съдържание на документа:

1. Област на проекта.

Основната идея, която стои зад проекта е изграждане на клиент-сървър архитектура с помощта на RMI. Сървърната част е в състояние да добавя нови потребители със съответните права за достъп. Правата за достъп са три: потребители, които могат да криптират, потребители, които могат да декриптират и такива, които са оторизирани да извършват и двете дейности. Освен това сървъра има достъп до всички крептирани и декриптирани кредитни карти и е в състояние във всеки един момент да изиска извеждането им в текстови формат, като данните могат да бъдат сортирани по дв критерия: по номер на кредитна карта или по крептирани номера. Клиентът има възможност за влизане в системата. Той може освен това да криптира и да извежда номер по дадена криптограма в зависимост от правата му за достъп. Криптирането се извършва с цел повишаване на нивото на сигурност при предаване на сензитивна информация по мрежата.

---

2. Логика на проекта.

RMI е начин за постигане на т.нар. разпределено изчисление. В буквален превод от английски означава отдалечено извикване на методи. Традиционните подходи за изпълнение на код на други машини, свързани в мрежа, са колкото объркващи, толкова и досадни и могат да бъдат допуснати грешки при тяхната реализация. Най-конвенционалния начин, по който можем да мислим за този проблем е, че някакъв обект се намира на друга машина и че можете да изпратите съобщение към отдалечения обект и да получите резултат все едно, че обектът се намира на локалната машина. Това опростяване е точно това, което Remote Method Invocation на JAVA позволява на се направи. Единственото, което сървърът, като активна среда, трябва да предостави на клиента е интерфейс. Цялата логика на прогрмата се намира на сървъра.

3. Клиент

Отдалечен интерфейс:

Когато искаме да създадем отдалечен обект, ние скриваме основополагащата имплеметация като подаваме интерфейс. По този начин, когато клиентът получи референция към отдалечения обкет, това което действително получава е референция на интерфейс, която се оказва свързана с някаква част от локален код (stub code), предаван по мрежата. Ние обаче не мислим за това, а просто изпращаме съобщение с помощта на тази референция на интерфейса. Отдалеченият интерфейс носи името **ServiceProvider**. По презумпция той е public (не трябва да разрешава т.нар, "приятелски достъп"). В противен случай клиентът ще получи съобщение за грешка, когато се опита да зареди отдалечен обект, който използва отдалечения интерфейс. **ServiceProvider** е производен на интерфейса **java.rmi.Remote**. Той има три метода:

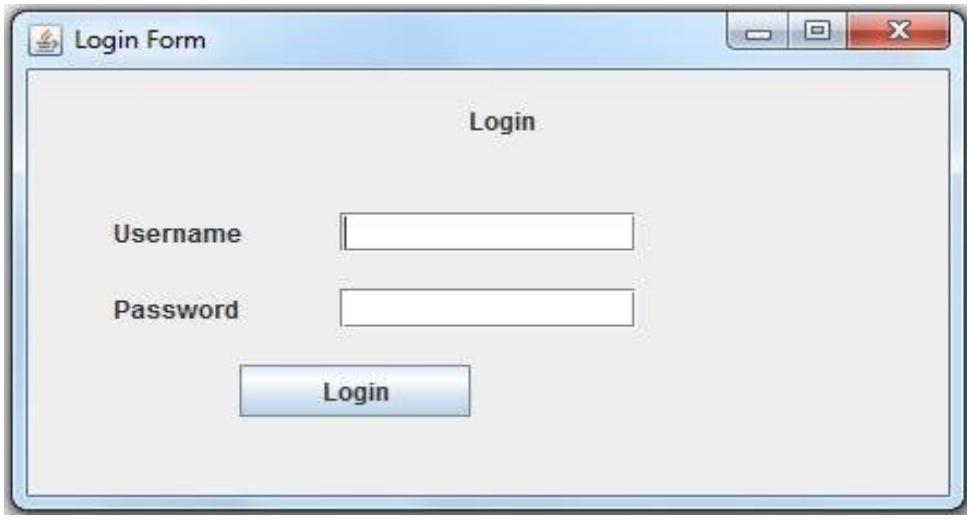
boolean login(String userName, char[] password) throws RemoteException;
String encrypt(String creditCard) throws RemoteException, Exception;
String decrypt(String creditCard) throws RemoteException, Exception;

Всички те в **throws** клаузите си задължително декларират **java.rmi.RemoteException**, наред с останалите, специфични за приложението. Да разгледаме сигнатурата на метода:

boolean login(String username, char[] password) throws RemoteException;

Параметърът username отговаря за потребителското име. А масивът от знаци password, съдържа въведената от потребителя парола.

