**Курсов проект по Обектно-ориентирано програмиране с** **Java**

Проект №3: Криптиране на банкови карти с многонишков сървър

Разработено от: Виктор Христов, №62151

Дата: 14.02.2020

1. **Инструкции за компилиране**

* За компилиране на програмата е нужна библиотеката JavaFX.
* Задължително е да се стартира сървъра преди клиента. В противен случай клиентът ще се опита да се свърже към несъществуващ сокет и ще върне NullPointerException.

1. **Общ дизайн**
2. Модул **client**, който включва:
3. Пакет **wrappers** - съдържа класове, които обвиват информация в себе си, за да може лесно да се различат видовете заявки при комуникация между клиента и сървъра (използва се и от клиента, и от сървъра);
4. Клас **Client** – стартира графичния интерфейс на клиента;
5. Клас **ClientController** – управлява функционалностите на клиента и как той комуникира със сървъра
6. Модул **server**, който включва:
7. Пакет **server**.**app**, който реализира сървърното приложение чрез следните класове:

* Клас **Server** - стартира графичния интерфейс на сървъра;
* Клас **ServerController** - управлява функционалностите на сървъра и как той комуникира с клиента;
* Клас **RegistrationController** – управлява допълнителния прозорец за регистриране на потребител, който се достъпва чрез графичния интерфейс на сървъра

1. Пакет **utilities**, който помага да се реализира сървърното приложение чрез следните класове:

* enum **AccessRights** – стойности, които съответстват на правата на потребителите;
* Клас **BankCardFileControl** – управлява как се записват номерата на картите във файлове;
* Клас **SubstitutionCipher** – задава методите, по които се криптират и декриптират картите;
* Клас **User** – групира информацията за отделните потребители, която включва потребителско име, парола и право на достъп;
* Клас **UserWrapper** – обвиващ клас, който запазва информацията за всеки потребител в контейнер и дава възможност да го записва във файл или да го чете от файл;
* Клас **Validation** – задава методите, по които се валидират номерата на картите

1. **Използвани алгоритми**
2. Substitution Cipher – използван за криптиране и декриптиране на картите
3. формулата на Luhn – използван за валидация на номерата на банковите карти
4. **Тестване на проекта**
5. За тестването на регистрацията на потребител, свързване на клиент със сървър чрез логин и валидация на потребителските права при криптиране и декриптиране се използват потребители със следните данни:
6. Username: admin Password: admin Access Rights: FULL
7. Username: encman Password: 123456 Access Rights: ENCRYPTION
8. Username: decman Password: 654321 Access Rights: DECRYPTION
9. Username: nullman Password: null Access Rights: NONE
10. Username: fullman Password: full Access Rights: FULL
11. За тестване на валидационните алгоритми и попълването на картите в таблица са въведени следните двойки карти:
12. Номер на карта: 3084404451067497 Криптограма: 8539959906512942
13. Номер на карта: 3656969954921240 Криптограма: 8101414409476795
14. Номер на карта: 3958237601947280 Криптограма: 8403782156492735
15. Номер на карта: 4563960122001999 Криптограма: 9018415677556444
16. Номер на карта: 4976737695168251 Криптограма: 9421282140613706
17. Номер на карта: 6695620593399365 Криптограма: 1140175048844810
18. **Проблеми с реализацията на проекта**
19. Текущи проблеми
20. Графичният потребителски интерфейс на сървъра не е пакетиран и отделен като Jar файл;
21. Поради проблем с интегрирането на XStream със проекта, записването на регистрираните потребители е извършено с помощта на ObjectOutputStream и ObjectInputStream;
22. Не се брои колко пъти е криптирана една карта и отместването на шифъра не се променя;
23. Записването на информацията за картите е извършено чрез Formatter и Scanner, вместо да се използва NIO и Streams;
24. Бутонът за извеждане в текстов файл на таблица на номерата, сортирана по криптираните номера извежда таблицата в текстовото поле на сървъра, вместо да отваря самия файл;
25. Бутонът за извеждане в текстов файл на таблица на номерата, сортирана по банковите карти извежда таблицата в текстовото поле на сървъра, вместо да отваря самия файл
26. Срещнати валидационни проблеми за бъдещо подобряване на проекта
27. Един клиент може да се свърже със сървъра чрез един и същ профил множество пъти;
28. При регистриране на потребителски профил, един потребител може да се запише с едни и същи данни повече от веднъж;
29. Тъй като декриптирането също попълва запис в таблицата и се валидира само с регулярен израз, възможно е в таблицата да се запише номер на карта, която не е валидна спрямо формулата на Luhn
30. **Литературни източници**
31. How Security Can Help Grow Your Business: The Marketing Side of Tokenization, <http://merch.bankofamerica.com/documents/10162/50824/How+Security+Can+Help+Grow+Your+Business.pdf>
32. P. Deitel, H. Deitel, "Java How to Program (early objects)", Prentice Hall 9 ed. 2012, ISBN-10:0132575663, ISBN-13: 978-0-13-257566-9 (основна)
33. Y. Daniel Liang, “Introduction to Java Programming”, 7th ed.,Prentice Hall 2009 ISBN-13: 978-0-13-601267-2
34. Bruce Eckel “Thinking in Java”, 4th ed., Prentice Hall 2006 или българското й издание "Да мислим на Java" том 1 и 2, SoftPress,2001