

Werkzeuge für das wissenschaftliche Arbeiten

Python for Machine Learning and Data Science

Abgabe: 15.12.2023

Inhaltsverzeichnis

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

1. Projektaufgabe

In dieser Aufgabe beschäftigen wir uns mit Objektorientierung in Python. Der Fokus liegt auf der Implementierung einer Klasse, wobei insbesondere Magic Methods genutzt werden.

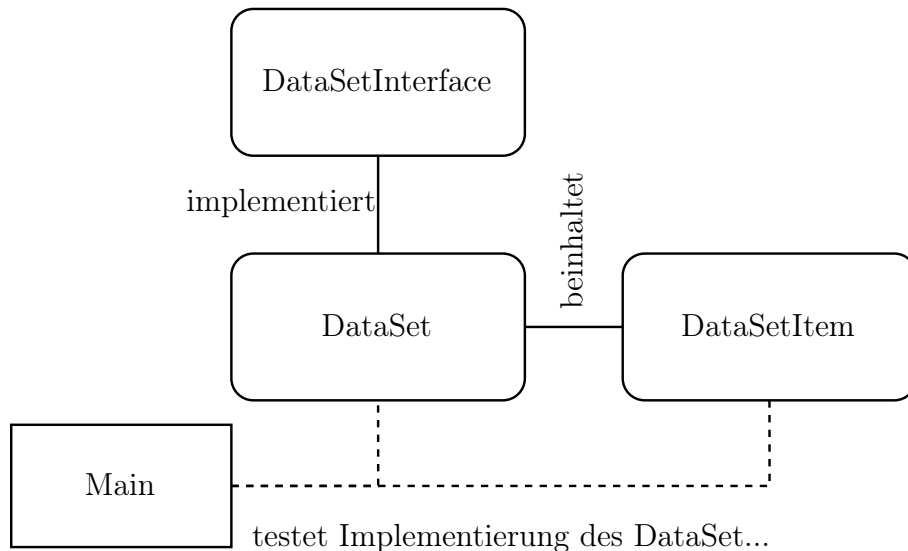


Abbildung 1: Darstellung der Klassenbeziehungen.

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, Objekte vom Typ `DataSetItem`, sollen in einem Datensatz zusammengefasst werden. Die Schnittstelle und benötigten Operationen sind bereits definiert. Die Klasse `DataSetInterface` gibt die Schnittstelle vor, während die Implementierung der Datensatz-Klasse noch aussteht.

Implementieren Sie eine Klasse `DataSet` als Unterklasse von `DataSetInterface`.

1.2. Aufbau

Die Aufgabe umfasst drei Dateien:

- `dataset.py`: Beinhaltet die Klassen `DataSetInterface` und `DataSetItem`.
- `implementation.py`: Hier wird die Klasse `DataSet` implementiert.
- `main.py`: Testet die Klassen und die Schnittstelle `DataSetInterface`.

1.3. Methoden

Die Klasse `DataSet` muss folgende Methoden implementieren (Details in `dataset.py`):

- `__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 basierend auf seinem Namen.
- `__contains__(self, name)`: Überprüfen, ob ein Datum mit diesem Namen existiert.
- `__getitem__(self, name)`: Abrufen eines Datums über seinen Namen.
- `__and__(self, dataset)`: Schnittmenge zweier Datensätze als neuen Datensatz zurückgeben.
- `__or__(self, dataset)`: Vereinigung zweier Datensätze als neuen Datensatz zurückgeben.
- `__iter__(self)`: Iteration über alle Daten im Datensatz (optional sortiert).
- `filtered_iterate(self, filter)`: Gefilterte Iteration über den Datensatz mit einer Lambda-Funktion.
- `__len__(self)`: Anzahl der Daten im Datensatz abrufen.

2. Abgabe

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

Die Datei `main.py` wird im VPL (Virtual Programming Lab) mit zusätzlichen Tests erweitert, um Ihre Lösung zu überprüfen.

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