Чрез имплеметацията на този метод се резлизира влизането на



фиг.1

потребителя. След извикването на този метод се показва форма за въвеждане на потребителски име (поле: Username) и парола (поле: Password). След тяхното изпращане следва проверка дали те са

правилни, ако не са се издава съобщение за грешка. При правилно въведена комбинация потребителско име/парола, на вниманието на потребителя се предоставя интерактивен интерфейс, с помощта на които може да извършва криптиране и декриптиране на различни кредитни карти. Формата, която ще попълва потребителя ще има графичната форма показана на фиг.2. Самата форма се разделя на две секция. Първата секция се казва Encrypt. Криптирането се осъществява чрез метода:

String encrypt(String creditCard) throws RemoteException, Exception;

Параметърът creditCard представлява номерът на кредитната карта, която трябва да се криптира. Чрез попълването на номера на кредитната карта, която трябва да се криптира и при натискането на бутона Encrypt, при правилно криптиране в текстовото поле Encrypted card number се изписва криптираният номер, придружен със диалогов прозорец, съдържащ потвърждение за успешно извършено криптиране. Под успешно криптиране се вземат предивид следните условия:

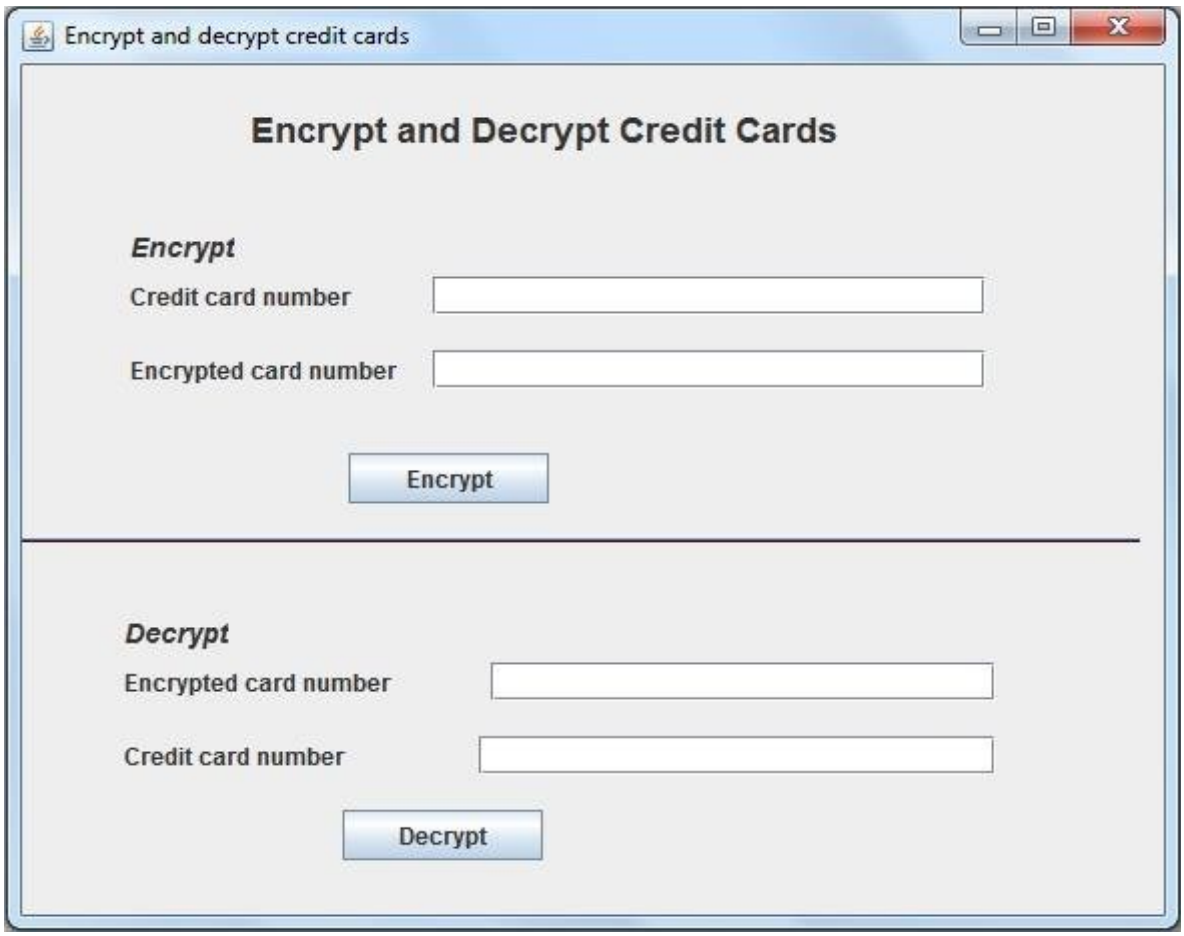
1) Потребителят да има правата за криптиране (т.е. да може или да криптира или да криптира и декриптира);

