
A Aufgabenstellung

Lehrer müssen oft Noten- und Schülerlisten in verschiedenen Formaten zusammenführen. Entwickle ein kleines Hilfsprogramm für die Kommandozeile (CLI), die einem Lehrer beim Zusammenführen einer Noten- und Schülerliste unterstützt.

Wir verwenden dazu *Regular Expressions*¹ zum Einlesen einer .xml-Datei für die Stammdaten der Schüler und üben wieder den Umgang mit . Die mächtige Bibliothek² hilft uns dabei,

- die .csv-Datei mit den Schülernoten einzulesen,
- die Notenliste und die Schülerliste zu verknüpfen,
- das Ergebnis zu filtern
- und wieder in eine csv-Datei zu schreiben.

B Allgemeine Hinweise:

- Benutze die Python-Bibliothek `argparse`– siehe Foliensatz / Vorbesprechung
- Überprüfe immer vor dem Einlesen einer Datei, ob diese auch existiert³
- Fehlermeldungen sollen– wie unter Linux üblich– auf dem `Stderror-Device` ausgegeben werden⁴. Benutze standardisierte Fehlermeldungen⁵
- Benutze die beiliegenden Dateien, um die Lösung zu überprüfen
- Alle Beispiele kommen in das Git Repository (2024_4cn_xxxx_sew4_sem1p xxxx=4-stellige Nummer), Dateiname = .py.
- Type-Annotations = Type-Hints verwenden⁶
- Dokumentation mit PyDoc
- “Hauptprogramm” mit `if __name__ == "__main__":`

C Beispielaufufe

Die Beispielaufufe führen zu den beigefügten Dateien.

C.1 Hilfe

```
python noten.py -h
```

```
usage: noten.py [-h] [-n N] [-s S] [-m M] [-f F] [-v | -q] outfile
```

noten.py by Max Mustermann / HTL Rennweg

positional arguments:

outfile Ausgabedatei (z.B. result.csv)

options:

```
-h, --help       show this help message and exit
-n N             csv-Datei mit den Noten
-s S             xml-Datei mit den Schülerdaten
-m M             AName der Spalte, die zu verknüpfen ist (default = Nummer)
-f F             Name des zu filternden Gegenstandes (z.B. SEW)
-v, --verbose    Gibt die Daten Kommandozeile aus
```

¹`import re`

²https://pandas.pydata.org/docs/user_guide/

³<https://www.w3docs.com/snippets/python/how-do-i-print-to-stderr-in-python.html>

⁴<https://www.geeksforgeeks.org/python-os-strerror-method/>

⁵<https://www.geeksforgeeks.org/python-os-strerror-method/>

⁶siehe z.B. https://mypy.readthedocs.io/en/stable/cheat_sheet_py3.html

`-q, --quiet` keine Textausgabe

keine Textausgabe

C.2 Ohne Filter, verbose

Verknüpft beide Dateien. Man erhält eine .csv-Datei mit den relevanten Daten aller SchülerInnen und sämtlichen Gegenständen.

```
python noten.py -n noten.csv -s schueler.xml -v result_all.csv
csv-Datei mit den Noten : noten.csv
xml-Datei mit den Schülerdaten : schueler.xml
Name der Spalte, die zu verknüpfen ist : Nummer
Output-Datei: result_all.csv
```

C.3 Mit Filter auf SEW und ITP, weder verbose noch quiet

Hier erhält man als Ergebnis nur die Noten aus den angegebenen Gegenstand.

Bonus: Man kann mehrere Gegenstände angeben; das Programm bildet eine zusätzliche Spalte *Schnitt*, um die Durchschnittsnote der ausgewählten Gegenstände zu speichern.

```
python noten.py -n noten.csv -s schueler.xml -f SEW,ITP result_SEW.csv
Output-Datei: result_SEW.csv
```

D Hinweise und Tipps

- Die .xml-Datei soll nicht mittels Pandas-Methode eingelesen werden, sondern mit der zu implementierenden Methode `read_xml(filename: str) -> pd.DataFrame`. Lies die gesamte Datei ein – z.B. mit `content = f.read()`. Finde mittels Regular Expression alle Treffer für *Nummer*, *Anrede*, *Vorname*, *Nachname*, *Geburtsdatum* und speichere sie in einem Pandas-DataFrame.

```
pattern = re.compile("regular expression", flags=re.DOTALL)
result = re.findall(pattern, content)
df = pd.DataFrame(result, columns=[.....], dtype=str)
```

- Die .csv-Dateien kann man mittels Hilfsfunktion manuell parsen – oder eine geeignete Pandas-Methode zum Lesen/Schreiben verwenden.
- Zum Verknüpfen der DataFrames eignet sich die Methode `.join()`. Tipp: setze den Index der DataFrames auf die geeignete Spalte.