Werkzeuge für das wissenschaftliche Arbeiten

Python for Machine Learning and Data Science

Abgabe: 15.12.2023

Inhaltsverzeichnis

1 Projektaufgabe **1**

1.1 Einleitung 2

1.2 Aufbau 2

1.3 Methoden 2

2 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.

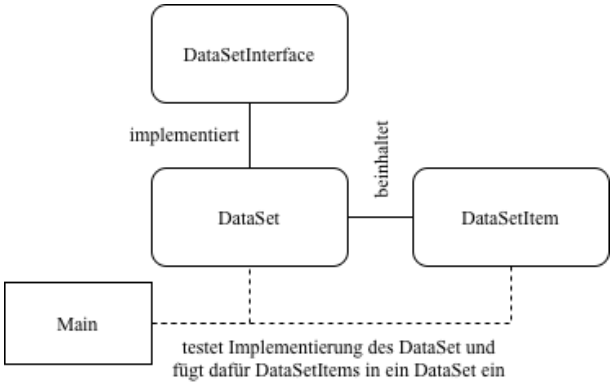

Abbildung 1: Abbildung 1.: Darstellung der Klassenbeziehungen

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 bel. Inhalt.

Nun sollen mehrere Daten, Objekte vom Typ "DataSetItem", in einem Datensatz zusammengefasst werden. Sie haben sich schon auf eine Schnittstelle und die benötigen Operationen, die ein Datensatz unterstützen muss, geeinigt. Es gibt eine Klasse "DataSetInterface", die die Schnittstelle definiert und Operationen jedes Datensatzes angibt. Bisher fehlt aber noch die Implementierung eines Datensatzes mit allen Operationen.

Implementieren Sie eine Klasse "DataSetäls eine Unterklasse von "DataSetInterface".

1.2 Aufbau

Es gibt drei Dateien, "dataset.py", "main.pyünd ïmplementation.py"*. In der "dataset.py"befinden sich die Klassen "DataSetInterfaceünd "DataSetItem", in der Datei ïmplementation.py"muss die Klasse "DataSetïmplementiert werden. Die Datei "main.py"nutzt die Klassen "DataSetünd "DataSetItemäus den jeweiligen Dateien und testet die Schnittstelle und Operationen von "DataSetInterface".

1.3 Methoden

Bei der Klasse "DataSetßind insbesondere folgenden Methoden zu implementieren, die genaue Spezifikation finden Sie in der "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 auf Basis des Namens.
  Das heißt auch, dass der Name eines Datums ein eindeutiger Schlüssel ist und jeder Name nur einmal pro Datensatz vorkommen kann.

- ___contains___(self, name)
  Prüfung, ob ein Datum mit diesem Namen im Datensatz vorhanden ist.

- ___getitem___(self, name)
  Abrufen des Datums über seinen Namen.

- ___and___(self, dataset)
  Schnittmenge zweier Datensätze bestimmen und als einen neuen Datensatz zurückgeben.

- ___or___(self, dataset)
  Vereinigungen zweier Datensätze bestimmen und als einen neuen Datensatz zurückgeben.

- ___iter___(self)
  Iteration über alle Daten den Datensatzes (es ist möglich eine Sortierung für die Reihenfolge der Iteration anzugeben).

- filtered_iterate(self, filter)
  Gefilterte Iteration über einen Datensatz, dabei dient eine Lambda-Funktion mit den Parametern Name und ID für jedes Datum als Filter.

- ___len___(self)
  Anzahl der Daten in einem Datensatz abrufen.

2 Abgabe

Programmieren Sie die Klasse "DataSetin der Datei implementation.pyßur Lösung der oben beschrieben Aufgabe im VPL. Sie können auch direkt auf Ihrem Computer programmieren, dazu finden Sie alle drei benötigten Dateien zum Download im Moodle.

Das VPL nutzt den gleichen Code, wobei die "main.pyüm weitere Testfälle und Überprüfungen erweitert wurde. Die Überprüfungen dienen dazu sicherzustellen, dass Sie die richtigen Klassen nutzen.

---

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