2) Въведеният номер на кредитната карта да отговаря на формулата на Лун. Формулата на Лун представлява своеобразен стандарт за верифициране на номера на дадена кредитна карта. (Повече информация за формулата на Лун може да се намери на следната уеб страница: http://en.wikipedia.org/wiki/Luhn_algorithm);

3) Въведеният номер трябва да започва с 3, 4, 5 или 6;

4) Въведената карта да е криптирана преди това не повече от 12 пъти.

При неспазване на поне едно от горе споменатите условия се издава съобщение за грешка и криптирането не е успешно.



фиг.2

В секцията Decrypt се позволява извличането на номер на кредитна карта по зададена криптограма. Декриптирането се извършва чрез метода:

String decrypt(String creditCard) throws RemoteException, Exception;

В текстовото поле Encrypted card number се задава криптограмата. Декриптиране се извършва след натискане на бутона Decrypt и при наличието на следните условия:

1) Потребителят, изискващ номера има правата да декриптира (т.е. е с права за декриптиране или с права да криптира и декриптира).

2) Криптограмата, зададена от него отговаря на номер на кредитна карта, която преди това е била декриптирана.

При неналичието на поне едно от гореспоменатите условия се издава съобщение за грешка с описание за вида на грешката.

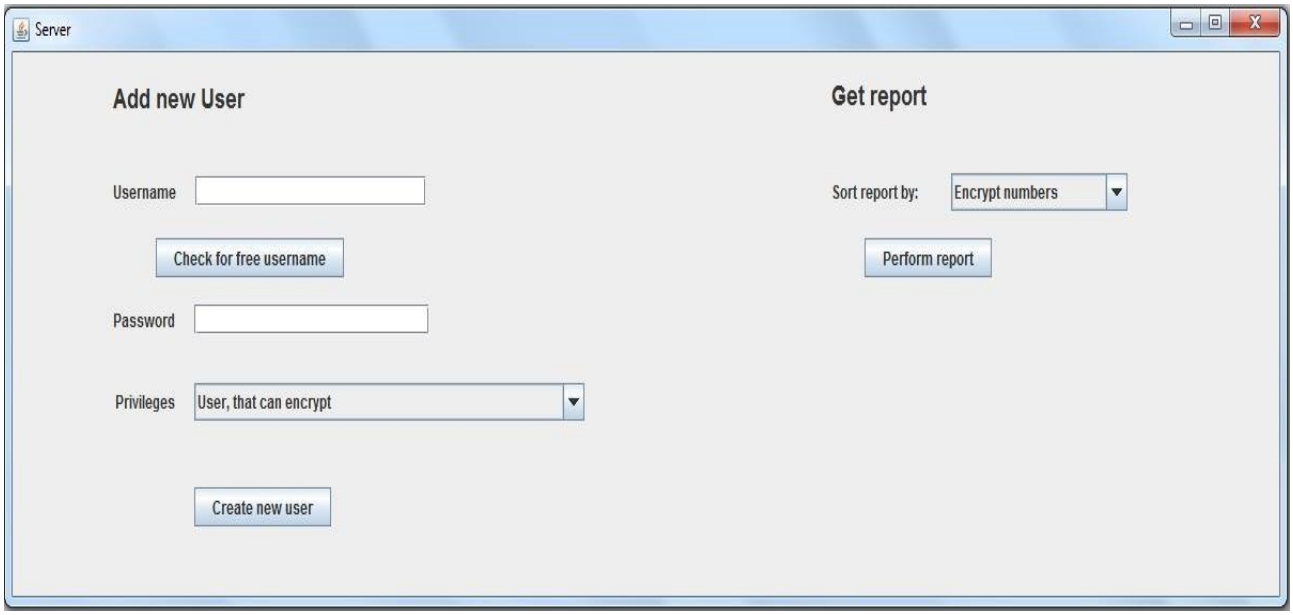
---

4. Сървър

Графичната част е представена на фиг.3.

