9. Datenbank III

9.1. Datenbank erstellen

```
# Einbinden DB-Modul
import mysql.connector

# Verbindung zum Datenbank-Server herstellen
try:
    conn = mysql.connector.connect(
        user="root",
        host="localhost"
    )
except mysql.connector.Errors as e:
    print(f"Error connecting to DB Platform: {e}")
    exit(1)

# Cursor-Objekt erstellen
cur = conn.cursor()

# neue Datenbank erzeugen
try:
    cur.execute("CREATE DATABASE neueDB")
except mysql.connector.Errors as e:
    print(f"Error executing SQL-stmt: {e}")
    exit(1)

# alle Datenbanken anzeigen
cur.execute("SHOW DATABASES")
databaseList = cur.fetchall()

for database in databaseList:
    print(database)

# Verbindung zum Datenbank-Server schließen
conn.close()
```

9.2. Tabelle erstellen

```
# Einbinden DB-Modul
import mysql.connector

# Verbindung zum Datenbank-Server herstellen
try:
    conn = mysql.connector.connect(
        user="root",
        host="localhost",
        database="neueDB"

    )
except mysql.connector.Errors as e:
    print(f"Error connecting to DB Platform: {e}")
    exit(1)

# Cursor-Objekt erstellen
try:
    cur = conn.cursor()
except mysql.connector.Errors as e:
    print(f"Error creating DB-Cursor object: {e}")
    exit(1)

# neue Tabelle erzeugen
sql_create_table = "CREATE TABLE klasse ( \"
                   " id INTEGER PRIMARY KEY AUTO_INCREMENT," \
                   " name varchar(10) NOT NULL," \
                   " klassenlehrer_id INTEGER NOT NULL )"
try:
    cur.execute(sql_create_table)
except mysql.connector.Errors as e:
    print(f"Error executing SQL-stmt {sql_create_table}: {e}")
    exit(1)

# Verbindung zum Datenbank-Server schließen
conn.close()
```

9.3. Einträge in Tabelle einfügen

```
# Einbinden DB-Modul
import mysql.connector

# Verbindung zum Datenbank-Server herstellen
try:
    conn = mysql.connector.connect(
        user="root",
        host="localhost",
        database="neueDB"
    )
except mysql.connector.Errors as e:
    print(f"Error connecting to DB Platform: {e}")
    exit(1)

# Cursor-Objekt erstellen
try:
    cur = conn.cursor()
except mariadb.Error as e:
    print(f"Error creating DB-Cursor object: {e}")
    exit(1)

# Datensätze einfügen
sql_insert = ["INSERT INTO klasse (name, klassenlehrer_id) " \
                    "VALUES ('8A', 2)",
              "INSERT INTO klasse (name, klassenlehrer_id) " \
                    "VALUES ('8B', 4)",
              "INSERT INTO klasse (name, klassenlehrer_id) " \
                    "VALUES ('8C', 1)",
              "INSERT INTO klasse (name, klassenlehrer_id) " \
                    "VALUES ('8D', 0)"]
for i in range(0,len(sql_insert)):
    try:
        cur.execute(sql_insert[i])
    except mysql.connector.Errors as e:
        print(f"Error executing SQL-stmt {sql_insert[i]}: {e}")
        exit(1)

# DB-Änderungen "festschreiben"
try:
    conn.commit()
except mysql.connector.Errors as e:
    print(f"Error DB commit: {e}")
    exit(1)

# Verbindung zum Datenbank-Server schließen
conn.close()
```

Aufgaben

1. Datenbankschnittstelle
   die Programmteile
   - für den Verbindungsaufbau zum Datenbank-Server
   - für die Definition des Cursor-Objekts
   - für die Ausführung von SQL-Anweisungen

   sind inklusive der Fehlerbehandlung als Funktionen zu definieren und in ein Python-Module auszulagern.
   Zur Ausführung von Datenbankzugriffen sollen dann nur noch diese Module verwendet finden.