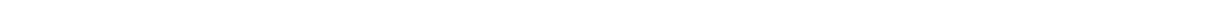


Werkzeuge für das wissenschaftliche Arbeiten

Python for Machine Learning and Data Science

Abgabe: 15.12.2023



Contents

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

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.

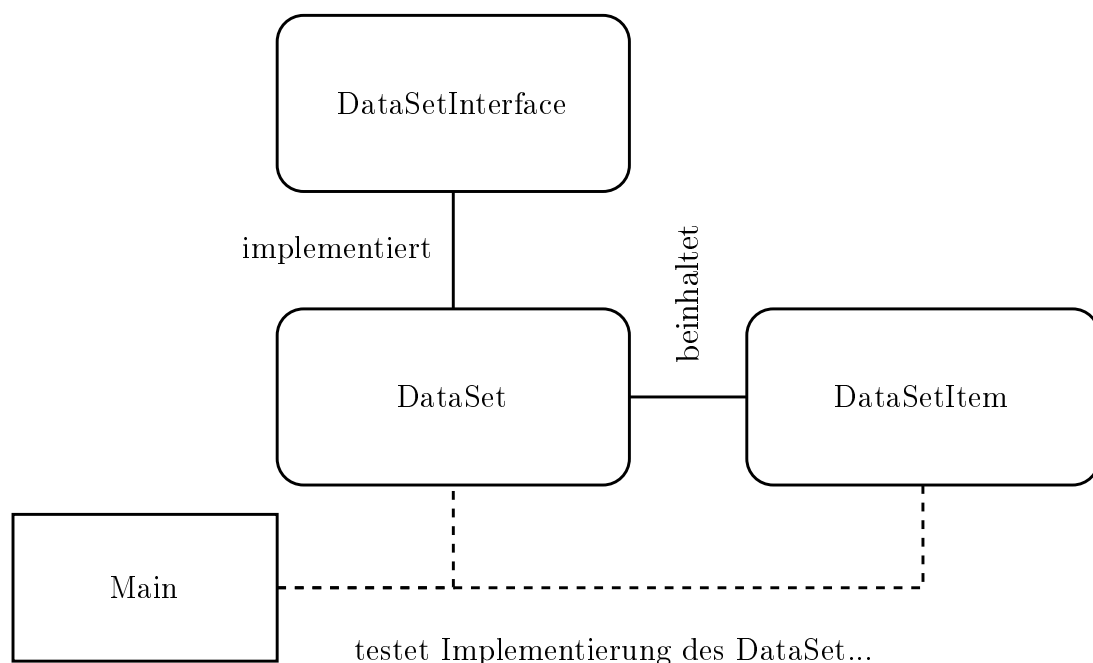


Figure 1: Darstellung der Klassenbeziehungen.

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`, werden in einem Datensatz zusammengefasst. Es existiert eine Schnittstelle `DataSetInterface`, die die Operationen eines Datensatzes definiert. Ihre Aufgabe ist es, die Klasse `DataSet` als Unterklasse von `DataSetInterface` zu implementieren.

Aufbau

Die Implementierung erfolgt in drei Dateien*:

- `dataset.py`: Enthält die Klassen `DataSetInterface` und `DataSetItem`.
- `implementation.py`: Hier erfolgt die Implementierung der Klasse `DataSet`.
- `main.py`: Testet die Klassen `DataSet` und `DataSetItem`.

Methoden

Folgende Methoden sind für die Klasse `DataSet` zu implementieren (Details finden Sie 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 anhand seines Namens (Namen sind eindeutige Schlüssel).
- `__contains__(self, name)`: Prüfung, ob ein Datum mit diesem Namen existiert.
- `__getitem__(self, name)`: Abrufen eines Datums über seinen Namen.
- `__and__(self, dataset)`: Schnittmenge zweier Datensätze berechnen und zurückgeben.
- `__or__(self, dataset)`: Vereinigungen zweier Datensätze berechnen und zurückgeben.
- `__iter__(self)`: Iteration über alle Daten im Datensatz (optional sortiert).
- `filtered_iterate(self, filter)`: Gefilterte Iteration mithilfe einer Lambda-Funktion mit Parametern Name und ID.
- `__len__(self)`: Anzahl der Daten im Datensatz abrufen.

Abgabe

Die Implementierung der Klasse `DataSet` erfolgt in der Datei `implementation.py`. Nutzen Sie das Virtual Programming Lab (VPL) oder programmieren Sie lokal, indem Sie die Dateien aus Moodle herunterladen.

Im VPL wird `main.py` mit zusätzlichen Testfällen erweitert, um sicherzustellen, dass Ihre Implementierung korrekt ist.

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