GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

GCDH

**Einführung in die Python-Programmierung für Geistes- und SozialwissenschaftlerInnen**

SoSe 2024

Caroline Sporleder

# Assignment 1, Hand in by: Sunday 14.7.2024, midnight

Please read the instructions entirely and carefully. Note: This assignment is ungraded ("pass" vs. "no pass", only)

Please solve the assignment on your own and don't share your solution with others. However, you may discuss general aspects of the assignment with others.

In this assignment, you will write a program to (i) read in a file with basic information on scientists, (ii) process this file and store the information in (a) suitable datastructure(s) and (iii) implement a loop which allows a user to interact with the programm and extract/query different types of biographical information.

**Data** The input file `BiographicDB.tsv` is available from StudIP (folder "00 Assignments/Assignment 1"). It is in so-called *tab separated format*, which means that it consists of lines and columns, where each pair of columns is separated by a tab(ulator). Tabs are only used in this (and no other) function in the file, i.e. you can identify the individual columns by looking for the tabs. Each line in the file contains the biographical information of one scientist. There are 40 lines (i.e. 40 scientists) altogether. Each column corresponds to a particular type of biographical information, namely:

1. first name(s)
2. surname (like the first name, this can consist of multiple 'words', e.g. *von Neumann*)
3. sex (only M and F, for simplicity)
4. year of birth
5. year of death (this can be missing if the scientist is still alive)

6. occupation (the field of expertise, e.g. *mathematician*. Note: Some scientists worked in several fields, which will all be listed in this column and separated by a semicolon: ";". An example is Marie Sklodowska-Curie, who was known as a chemist and a physicist. Occupations are restricted to *computer scientist*, *mathematician*, *physician*, *physicist*, *chemist*, and *engineer*).

**Task** The input file is effectively a very simple database and your tasks are:

1. **To read in the file and store its contents in one or more datastructures.** The datastructure(s) should be chosen in such a way that the further processing (see below) is made as easy as possible. Different solutions are possible. Remember that you learnt, among others, about the following complex data types: dictionaries, lists, sets and tuples. These can also be combined by embedding one data type inside another.

2. **To implement a loop that allows a user to interact with the program by sending different types of queries about the data.** The loop should be terminated when the users types `return` only, i.e. submits the empty string. As long as the user hasn't submitted an empty string, at each iteration the programm should output "How can I help you?" and then read the query from the command line. You do not have to check whether the user submitted query is well-formed, i.e. you can assume that the user is competent and cooperative.

3. **Your programm should be able to process the following types of queries.** Note: You can assume that the user only queries the attributes SEX, OCCUPATION, and —in a very specific context (see below)— the years in which a scientist was/is alive. The attributes FIRST NAME and SURNAME will not be queried, nor will the year of birth or the year of death be queried explicitly.

   - **Queries on <u>one</u> biographical attribute**, where the attributes are restricted to SEX and OCCUPATION. Here is a complete list of possible queries in the form in which the user will supply them:
     - `sex:f` (should return all female scientists)
     - `sex:m` (should return all male scientists)
     - `occupation:mathematician` (should return all mathematicians, independently of their gender)
     - `occupation:computer scientist`
     - `occupation:engineer`
     - `occupation:physician`
     - `occupation:physicist`
     - `occupation:chemist`

- **Queries about the intersection of two** (and only two) **biographical attributes**, e.g.
  - `sex:female AND occupation:physicist` (should return all female physicists, i.e. *Lise Meitner*, *Marie Salomea Sklodowska-Curie*, *Susan Jocelyn Bell Burnell*, *Angelita Castro Kelly*, and *Kwang Hwa Chung*)
  - `occupation:chemist AND occupation:physicist` (should return all scientists who are/were both chemists and physicists, i.e. *Marie Salomea Sklodowska-Curie*, *Sidney Harris Cox Martin* and *Michael Stefanidis*.
  - and so on for all combinations of two occupations and all combinations of sex and occupation
- **Queries about all scientists that were alive in a given year** (note this query type will not be combined with any other attribute), e.g.:
  - `alive-in:1806` (should return *Jozef Wojciechowsky* and *Isambard Kingdom Brunel*

Note: The output of each query should be a (possibly empty) list of names. Please use the complete names, i.e., all first names and the complete surname for a given person.

**Submission** Your program should be in Jupyter Notebook format. Ideally you should provide a very brief documentation of what your code does (in the form of in-code comments, describing the more complex aspects of your code) and you should also name your variables and functions in a reasonable transparent way. **Please use your surname in the title of your file, e.g.** `Schmidt.ipynb` **or** `SchmidtDBQuery.ipynb`**.** This makes it much easier for me to tell your submissions apart. **Submit your Jupyter notebook file by 14.7.2024 to the StudIP folder "00 Assignment/Assignment 1/SubmissionAssignment1".**