



# Modulararbeit M165

PYMONGO

ZAIN JILANI KHAN

## Inhaltsverzeichnis

<b>Lernjournal</b> .....	<b>1</b>
<b>Gesamtreflexion</b> .....	<b>4</b>
Was habe ich gelernt? .....	5
Arbeitsweise .....	6
Leistung/Resultat .....	6
Bilanz .....	6

Datum	Was habe ich getan?	Probleme/ Schwierigkeiten	Zeitaufwand
25.09.2023	Heute haben wir den Auftrag bekommen eine Modularbeit, im Modul 165 zu machen. Das Modul beinhaltet das Arbeiten mit No-SQL (Not only SQL). Wir haben in den vergangenen Wochen, mit dem No-SQL Datenbankmanagementsystem MongoDB gearbeitet, bei dem wir gelernt haben verschiedene Operationen mit dem System auszuführen. Nun neigt sich das Modul 165 dem Ende zu und zum Abschluss machen wir diese Modularbeit. MongoDB arbeitet mit Treibern, um die Schnittstellen zu anderen Programmiersprachen zu gewährleisten. In unserem Fall müssen wir, MongoDB mit Python verbinden. Diese Verbindung erfolgt mit dem Treiber PyMongo. Heute habe ich erstmal bisschen über PyMongo recherchiert und versucht es herunterzuladen, erstmals habe ich die Eingabeaufforderung geöffnet, danach habe ich es mit MongoDB verbunden, indem ich mongosh eingegeben habe.	Sobald ich mongosh eingegeben habe, hat es einen Fehler angezeigt, dass es nicht verbunden werden kann. Also habe ich mein Laptop einfach neugestartet. Nachdem aufstarten habe ich wieder mongosh eingegeben, dann ging es zwar aber die Lektion war um.	10 Min.
5.10.2023	Heute habe ich mich zum ersten Mal richtig mit der Modularbeit befasst. Ich habe mich als erstes auf der MongoDB Webseite angemeldet und das Tutorial von Pymongo angefangen. Ich habe ein bis zwei Videos geschaut. Ich habe nur bisschen geschaut, weil es schon 3:00 war.	Ich hatte keine Probleme	10 Min
06.10.2023	Heute habe ich das Tutorial weiter angeschaut und schon mal Pymongo via Cmd installiert. Danach habe ich einen Ordner namens python-test erstellt, eine Virtual Environment von Python eingerichtet und eine Python-File reingetan.	Die Schritte waren nicht einzeln beschrieben, was dazu führte, dass ich viel Recherchieren musste.	6-7 Min
07.10.2023	Das Tutorial hat mir wenig über die einzelnen Schritte gesagt, deswegen habe ich auf YouTube noch Videos angeschaut, wie man eine Virtual Environment verwendet. Im Tutorial hat es auch geheissen, dass ich ein Python Programm in der Virtual Environment erstellen muss. Also tat ich dies und nannte es connection.py	Ich musste viel recherchieren und Viele Videos anschauen. Sie haben teilweise Sachen nicht so erklärt wie ich es brauchte. Die Videos waren nicht	10-15 Min
13.10.2023	Heute habe ich gemerkt, dass ich einiges falsch gemacht habe, und zwar habe ich etwas im Unterricht falsch verstanden. Ich habe verstanden, dass man alles in CMD machen muss, obwohl eigentlich nur das Programm dort laufen muss. Ich konnte die Videos nicht schauen, weil es die ganze Zeit Netzwerkprobleme gab. Also musste ich es heute aufgeben.	All das sorgte für Verwirrung. Ich nahm mir vor alles vor den Ferien zu machen, was eigentlich möglich wäre, wenn ich alles richtig gemacht hätte. Ich musste	20 Min.

