

**Софийски Университет„Св. Климент Охридски”**

**Факултет по математика и информатика**

**Обектно Ориентирано Програмиране**

**(OOП с Java)**

**Криптиране на банкови карти с RMI**

**Изготвил**: Даниел Димитров, фак. номер: 62121, 3 курс, Софтуерно инжерество

Дата: 15.02.2020

1. **Цел на проект**

Цел на проекта е реализация на потребителско клиент-сървър приложение за криптиране на банкови карти. Основната услуга, която получават потребителите, е че могат да криптират или декриптират своите банкови карти, като за целта те трябва първо да се впишат в системата и да имат съответните права на достъп. Сървърът валидира данните на потребителите и ако имат съответните права извършва поисканата от тях услуга. Информацията за потребителските акаунти и правата им се пази в XML файл. Сървърът позволява и извеждането в текствов файл на таблица, в която се съдържа информация за кредитните карти и техните криптирани номера, като има възможност за сортиране на данните по определен признак. Системата се реализира чрез **RMI**.

1. **Архитектура на системата**

**2.1 Сървърна част**

* server.app – пакет, в който се намира графичната част и изпълнимият файл за пускане на сървъра
* serverRMIDefinitions – пакет, който съдържа интерфейса на услугата и обект с реализия на тази услуга, които ще извършва сметките
* substitutionCypher – пакет, който съдържа шифъра, които ще криптира данните по определен метод
* userPackage – пакет, който съдържа класове необходими за обработка на потребители
* Utilities – пакет, който съдържа функции за валидация на потребителски данни и извеждане на съобщения
* filesOperations – пакет, съдържащ необходимите класове за входно-изходни операции с файлове

**Класове**

* Server.java - хоста за стартиране на сървърната част от системата, който предлага услугата за криптиране и декриптиране на клиента
* Controller.java – клас, който управлява графичният интерфейс на сървъра
* samlple.fxml – съдържа дизайна на графичния интерфейс на сървъра
* ServerObjectInterface.java – интерфейс, описващ основните услуги на сървъра
* ServerObjectInterfaceImplementation.java – клас, който имплементира основните услуги на интерфейса
* EncryptCard.java – клас, които предоставя услуга за криптиране на банкова карта
* Privileges.java – изброим тип, определящ категориите на потребителите
* User.java – клас, описващ потребител
* Users.java – клас, wrapper на User, за записване в XML формат
* Utils.java – клас, съдържащ методи за валидация на банкови карти
* BankCardByCardNumberStream.java – клас, който записва таблица във файл, сортирана по номер на банкова карта
* BankCardByEncryptionStream.java – клас, който записва таблица във файл, сортирана по криптираните номера на банкова карта

**2.2 Клиентска част**

* client.app – пакет, съдържащ логиката и графичната среда на клиента
* clientUtils – пакет, съдържащ методи за проверка на входните данни на клиента

**Класове**

* Client.java – клас, стартиращ потребителското приложение
* Controller.java – клас, който управлява графичната среда на клиента
* sample.fxml – съдържа дизайна на графичния интерфейс на сървъра

1. **Структури от данни и алгоритми**

* Substitution cypher – извършва се отместване на символи на определено място в криптирания текст.
* Luhn algorithm – всяко второ число отдясно наляво се умножава по 2 и ако се получи двуцифрено число се взима сбора на цифрите му. Събират се с другите числа и резултатът се дали с отстатък на 10. Ако се получи резултат 0, то банковата карта е валидна.

1. **Проблеми**

* **Xstream** парсерът за обработна на XML файлове не поддържа модулно програмиране и няма как да бъде добавен към модул.

1. **Тестване на системата**

Добавени са по един акаунт от трите основни групи потребители. Администратор admin с парола a123, потребител user с парола u123 и гост guest с парола g123. Добавени са и четири банкови карти като три от тях са криптирани по 5 пъти, а последната само 1 път. Шифърът работи със стандартно отместване 5.

3379513561108795 - 5 пъти

4563960122001999 - 5 пъти

6011456738483344 - 5 пъти

5319699240514459 - 1 път

Всяка банкова карта започва с начално криптиране:

3379513561108795 - 8824068016653240

4563960122001999 - 9018415677556444

6011456738483344 - 1566901283938899

5319699240514459 – 0864144795069904

Всички получени резултати са записани в sortedByCardNumber.txt и sortedByEncryption.txt като разликата между тях е в признака на сортиране. Информацията за всеки потребител се записва в data.xml . Тези файлове се намират в src папката на проекта.