

Bsp_405_BruteForce

In der Datei passwd_file.csv befinden sich Vorname, Nachname, Passwort und Passwort-Hash von 100 verschiedenen Usern:

```
first_name,last_name,passwd,hash
Lamar,Shinton,9j0g3,0216f81e9af45048d6960b271f306243
```

Das Passwort besteht aus genau 5 Zeichen, die entweder Kleinbuchstaben (a-z) oder Ziffern (0-9) sind. Der Hash ist MD5-verschlüsselt und wird aus einem String generiert, der aus Vorname-Nachname+ Passwort besteht.

Erstelle die **BruteForceManager** Klassen, **BruteForceWorker** und **Person**

BruteForceWorker

- ☒ Der Konstruktor bekommt ein Objekt der Klasse Person übergeben.
- ☒ Callable-Klasse, die entweder das gefundene Passwort oder null zurückgibt.
- ☒ Ausgabe auf der Konsole welcher Thread gerade läuft.
- ☒ Mittels BruteForce wird versucht das Passwort zu knacken indem alle möglichen Kombinationen von Passwörtern erzeugt und verschlüsselt werden und anschließend mit dem Hash verglichen werden.
- ☒ Umwandeln des `byte[]`-Hash in einen Hex-String mit der Methode:
`DatatypeConverter.printHexBinary()`
- ☒ Sobald das Passwort gefunden wurde bricht der Thread ab.
- ☒ Ausgabe der Zeitmessung wie lange der gebraucht Thread hat (in Sekunden und Millis)

BruteForceManager

- ☒ `loadData()` : liest alle Benutzerdaten von der Datei auf eine Collection ein.
- ☒ `crackPasswords()` : Erzeugt einen Pool bestehend aus 4 Threads.
Einlesen wie viele Benutzer gehackt werden sollen. Für jeden Benutzer wird ein Thread in den Pool submittet.
Ein `CompletableFuture` nimmt die gehackten Passwörter entgegen und gibt sie auf der Konsole aus.

Person

- ☒ Datenklasse für eine Person, die Vornamen, Nachname, Passwort und Hash enthält.
- ☒

Lernstoff:

- ☐ **Threads**
- ☐ **Callable, Future**
- ☐ **ThreadPool, ExecutorService, CompletableFuture**
- ☐ Verschlüsselung: **MessageDigest, DatatypeConverter**

Umfang: 2 DP-Std