		jetzt alles in Visual Code machen. Da ich im Zug arbeiten musste, gab es Internetprobleme, weil der Zug, von Stadt zu Stadt fuhr und es deswegen Probleme mit dem Netzwerk gab.	
23.10.2023	<p>Heute habe ich weitergearbeitet und schlussendlich, dass Programm erstellt. Ich habe dafür die connection.py Datei, in Visual Studio aufgemacht, danach habe ich im Terminal von Visual Studio, wieder Pymongo installiert, indem ich pip install pymongo eingegeben habe. Als es das Installiert hat habe ich PyMongo importiert.</p> <p>Ich habe dann noch eine Datenbank namens Spiele und eine collection namens</p> <pre>from pymongo import MongoClient</pre> <p>pcgames erstellt. Um dann Das Programm mit der Datenbank zu verbinden habe ich den Connection String mit dem MongoDB Client verbunden und anschliessend auch die Datenbank und Collection.</p> <pre>connection_string = "mongodb://localhost:27017" client = MongoClient(connection_string) gamedb = client.spiele pcgames_coll = gamedb.pcgames</pre>	Ich musste bisschen recherchieren, wie man das verbindet.	10-15 Min
26.10.2023	<p>Heute habe ich dann angefangen die Datenbank mit den Spielen zu füllen. Das tat ich, indem ich alle Spiele in einer Liste nahm, die ich games nannte. Dann gab ich alle Daten ein. Um die Spiele dann einzufügen, gab ich folgenden Befehl ein:</p> <pre>inserts = pcgames_coll.insert_many(games) game_ids = inserts.inserted_ids print("# of documents inserted: " + str(len(game_ids))) print(f"ids of inserted games: {game_ids}")</pre>	Ich konnte die Spiele nicht hinzufügen, weil ich zuerst den Connection String von der MongoDB Webseite genommen habe. Danach habe ich ein Video geschaut, wo jemand diesen Connection String von vorher eingegeben hat, dann ging es.	20 Min

28.10.2023	Heute habe ich angefangen das Programm richtig zu gestalten. Ich habe also ein neues Python-File erstellt, es wieder verbunden und dann erstmal versucht einzurichten, dass man die Spiele Suchen kann.	Ich hatte Probleme mit den ganzen if und else um das spiel zu finden. Sobald ich ein Fehler korrigierte wusste das Programm einen anderen Fehler auf.	30 Min
29.10.2023	Heute habe ich weiter gemacht, was ich gestern gemacht habe. Ich habe mir das Tutorial mehrmals angeschaut, weil ich es immer noch nicht hinbekommen habe. Dann endlich habe ich es hingekriegt.	Es gab immer Fehler im Code die erst nachdem ausführen des Programms erschienen. Deswegen verlor ich ganz viel Zeit	1 Stunde
30.10.2023	Heute wäre eigentlich die Abgabe, sie wurde jedoch verschoben. Dass war sehr wichtig, weil ich wieder etwas falsch verstanden hatte. Eigentlich hätte ich am Lernjournal weitergearbeitet, aber ich musste noch im Programm einfügen, dass man Spiele löschen, einfügen und bearbeiten kann. Also habe ich das eingefügt.	Es hat mich sehr gestresst. Ich musste was komplett neues einbauen. Ich musste, suchen wie man das macht. Das hat mir einiges an Zeit gekostet.	3 Lektionen

