

# Portfolioprüfung – Werkstück A – Alternative 4

## 1 Aufgabe

**Entwickeln Sie einen Passwort-Manager als Kommandozeilenanwendung.**

Im heutigen Internet-Umfeld benötigen die Nutzer im Alltag eine Vielzahl unterschiedlicher Passwörter. Damit diese ausreichend sicher sind, sollten sie möglichst lang und komplex sein und zudem nicht mehrfach verwendet werden. Diese Bedingungen machen es schwierig, einen Überblick über die eigenen Passwörter zu behalten. Ein Passwort-Manager kann hierbei Abhilfe schaffen.

Ein Passwort-Manager funktioniert im Prinzip wie ein Tresor. Beim Erstellen eines neuen Passworts speichert die Anwendung dieses ab. Alternativ kann der Passwort-Manager auch ein sicheres Passwort generieren.

## 2 Anforderungen

- Benutzer sollen Einträge mit einem Titel, Benutzernamen und Passwort speichern können.
- Benutzer sollen Einträge mit der Anwendung auch selbst löschen können. Vor dem Löschen soll aber noch einmal interaktiv nachgefragt (bestätigt) werden.
- Die gespeicherten Daten sollen sicher und persistent im Dateisystem abgelegt werden.
- Passwörter sollen automatisch generiert werden können. Dazu sollen Benutzer verschiedene Optionen angeben können (Länge, Groß-/Kleinschreibung, Sonderzeichen/Symbole, etc.).
- Der Passwort-Manager soll mit einem Master-Passwort gesichert werden. Dieses wird vom Benutzer initial selber gewählt und kann mit Hilfe der Anwendung geändert werden.
- Beim Eingeben soll das Master-Passwort nicht im Klartext erscheinen.
- Das Master-Passwort wird nicht bei jeder Abfrage angefordert, sondern wird nach einer einmaligen Eingabe für eine vom Benutzer festgelegten Zeit nicht abgefragt. Legen Sie hierfür selbst eine sinnvolle Zeitspanne (5min - 24h) fest.
- Passwörter sollen nicht im Klartext angezeigt werden. Beim Abrufen soll ein Passwort automatisch in der Zwischenablage (*Clipboard*) gespeichert werden, sodass dieses direkt auf z.B. einer Webseite eingefügt werden kann. Ein Passwort darf höchstens 30 Sekunden in der Zwischenablage gespeichert werden.

- Es soll die Möglichkeit geben, über die Anwendung alle Einträge als Tabelle in der Shell auszugeben.
- Denken Sie sich mindestens zwei weitere (sinnvolle) Funktionen aus und implementieren Sie diese.
- Implementieren Sie die Lösung so, dass nicht zuerst ein Skript gestartet werden muss, sondern ein eigener Kommandozeilen-Befehl existiert (siehe Abschnitt 3).
- **Entwickeln und implementieren Sie Ihre Lösung als Bash-Skript, als C-Programm oder als Python-Skript** als freie Software (Open Source) und verwenden Sie hierfür ein Code-Repository, z.B. bei GitHub.
- Der Quellcode soll durch Kommentare verständlich sein.
- Bearbeiten Sie die Aufgabe in Teams zu **3 Personen**.
- Schreiben Sie eine aussagekräftige und ansehnliche Dokumentation (Umfang: **8-10 Seiten**) über Ihre Lösung.
- Die Funktionalität der Lösung müssen Sie in der Übung demonstrieren. Bereiten Sie einen Vortrag mit Präsentationsfolien und eine Live-Demonstration (Umfang: **15-20 Minuten**) vor.

### 3 Beispielhafte Befehle

Neues Passwort für Moodle generieren:

```
$ passman add -title moodle -username uas123456 -generatepassword  
--> OK password created and copied to clipboard
```

Passwort für Moodle abrufen:

```
$ passman copy -title moodle  
--> Master-Password:  
--> Master ok  
--> Username: uas123456 | password copied to clipboard
```