

Übungsblatt 05

Aufgabe 1

Bei einem Marathonlauf müssen am Zielpunkt die Ankunftszeiten der Läufer registriert werden. Schreiben Sie ein interaktives Python-Programm, das Folgendes leistet:

- Immer wenn ein neuer Läufer das Ziel erreicht, wird seine Startnummer von Ihnen eingegeben
- Nach Eingabe einer Startnummer wird die Ankunftszeit über die Systemuhr ermittelt
- Startnummer und Ankunftszeit werden in einer Textdatei mit dem Pfad `marathon/daten.txt` zeilenweise abgespeichert
- Das Programm wird beendet, indem Sie ohne Eingabe auf *Enter* drücken
- Damit im Falle eines Systemabsturzes keine Daten verloren gehen, soll nach jeder Eingabe eine Zwischenspeicherung erfolgen

Hinweis zur Systemzeit: mit folgender Anweisungsfolge erhalten Sie in `zeit` die aktuelle Systemzeit als `string`:

```
import time
zeit = time.asctime()
```

In []: `# TODO Aufgabe 1`

Aufgabe 2

Gegeben ist eine CSV-Datei, die das Ergebnis der Analyse einer Metallprobe darstellen soll. Jeder Eintrag enthält den Namen eines Edelmetalls und dessen Gehalt in Prozent:

Edelmetall	Gehalt in Prozent
Gold	0,1234
Silber	23,45
Platin	0,0678

Schreiben Sie ein Skript, das den Inhalt der CSV-Datei auf dem Bildschirm tabellarisch in folgenden Formaten ausgibt:

```
Gold 0.1234 // Silber 23.45 // Platin 0.0678 //
```

```
Gold    0.1234
Silber  23.45
Platin  0.0678
```

Gold: 0.12

Silber: 23.45

Platin: 0.07

Nutzen Sie dafür die in den Vorlesungen behandelten Möglichkeiten von `print()` und `f-String`.

Achtung: Das Dezimaltrennzeichen in Excel ist ein "," und in Python ein "."

Tipp: Nutzen Sie z.B die `string`-Funktion: `str.replace()`

Welche Darstellungsform halten Sie für die beste?

In [12]: `# TODO Aufgabe 2`

Aufgabe 3

In folgender Aufgabe soll ein Verwaltungsprogramm für Telefonnummern entwickeln werden. Die Besonderheit soll hierbei das *User-Interface* sein, denn nach jeder Aktion soll auf dem Bildschirm ein Menü erscheinen, aus dem der Nutzer eine von folgenden Funktionen auswählen kann:

- Suche nach Telefonnummer: Der Benutzer gibt über die Tastatur einen Namen ein und erhält als Antwort die Telefonnummer, falls sie gespeichert ist. Ansonsten gibt es die Meldung `Name unbekannt`.
- Neue Nummer eintragen: Der Benutzer wird nach einem Namen und der zugehörigen Telefonnummer gefragt.
- Alle Nummern ausgeben: Auf dem Bildschirm erscheint eine gut lesbare Tabelle mit allen gespeicherten Namen und Telefonnummern
- Ende: Das aktualisierte Telefonbuch wird gespeichert und das Programm beendet

Hinweise zur Programmierung:

- Verfahren Sie nach dem Prinzip der schrittweisen Verfeinerung und definieren Sie Funktionen.
- Beim Auswählen der Funktion soll nicht zwischen Groß- und Kleinschreibung unterschieden werden.
- Verwenden Sie ein `dictionary` mit dem Namen als Schlüssel und den Telefonnummern als Werten.
- Beim Start des Programms wird das `dictionary` mithilfe einer `pickle`-Funktion geladen und vor Beendigung wieder gespeichert. Der Pfad soll sein `verwaltung/telefonbuch.pkl`

Wechseln Sie zum Bearbeiten in das Python-File `aufg3.py`. Zum Ausführen klicken Sie in VS Code rechts oben auf den Dropdown-Pfeil neben dem `Run`-Button und wählen Sie

Run Python File in Dedicated Terminal, um unten im Terminal Eingaben und Ausgaben zu sehen

Beispieldialog:

(S)uche nach Telefonnummer
(N)eue Nummer eintragen
(A)lle Nummern ausgeben
(E)nde
Ihre Wahl: n
Name: Max
Nummer: 89923
Neuer Eintrag gespeichert

(S)uche nach Telefonnummer
(N)eue Nummer eintragen
(A)lle Nummern ausgeben
(E)nde
Ihre Wahl: s
Name: Tim
Nummer: 85675

(S)uche nach Telefonnummer
(N)eue Nummer eintragen
(A)lle Nummern ausgeben
(E)nde
Ihre Wahl: a
Name Nummer

Tim	85675
Jenny	233325
Max	89923

(S)uche nach Telefonnummer
(N)eue Nummer eintragen
(A)lle Nummern ausgeben
(E)nde
Ihre Wahl: e
Danke, dass Sie dieses Produkt verwendet haben.
Beenden mit <ENTER>