Интерфейсът на сървъра се състои от две секции. Едната секция се казва Add new User. Тя позволява добавянето на нови потребители. Другата – Get report, позволява извеждането не подорбна справка за

банковите карти и техните криптирани аналози.



Секция "Add new User" *фиг. 3*

В нея има няколко текстови полета. Първото текство поле Username е предназначено за попълване на потребителското име на потребителя. Предоставена е възможност за проверка дали има потребител със същото потребителско име, т.е. дали потребителското име е свободно, посредством бутона Check for free username. Ако е свободно се извежда съобщение, че въведеното потребителско име е свободно. След това в полето Password се попълва паролата. След това от combo box се избира типът на провилегия на потребителя. Създава се нов потребител, след като се натисне бутонът Create new user и ако са налице следните условия:

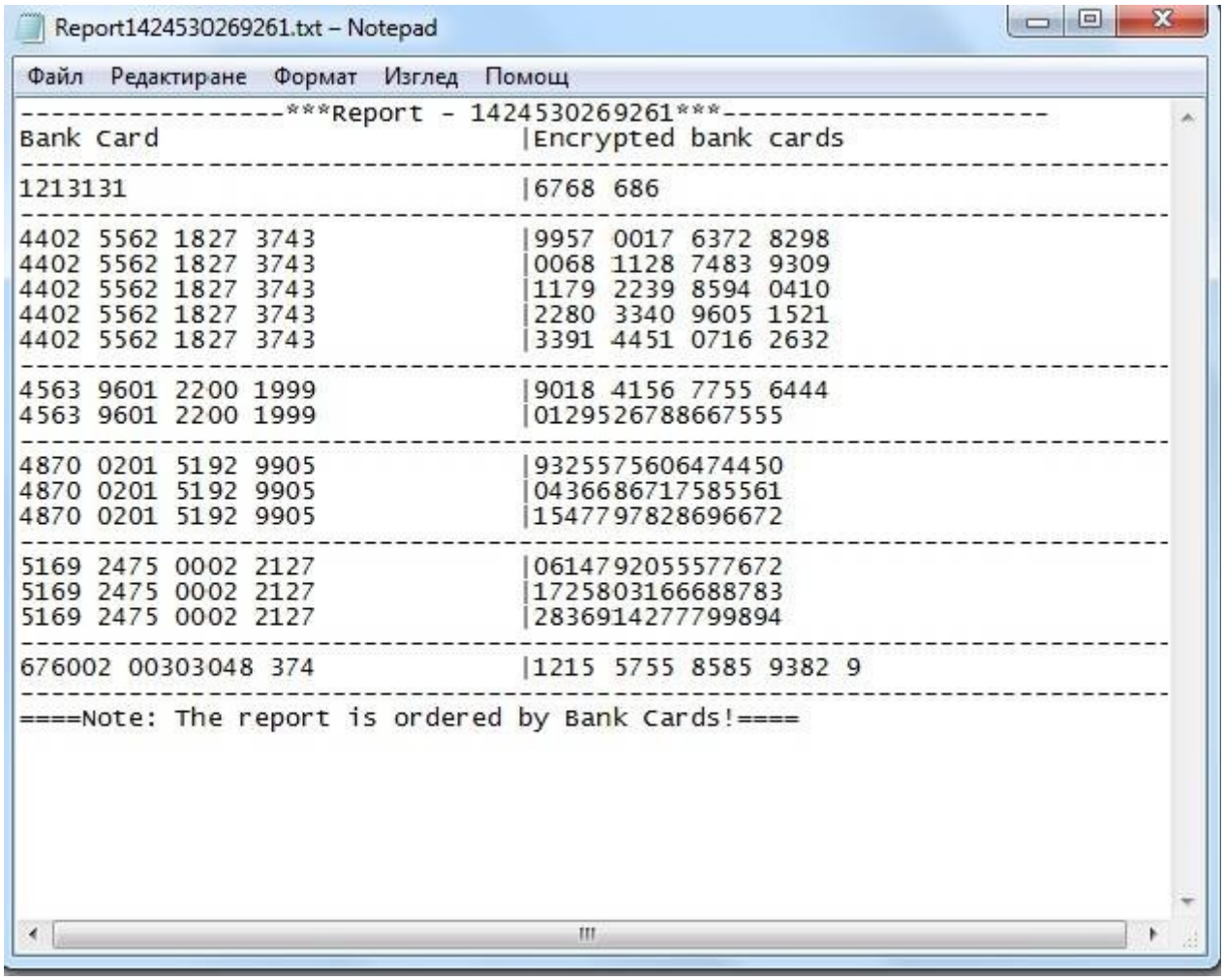
1) Полетата Username и Password не са празни;

