Passwortmanager

Rania Hajjout, Issam Boutachdat, Luke Cefariello

Aufbau des Programms

- Alle wichtigen Module importiert
- Entsprechende Klassen erstellt
- Methoden für Funktionen erstellt (z.B Passwortgenerator)
- Main-Part (mit Befehle und Methoden)
  - → Generate
  - → Store
  - →Delete
  - → Modify
  - → List
  - → Search
  - → Change
  - → Check
  - → Exit

Befehl: „generate“

Probleme – Lösungen

```
if laenge >= anzahl_anforderungen and anzahl_anforderungen >= 1:
    #Abfragen ob User gewisse Option haben möchte oder nicht
    while laenge > len(kombiniert):

        if soll_zahlen_enthalten:
            kombiniert += secrets.choice(zahlen)

        if soll_kleinbuchstaben_enthalten and laenge > len(kombiniert):
            kombiniert += secrets.choice(kleinbuchstaben)

        if soll_grossbuchstaben_enthalten and laenge > len(kombiniert):
            kombiniert += secrets.choice(grossbuchstaben)

        if soll_sonderzeichen_enthalten and laenge > len(kombiniert):
            kombiniert += secrets.choice(sonderzeichen)

    #Eingabe Passwortlänge mit for-schleife durchlaufen, um Optionen (siehe oben einzufügen und zu


    #hier wird das Passwort durchgemischt!
    pw = ''.join(random.sample(kombiniert,len(kombiniert)))

    print("Passwort: " + verdecken(pw) + " in Zwischenablge gespeichert.")

    zwischenablage_speichern(pw)
    time.sleep(0.2)
    passwortsicherheit(pw)

elif laenge == 0:
    print("Passortlänge muss mindestens 1 betragen.")
else:
    print("Sie können kein Passwort generieren mit " + str(anzahl_anforderungen) + " verschiednen
```

Problem/ Lösung Befehl: „generate“

Problem:

- Passwortlänge größer als gewünschte Länge

Lösung:

- in While-Schleife drinne, solange gewünschte Länge != Länge des generierten Passworts.
  If-Bedingungen → prüft Zeichenart gefordert & ob gewünschte Länge = Länge des generierten Passwortes

Problem/ Lösung Befehl: „generate“

Problem:

- Angegebene Länge war kleiner als die gegebenen Anforderungen

Lösung:

- If-Bedingung → prüft, dass nicht die Möglichkeit besteht, mehr Anforderungen zu wählen, als die Länge des gewünschten Passworts ist

Problem/ Lösung Befehl: „generate“

Problem:

- Zeichenarten wurden gewählt die NICHT angefordert wurden
- Die übrigen plätze wurden random aus allen Zeichenkategorien gefüllt

Lösung:

- Variable gelöscht, indem (alle Zeichen gespeichert sind)

Passwortsicherheitsstufen

Probleme – Lösungen

```
def passwortsicherheit(passwort):
    staerke = ""
    #prüfen welche sicherheitsstufe erreicht wurde, je nach Inhalt des Passwortes
    if check_grossbuchstaben(passwort) or check_kleinbuchstaben(passwort):
        staerke += "*"
    if check_zahlen(passwort):
        staerke += "*"
    if check_sonderzeichen(passwort):
        staerke += "*"

    if staerke == "*":
            print("Passwortstärke: "+ Fore.LIGHTRED_EX + staerke + Fore.RESET)

    elif staerke == "**":
            print("Passwortstärke: "+ Fore.LIGHTYELLOW_EX + staerke + Fore.RESET)

    elif staerke == "***":
            print("Passwortstärke: "+ Fore.LIGHTGREEN_EX + staerke + Fore.RESET)
```

```
def check_kleinbuchstaben(passwort):
    for b in passwort:
        if b in string.ascii_lowercase:
            return True
    return False

def check_grossbuchstaben(passwort):
    for b in passwort:
        if b in string.ascii_uppercase:
            return True
    return False

def check_zahlen(passwort):
    for b in passwort:
        if b in string.digits:
            return True
    return False

def check_sonderzeichen(passwort):
    for b in passwort:
        if b not in string.digits and b not in string.ascii_lowercase and  b not in string.ascii_uppercase:
            return True
    return False
```

Passwortsicherheit

Problem:

- Passwörter sollen nicht sichtbar sein

Lösung:

- Einen Hard codierten Schlüssel für den „Ent-Verschlüssler“

Passwortsicherheit

Problem:

- Open-Source code

Lösung:

- Masterpasswort als Schlüssel für den „Ent-Verschlüssler“

Passwortsicherheitsstufen

Erstmal ohne Variable “staerke” versucht, sondern mit print-Anweisungen

Problem:

- es wurde nie die richtige Stäerke ausgegeben.

Lösung:

- mit Variable “staerke” und einzelnen if-Abfragen, so wird jede Bedingung geprüft

Passwortsicherheitsstufen

Erstmal alle Abfragen in der Methode passwortsicherheit eingefügt.

Problem:

- zu unüberscihtlich, es ist nicht in jeden zweig reingegangen →Fehler schleichen sich ein

Lösung:

- Für jede Bedingung eigene Methode definieren und nur die Methode in passwortsicherheit() aufrufen

Passwortsicherheitsstufen

Erstmal eigene Variablen deklariert, die die jeweiligen Zeichen enthalten.

Problem:

- Zu unübersichtlich, es kann etwas vergessen werden.

Lösung:

- Asciii-Tabelle, beinhaltet alle Zeichen, die man braucht. Sonderzeichen umgehen → nicht Zahlen, nicht Buchstaben