**Курсов проект**

**Обектно Ориентирано Програмиране  
(OOП с Java)**

**Токенизация на банкови карти с многонишков сървър**

Проектът реализира многонишков сървър-клиент за регистрация и токенизация на банкови карти. Всеки клиент се регистрира и после идентифицира с потребителско име и парола. При коректни такива, потребителят влиза в сесия.

**Пакет primitives**

В **пакет primitives** се намират реализациите на основните класове в проекта. Това са: User, CreditCard, Token и Database.

Kлас **User** описва потребителят, ползващ приложението. Той се характеризира с име и парола, булеви флагове за права и структура от данни Map, в която се съхранява съотношението на карта <-> токен.

Клас **CreditCard** описва една кредитна карта. Извършват се проверки за коректността ѝ. Метод **tokenize()** връща токен.

Клас **Token** описва създадения токен.

Клас **UserDatabase** описва базата от данни посредством XML сериализация.

**Пакет comparators**

Компаратори, използващи се при сортирането на картите.

**CardComparator -**  Сравнява кредитните карти.

**TokenComparator -** Сравнява токените, с които са регистрирани картите.

**Пакет client.gui**

В него са реализирани 3 JavaFrame класа, осигуряващи графичния потребителски интерфейс.

**SignFrame -** Графичен интерфейс, позволяващ на клиента да се регистрира или идентифицира

**InSessionFrame** - Графичен интерфейс, който позволява избирането на опции от потребителя, когато е в сесия.

1) **Регистрация на карта** - Потребителят въвежда коректна кредитна карта. В случай, че не е, бива сигнализиран. При успех се извежда съобщение , съдържащо автоматично генерирания токен.

2) **Извличане на карта по токен -** Потребителят въвежда токен. Ако такъв съществува в базата от данни, се извежда съобщение, съдържащо номера на картата. Извежда се съобщение за грешка, ако не съществува или ако потребителят няма права.

3) **Експортиране** на всички регистрирани от потребителя карти във файл, сортирани по номера на картата или по номера на токена. В случай на липса на подобни права се извежда съобщение за грешка.

**CardRegFrame -** реализира прост Input Box, който приема номер на кредитна карта. Ако номера е невалиден, то се извежда предупредително съобщение в JOptionPane прозорец.

**Пакет communication**

Съдържа всички класове, нужни за комуникацията между клиента и сървъра.Главният от които е:

**ClientServerCommunication -** В него се отваря сокет между потребителя и сървъра, както и нужните потоци за вход и изход на данните.

Клас **Message** осигурява един от типовете предавани обекти по сокета.

Клас **AckHandler -** Вж. "Бъдещи подобрения"

Последния пакет се казва server.Той съдържа – Server.java в който е реализиран многонишковия сървър , Session(потребителска сесия), която се използва когато клиентът се идентифицира .

**Пакет server  
 Server.java -** съдържа многонишковия сървър

**WorkerRunanble.java**

**Пакет еxceptions -** съдържа всички видове изключения, които могат да се "хвърлят".

**Бъдещи подобрения**

Вместо switch в proccessAction() в server.WorkerRunnable.java може да се използва структурата Маp<String, Runnable>. За всеки зададен стринг да отговаря един Runnable interface, т.е. функция.