05.10.2023

Heute ist die Abgabe. Ich habe heute die Funktionen eingefügt, um die Spiele hinzuzufügen:

```
!Spiele in der Datenbank einfügen

def neues_Spiel_einfuegen():
    neues_Spiel = {
        "Titel": input("Geben Sie den Titel des Spiels ein, welches Sie einfügen möchten:"),
        "Ausgabejahr": None,
        "Downloadzahlen": None,
        "Altersgrenze": None,
        "Art": None,
        "Wertung von 1-10": None
    }

    ausgabejahr_input = input("Geben Sie das Ausgabejahr des Spiels ein:")
    while True:
        try:
            neues_Spiel["Ausgabejahr"] = int(ausgabejahr_input)
            break
        except ValueError:
            print("Ungültige Eingabe, bitte geben Sie eine ganze Zahl ein.")
            ausgabejahr_input = input("Geben Sie erneut das Ausgabejahr des Spiels ein:")

    downloadzahlen_input = input("Geben Sie die Downloadzahlen des Spiels ein:")
    while True:
        try:
            neues_Spiel["Downloadzahlen"] = int(downloadzahlen_input)
            break
        except ValueError:
            print("Ungültige Eingabe, bitte geben Sie eine ganze Zahl ein.")
            downloadzahlen_input = input("Geben Sie erneut die Downloadzahlen des Spiels ein:")

    altersgrenze_input = input("Geben Sie die Altersgrenze des Spiels ein:")
    while True:
        try:
            neues_Spiel["Altersgrenze"] = int(altersgrenze_input)
            break
        except ValueError:
            print("Ungültige Eingabe, bitte geben Sie eine ganze Zahl ein.")
            altersgrenze_input = input("Geben Sie erneut die Altersgrenze des Spiels ein:")

    neues_Spiel["Art"] = input("Geben Sie die Art des Spiels ein:").split(',')
    while True:
        wertung_input = input("Geben Sie die Wertung von 1-10 ein:")
        try:
            neues_Spiel["Wertung von 1-10"] = int(wertung_input)
            break
        except ValueError:
            print("Ungültige Eingabe, bitte geben Sie eine ganze Zahl zwischen 1 und 10 ein.")

    Eingabedoc = pcgames_coll.insert_one(neues_Spiel)

    if Eingabedoc.acknowledged:
        print("Das Spiel wurde erfolgreich hinzugefügt, die Id lautet:", Eingabedoc.inserted_id)
    else:
        print("Es gab einen Fehler beim Hinzufügen des Spiels.")
```

Dann um sie zu bearbeiten bzw. zu ändern:

```
def Spiele_in_der_datenbank_aendern():
    spiel_Titel = input("Geben Sie den Spielnamen ein, bei dem Sie etwas ändern möchten:")
    existenz_Spiel = pcgames_coll.find_one({"Titel": spiel_Titel})

    if existenz_Spiel:
        print("Spiel gefunden, Die aktuellen Informationen:")
        print(existenz_Spiel)

        Spiel_editiert = {
            "Titel": input("Geben Sie den neuen Titel ein, oder drücken Sie Enter, um nichts zu ändern:") or existenz_Spiel["Titel"],
            "Ausgabejahr": int(input("Geben Sie das neue Ausgabejahr ein, oder drücken Sie Enter, um nichts zu ändern:") or existenz_Spiel["Ausgabejahr"]),
            "Downloadzahlen": int(input("Geben Sie die neuen Downloadzahlen ein, oder drücken Sie Enter, um nichts zu ändern:") or existenz_Spiel["Downloadzahlen"]),
            "Altersgrenze": int(input("Geben Sie die neue Altersgrenze ein, oder drücken Sie Enter, um nichts zu ändern machen:") or existenz_Spiel["Altersgrenze"]),
            "Art": input("Geben Sie die neue Art des Spiels ein, oder drücken Sie Enter, um nichts zu ändern:") or existenz_Spiel["Art"],
            "Wertung von 1-10": int(input("Geben Sie die neue Wertung von 1-10 ein, oder drücken Sie Enter, um nichts zu ändern:") or existenz_Spiel["Wertung von 1-10"])
        }

        pcgames_coll.update_one({"Titel": spiel_Titel}, {"$set": Spiel_editiert})
        print("Die Spielinformationen wurden aktualisiert.")
    else:
        print("Die Spielinformationen konnten nicht aktualisiert werden")
```

