

Werkzeuge für das wissenschaftliche Arbeiten

Python for Machine Learning and Data Science

15. Dezember 2023

Contents

1	Projektaufgabe	2
1.1	Einleitung	2
1.2	Aufbau	2
1.3	Methoden	2
2	Diagramm	3
3	Abgabe	3

1 Projektaufgabe

In dieser Aufgabe beschäftigen wir uns mit Objektorientierung in Python. Der Fokus liegt auf der Implementierung einer Klasse, dabei nutzen wir insbesondere auch Magic Methods.

1.1 Einleitung

Ein Datensatz besteht aus mehreren Daten, ein einzelnes Datum wird durch ein Objekt der Klasse `DataSetItem` repräsentiert. Jedes Datum hat einen Namen (Zeichenkette), eine ID (Zahl) und beliebigen Inhalt.

Mehrere Daten, also Objekte vom Typ `DataSetItem`, sollen in einem Datensatz zusammengefasst werden. Es gibt eine Klasse `DataSetInterface`, die die Schnittstelle definiert und Operationen angibt, die jeder Datensatz unterstützen muss. Die Implementierung dieser Klasse fehlt jedoch noch.

1.2 Aufbau

Es gibt drei Dateien:

- `dataset.py`: Enthält die Klassen `DataSetInterface` und `DataSetItem`.
- `implementation.py`: Hier soll die Klasse `DataSet` implementiert werden.
- `main.py`: Diese Datei nutzt die Klassen `DataSet` und `DataSetItem` und testet die Schnittstelle und Operationen von `DataSetInterface`.

1.3 Methoden

Die Klasse `DataSet` muss folgende Methoden implementieren:

- `__setitem__(self, name, id_content)`: Hinzufügen eines Datums mit Name, ID und Inhalt.
- `__iadd__(self, item)`: Hinzufügen eines `DataSetItem`.
- `__delitem__(self, name)`: Löschen eines Datums anhand des Namens.
- `__contains__(self, name)`: Überprüfung, ob ein Datum mit diesem Namen im Datensatz vorhanden ist.
- `__getitem__(self, name)`: Abrufen eines Datums anhand seines Namens.
- `__and__(self, dataset)`: Schnittmenge zweier Datensätze bestimmen und als neuen Datensatz zurückgeben.
- `__or__(self, dataset)`: Vereinigung zweier Datensätze bestimmen und als neuen Datensatz zurückgeben.
- `__iter__(self)`: Iteration über alle Daten des Datensatzes.
- `filtered_iterate(self, filter)`: Gefilterte Iteration über einen Datensatz.
- `__len__(self)`: Anzahl der Daten im Datensatz abrufen.

2 Diagramm

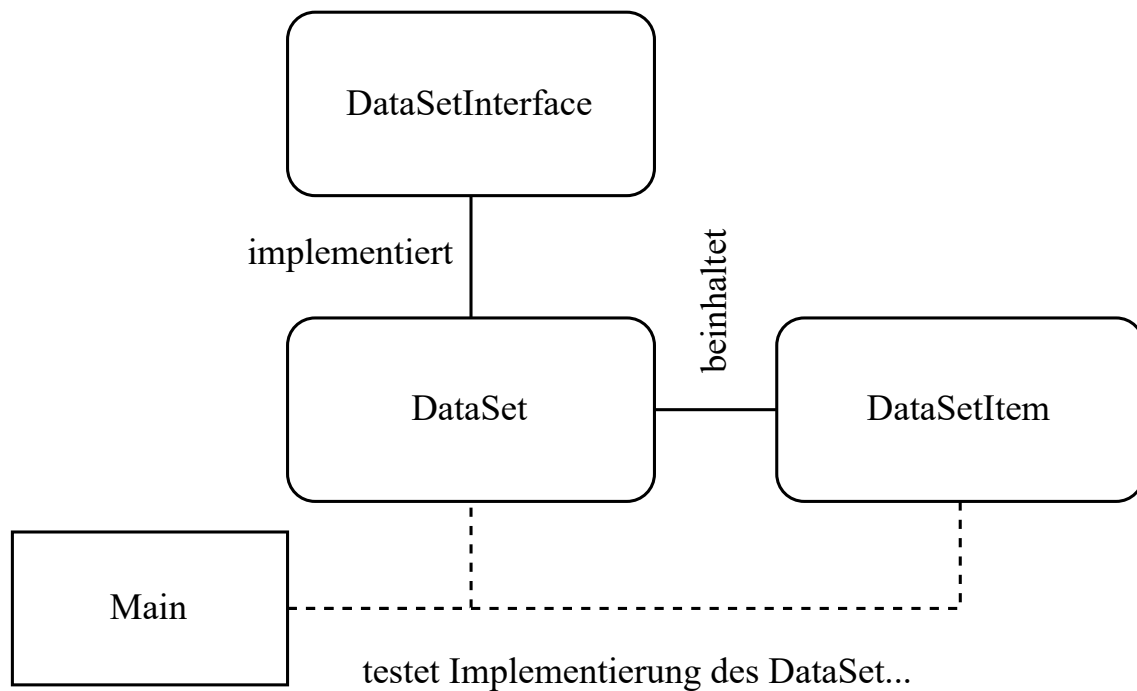


Figure 1: Darstellung der Beziehungen zwischen den Klassen `DataSetInterface`, `DataSet` und `DataSetItem`

3 Abgabe

Implementieren Sie die Klasse `DataSet` in der Datei `implementation.py` zur Lösung der oben beschriebenen Aufgabe.

Sie können auch lokal auf Ihrem Computer arbeiten. Alle drei benötigten Dateien sind im Moodle zum Download verfügbar.

Das VPL verwendet den gleichen Code, wobei `main.py` zusätzliche Testfälle und Prüfungen enthält.

Dateien befinden sich im Ordner `/code/` dieses Git-Repositories.