UPB – Zadanie 4

IMPLEMENTÁCIA SPRÁVY POUŽÍVATEĽSKÝCH HESIEL

```
public class Registration {
    protected static MyResult registracia(String meno, String heslo) throws NoSuchAlgorithmException, Exception{
        if (Database.exist("hesla.txt", meno)){
            System.out.println("Meno je uz zabrate.");
            return new MyResult(false, "Meno je uz zabrate.");
        }
        else {
            Random random = new Random();
            //GENEROVANIE SALT
            int salt = random.nextInt(Integer.MAX_VALUE - 1) + 1;
            //HASHOVANIE
            Database.add("hesla.txt", meno + ":" + Hashing.toHex(Integer.toString(salt), heslo) + ":" + salt);
        }
        return new MyResult(true, "");
    }
    }
```

V tejto časti sa generuje **SALT**, následne sa vykonáva **HASHING** hesla pomocou funkcie ***Hashing***.

Údaje sa ukladajú vo forme (**meno**, **HASH**, **SALT**):

meno:DFA7DDEA8B4ED94D806E5B22CFCF610BDA0CE56DB0A098D0F6B829A7A5216F0FE18AD
D20155580D636839F134A8AC8FD55A0A7ACC0BEFF8F4148D7BFC57B9101:2019812137

```
public class Hashing {
    public static String toHex(String salt, String heslo) throws NoSuchAlgorithmException {
        String password = salt + heslo + salt;

        MessageDigest digest = MessageDigest.getInstance("SHA-512"); //HASHOVANIE SHA-512
        byte[] encodedhash = digest.digest(
                password.getBytes(StandardCharsets.UTF_8));

        StringBuilder sb = new StringBuilder();

        for (byte b : encodedhash) {
            sb.append(String.format("%02X", b)); //PREVOD NA HEXA
        }

        return sb.toString();
    }
}
```

Takto vyzerá funkcia na **HASHING**, využíva sa **SHA-512**

```java
public class Login {

    protected static MyResult prihlasovanie(String meno, String heslo) throws IOException, Exception{
        MyResult account = Database.find("hesla.txt", meno);
        if (!account.getFirst()){
            return new MyResult(false, "Nespravne meno.");
        }
        else {
            StringTokenizer st = new StringTokenizer(account.getSecond(), ":");
            st.nextToken();      //prvy token je prihlasovacie meno
            String hash512 = st.nextToken(); //HESLO
            String salt = st.nextToken(); //SALT
            boolean rightPassword = Hashing.toHex(salt, heslo).equals(hash512); // KONTROLA HESIEL
            if (!rightPassword) {
                Thread.sleep(100); //DELAY pred kazdym prihlasenim
                return new MyResult(false, "Nespravne heslo.");
            }
        }
        return new MyResult(true, "Uspesne prihlasenie.");
    }
}
```

V tejto časti prebieha kontrola zadaného hesla s heslom v databáze. Pred každým overovaním hesla prebehne kratší ***delay***, ktorý bráni pred *brute force* útokom. *Delay* sa spustí v prípade že je zadané zlé heslo. Čas je krátky, takže človek si to nepovšimne ale voči útokom je to efektívne.

PS: Ten .jar mi ani pána nejde vo Visualku otvoriť ja Ideu sa mi sťahovať nechce. Funguje to, keď to pekne otvoríš, čo dúfam že zvládneš, keďže si tiež toto zadanie robil.