2) Въведеното потребителско име е свободно;

При неналичие на поне едно от гореспоменатите условия се извежда съобщение за грешка, съдържащо причина за грешката.

Секция "Get report"

Тази секция позволява извеждането в текстови формат на справка - таблица на криптираните номера и съответните им банкови карти, сортирана по криптираните номера, както и същата справка, само че



сортирана по банкови карти. Различните методи за сортирани се избират с помощта на combo box. След натискането на бутнона Get report се ивежда съобщение с името на справката. Текстовият документ има следният вид:

Данните на потребителите се пазят в XML сериализация.

---

5. Логика на кода

Проектът е реализиран в два пакета.

5.1. Пакет client

5.1.1. Клас ClientLogin – Този клас е Jframe. Има main метод, чрез които се извършва откриването и извличането на отдалечения интерфейс от сървъра. Порта на които се закача потребителят 1099. След обработка на събитието loginActionPerformed в зависимост от това дали влизането на потребителят е успешно, се изписва или съобщение за грешка или се предоставя на вниманието на потребителя формата за криптиране и декриптиране на кредитни карти.

5.1.2. Клас AfterLogin – Този клас служи като графичен интерфейс,

чрез които се обменя информация между клиент-сървъра.
5.2. Пакет server
5.2.1. Клас Broker – Настройва сървъра и публикува интерфейса. Изходна точка за сървър частта (съдържа main метод).
5.2.2. Клас ClientServiceUtillity имплементира интерфейсът ServiceProvider и наследява класът UnicastRemoteObject. Дефинира функционалността, достъпна за клиента.
5.2.3. Клас ServerSide наследява класът JPanel и като такъв се явява графичен интерфейс за сървъра. Освен това той поддържа и основните дейности, които сървъра може да извършва.
5.2.4. Клас User обуславя представянето на потребител, включва полетата username, password и priviliges.
5.2.5. Клас Validator извършва валидация на коректността на въведените от потребителя кредитни карти, освен това чрез метода tokenize() извършва токенизация за аргумента, които му е подаден.
5.2.6. Клас UIServer е JFrame, добавя референция на класа ServerSide.
5.2.7. Интерфейсът ServiceProvider съдържа методите, които се подават на клиента.

---

6. Указания за стартиране на проекта
Неразделна част от този проект е и папката с име: Project_61676. Проектът е разработен изцяло в средата за програмиране NetBeans 8.0.2, използвайки JDK 8. Под средата на NetBeans проектът се стартира по следният начин: File → OpenProject → Project_61676. След това в Source Packages → server с десен клик на мишката върху Broker.java и избор на опцията Run се стартира сървърната част. Клиентската част може да бъде стартирана успешно само, ако преди това сървърната е стартирана. За да се стартира клиентската част – десен клик на мишката върху ClientLogin в пакета client, опцията Run.

---

7. Използвана литература и източници:
http://de.wikipedia.org/wiki/Luhn-Algorithmus#Java .
Да мислим на JAVA т. 2, Брус Екел, 2001, СофтПрес, с. 415 – с. 422
Java Platform Standard Edition 8 Documentation - http://docs.oracle.com/javase/8/docs/ .
Лекции на доц. д-р Евгений Кръстев четени по курса ООП с JAVA, зимен семестър 2014 – 2015, СУ, ФМИ.
Статии в http://stackoverflow.com/ .