**Имена:**  Иван Ивов Чучулски **фн:**  *4MI3400043***Начална година:** 2021г. **Програма: магистър ЗИКСМ** **Курс: 1**   
**Тема: Приложение за съхранение на пароли  
Дата: 2022-01-22 Предмет: netJava2021 имейл: ivan.iv.chuchulski@gmail.com**

**преподавател:** доц. д-р Милен Петров

**Предаване:** Задачата се предава в архив с попълнен настоящия документ, проекта/проектите с кодовете, README.txt файл, който описва съдържанието на архива; папка с допълнителни компоненти и използваниресурси**, архива да се казва netJava2015\_fn9999\_FinalProject.zip. (Успех!). (Редактирайте маркираните зони в жълто с коректната информация)**

# ТЕМА: Приложение за съхранение, проверка и генериране на сигурни пароли

(курсов проект по netJava2021)

## 1. Условие

Целта на проекта е да се проектира и имплементира клиент-сървър приложение, което да позволява на потребителите да записват пароли за използвани от тях уеб сайтове или други услуги . Приложението ще предоставя възможност за генериране на сигурна парола за конкретна услуга, ако потребителите не желаят те да измислят такава.

## 2. Въведение

В днешни времена, когато технологиите са навлезли във почти всеки един аспект от живота ни и едно от най-важните неща е да предпазваме личните си данни от изтичане в Интернет. Текстовите пароли са изключително чувствителна информация, която трябва да се знае единствено от потребителя, тъй като чрез нея той удостоверява своята самоличност в услугите, които използва.

Една от често срещаните грешки, която доста хора правят е преизползването на една и съща парола в множество различни сайтове и услуги. Също така често самите пароли са слаби, тъй като са къси и се състоят от рождени дати и имена. Проблемът често се корени в това, че потребителите не могат да помнят много на брой дълги и разнообразни пароли.

Един начин да подобрим своята сигурно е като използваме приложение, което да реши проблема със запомнянето на паролите, а също така и предоставя възможност за генериране на достатъчно сигурна парола вместо потребителя.

## 3. Теория

При съхранението на чувствителна информация като пароли за достъп е необходимо да се предприемат мерки за защита от неоторизиран достъп, модификация или кражба. За целта ще използваме концепциите за криптографска хеш функция и симетрично криптиране на данните.

Паролите, които потребителите подават при регистрация в системата ще преобразуваме с помощта на криптографска хеш функция и ще пазим в базата от данни резултата от пресмятането, което е байтовете на техния хеш. При опит за влизане в системата се пресмята хеша на въведената от потребителя парола и за успешен вход е необходимо той да съвпада с този в базата данни. Допълнителна сигурност ще ни донесе използването на т.нар. salt – това е поредица от байтове, която произволно генерираме при регистрация, долепяме към подадената парола и така пресмятаме хеша. По този начин се предпазваме от варианта, в който паролите за достъп на потребителите съвпадат и резултата от хеш функцията ще е еднакъв, а също така и на практика обезсмисляме много от brute-force атаките и атаките с rainbow таблици. Ще използваме 16-байтов salt и криптографската хеш функцията SHA-512.



Фигура 1 Хеширане на парола със salt

Тъй като приложението цели да предостави възможността на потребителите да могат да записват пароли за използвани от тях услуги е нужно записаната информация да бъде записана достатъчно сигурно, но да може да възстановена в явен вид. Това се постига с използването на симетрична криптографска функция, която използва един частен ключ, който трябва да остане таен за криптиране и декриптиране на информацията.



Фигура Симетрично криптиране

## 4. Използвани технологии

За реализирането на приложението е използван езика Java SE, версия 17.0.1. За базата данни е използвана СУБД MariaDB версия 10.4.11, под формата на софтуера “XAMPP for Windows” версия 7.4.2. За изпълнение на програмата, тестовете, както и за набавянето на нужните библиотеки се използва инструментът за автоматизация Gradle, версия 7.3.

Използвани са следните Java софтуерни библиотеки :

* GSON 2.8.9 за работа с JSON формат
* Mockito 3.7.7, използвана при тестовете на класовете, използващи уеб услугите
* MariaDB Java client 2.1.2, конектор за осъществяване на връзка с базата данни
* Junit 4.13, за изпълнение на тестовете на програмни единици

## 5. Инсталация и настройки

...

## 6. Кратко ръководство на потребителя

...

## 7. Примерни данни

...

## 8. Описание на програмния код

...

## 9. Приноси на студента, ограничения и възможности за бъдещо разширение

Възможни разширения на функционалността на приложението може са

* добавяне на графичен-потребителски интерфейс на клиентското приложение
* създаване на администраторско приложение с графичен-потребителски интерфейс за следене на системата
* възможност потребител да експортира/импортира записаните пароли
* възможност за автоматично попълване на форма със съответната парола и копиране на парола на clipboard-а на потребителя
* възможност за съхранение на друг вид удостоверителна информация като API ключове

## 10. Какво научих

...

## 11. Използвани източници

[1] "Java NIO", Ron Hitchens, O'Reilly Media 2002

[2] "Pro Java 7 Nio.2", Anghel Leonard, Apress 2011

[3] "Hashing a Password in Java", Eugen Paraschiv, 12.01.2022, https://www.baeldung.com/java-password-hashing

[4] "Java AES Encryption and Decryption", Eugen Paraschiv, 14.11.2021, https://www.baeldung.com/java-aes-encryption-decryption

[5] "Java AES encryption and decryption", Yong Mook Kim, 02.06.2020, линк: https://mkyong.com/java/java-aes-encryption-and-decryption/

[6] "Cryptography 101 for Java developers by Michel Schudel", Michel Schudel,

линк: <https://github.com/MichelSchudel/crypto-demo>,

видео презентация: https://www.youtube.com/watch?v=1925zmDP\_BY

[7] "Java Cryptography Architecture (JCA) Reference Guide", линк: https://docs.oracle.com/en/java/javase/17/security/java-cryptography-architecture-jca-reference-guide.html

[8] Услуга за проверка на сигурността на пароли, линк: https://www.enzoic.com/docs-passwords-api/

[9] Услуга за генериране на пароли, линк: https://happi.dev/docs/password-generator

[10] "Exploring the New HTTP Client in Java", Eugen Paraschiv, 31.12.2021, линк: https://www.baeldung.com/java-9-http-client

[11] "Java 11 HttpClient Examples", Yong Mook Kim, 17.05.2020, линк: https://mkyong.com/java/java-11-httpclient-examples/

[12] "Sharing Build Logic between Subprojects", https://docs.gradle.org/current/userguide/sharing\_build\_logic\_between\_subprojects.html

[13] "Declaring Dependencies between Subprojects", <https://docs.gradle.org/current/userguide/declaring_dependencies_between_subprojects.html>

[14] Фигура 1. линк: https://www.okta.com/blog/2019/03/what-are-salted-passwords-and-password-hashing/

[15] Фигура 2. линк: https://sectigostore.com/blog/5-differences-between-symmetric-vs-asymmetric-encryption/