Um die Spiele zu Löschen:

```
def Spiele_in_der_datenbank_loeschen():
    spiel_Titel1 = input("Geben Sie ein Spieltitle ein:")
    spiel_existenz = pcgames_coll.find_one({"Titel": spiel_Titel1})

    if spiel_existenz:
        print("Spiel existiert")
        print(spiel_existenz)

        bestaetigung = input("Möchten Sie das Spiel wirklich Löschen? (ja/nein)").lower()

        if bestaetigung == "ja":
            pcgames_coll.delete_one({"Titel": spiel_Titel1})
            print("Das Spiel wurde gelöscht")
        elif bestaetigung == "nein":
            print("Das Löschen des Spiels wurde abgebrochen")
        else:
            print("Ungültige Eingabe. Bitte geben Sie 'ja' oder 'nein' ein")
    else:
        print("Spiel nicht gefunden")
```

Es gab immer wieder Fehler im code. Sobald ich die korrigierte gab es einen Fehler bei einer anderen Funktion und immer so weiter. Es hat nicht direkt alle Fehler direkt angezeigt

Und Um sie aufzurufen:

```
#Spiele in der Datenbank aufrufen

def spiele_finden():
    Ausgabejahr = int(input("Geben Sie das Ausgabejahr ein:"))
    Downloadzahlen = int(input("Geben Sie die Downloadzahlen ein:"))
    Altersgrenze = int(input("Geben Sie die Altersgrenze ein:"))
    Art = input("Geben Sie die Art des Spiels ein (kommagetrennt:").split(',')
    Wertung = int(input("Geben Sie die Wertung von 1-10 ein:"))

    Artset = re.compile("|".join(re.escape(a.strip()) for a in Art), re.IGNORECASE)

    Kriterien = {
        "Ausgabejahr": {"$eq": Ausgabejahr},
        "Downloadzahlen": {"$lte": Downloadzahlen},
        "Altersgrenze": {"$eq": Altersgrenze},
        "Art": {"$regex": Artset},
        "Wertung von 1-10": {"$eq": Wertung}
    }

    anzahl_spiele = pcgames_coll.count_documents(Kriterien)

    if anzahl_spiele == 0:
        print("Es gibt leider kein Spiel, das den Kriterien entspricht.")
    else:
        spiele = pcgames_coll.find(Kriterien)
        for ergebnis in spiele:
            print("_id", ergebnis["_id"])
            print("Titel", ergebnis["Titel"])
            print("Ausgabejahr", ergebnis["Ausgabejahr"])
            print("Downloadzahlen", ergebnis["Downloadzahlen"])
            print("Altersgrenze", ergebnis["Altersgrenze"])
            print("Art", ergebnis["Art"])
            print("Wertung von 1-10", ergebnis["Wertung von 1-10"])
```

## Gesamtreflexion

### **Was habe ich gelernt?**

Ich habe gelernt, wie man sich neues Wissen aneignet bezüglich Programmiersprachen. Es war das erste Mal, dass ich eine Arbeit ohne grosse Erklärung bekam. Ich konnte mit dieser Arbeit mein Durchhaltevermögen stärken, daraus lernen, wie man mit Fehlern im Code umgeht und sie richtig behebt.

Positiv fand ich, dass wir mit Python gearbeitet haben, welche meine Lieblingsprogrammiersprache ist. Sie ist recht einfach zu verstehen und man kommt schnell rein.

Mir gefiel nicht, dass man auch noch die CRUD-Operationen einfügen musste, weil für mich das am aufwändigsten war.

### **Arbeitsweise:**

Ich habe nicht sehr konzentriert gearbeitet, was dann schlussendlich dazu führte, dass ich bis zur Abgabe dran war. Ich war immer wieder abgelenkt und ich habe immer bisschen gebraucht wieder reinzukommen.

### **Leistung/Resultat:**

Es ist trotz meiner nicht so guter Leistung recht gut geworden. Mit dem Resultat bin ich sehr zufrieden und es hat mir Freude bereitet, daran zu Arbeiten.

### **Bilanz:**

Im Großen und Ganzen war das ein Erfolg für mich, ich habe vieles dazu gelernt. Ich konnte ausserdem auch meine Arbeitsfähigkeiten stärken und ich würde gerne nochmal eine Arbeit in der Art machen.