In [ ]:

```
from pymongo import MongoClient
from .database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self, host, collection_name):
        ...

    def connect(self):
        ...

    def disconnect(self):
        ...

    def insert_data(self, query):
        ...

    def delete_data(self, query):
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...



"""_____"""


import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self, host, collection_name):
        self.host = host
        self.collection_name = collection_name
        self.client = None
        self.db = None
        self.collection = None

    def connect(self):
        self.client = pymongo.MongoClient(host=self.host)

    def disconnect(self):
        ...

    def insert_data(self, query):
        ...

    def delete_data(self, query):
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...



"""_____"""


import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
```

```
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]

    def disconnect(self):
        ...

    def insert_data(self, query):
        ...

    def delete_data(self, query):
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...



""" ____________________________________________________________________ """



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        ...

    def insert_data(self, query):
        ...

    def delete_data(self, query):
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...



""" ____________________________________________________________________ """



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
```

```
                database_name: str,
                collection_name: str,
                user: str = None,
                password: str = None,):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        ...

    def insert_data(self, query):
        ...

    def delete_data(self, query):
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...


"""_____"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                host: str,
                port: int,
                database_name: str,
                collection_name: str,
                user: str = None,
                password: str = None,):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        self.client = None

    def insert_data(self, query):
        ...

    def delete_data(self, query):
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...
```

```
"""_____"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        self.client = None
        self.db = None

    def insert_data(self, query):
        ...

    def delete_data(self, query):
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...
"""_____"""
import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        self.client = None
        self.db = None
        self.collection = None
```

```
    def insert_data(self, query):
        ...

    def delete_data(self, query):
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...



"""_____"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pass
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, query):
        ...

    def delete_data(self, query):
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...



"""_____"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
```

```
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pass
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, data):
        self.collection.insert_one(data)

    def delete_data(self, query):
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...



"""_____________________________________________________________________"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pass
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, data):
        if not self.client:
            raise Exception("Not connected to the database.")
        self.collection.insert_one(data)

    def delete_data(self, query):
```

```
        ...

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...



"""_____"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, data):
        if not self.client:
            raise Exception("Not connected to the database.")
        self.collection.insert_one(data)

    def delete_data(self, query):
        self.collection.delete_one(query)

    def update_data(self, query):
        ...

    def select_data(self, query):
        ...



"""_____"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        self.host = host
        self.port = port
        self.user = user
```

```
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None

    def connect(self):
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, data):
        if not self.client:
            raise Exception("Not connected to the database.")
        self.collection.insert_one(data)

    def delete_data(self, data):
        self.collection.delete_one(data)

    def update_data(self, where, data):
        self.collection.update_one(where, data)


    def select_data(self, query):
        ...



"""_____________________________________________________________________"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None
        self.collection = None

    def connect(self):
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, data):
        if not self.client:
            raise Exception("Not connected to the database.")
        self.collection.insert_one(data)
```

```
    def delete_data(self, data):
        self.collection.delete_one(data)

    def update_data(self, where, data):
        self.collection.update_one(where, data)


    def select_data(self, query=None):
        result = list(self.collection.find(query))

        return result


"""_____"""


import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    """
    This class implements the DatabaseModule interface for MongoDB.

    Attributes:
        host (str): The host address of the database.
        port (int): The port of the database.
        user (str): The user of the database.
        password (str): The password of the database.
        database_name (str): The name of the database.
        collection_name (str): The name of the collection.
        client (MongoClient): The MongoClient object.
        db (Database): The Database object.

    Methods:
        connect: Connects to the database.
        disconnect: Disconnects from the database.
        insert_data: Inserts data into the database.
        delete_data: Deletes data from the database.
        update_data: Updates data in the database.
        select_data: Selects data from the database.
    """
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        """
        Constructor method.

        Args:
            host (str): The host address of the database.
            port (int): The port of the database.
            database_name (str): The name of the database.
            collection_name (str): The name of the collection.
            user (str): The user of the database.
            password (str): The password of the database.
        """
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None
        self.collection = None

    def connect(self):
        """
        Connect to the database.
        Raises:
            Exception: If already connected to the database.
        """
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
```

```
        """
        Disconnect from the database.
        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, data):
        """
        Execute a database query.

        Args:
            query (str): The query to execute.

        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.collection.insert_one(data)

    def delete_data(self, data):
        """
        Delete data from the database.

        Args:
            data (dict): The data to delete.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.delete_one(data)

    def update_data(self, where, data):
        """
        Update data in the database.

        Args:
            where (dict): The data to update.
            data (dict): The data to update to.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.update_one(where, data)

    def select_data(self, query=None):
        """
        Fetch data from the database.

        Args:
            query (dict): The query to execute.

        Returns:
            list: The result of the query.
        """
        ...



"""_____"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    """
    This class implements the DatabaseModule interface for MongoDB.

    Attributes:
        host (str): The host address of the database.
        port (int): The port of the database.
        user (str): The user of the database.
        password (str): The password of the database.
        database_name (str): The name of the database.
        collection_name (str): The name of the collection.
        client (MongoClient): The MongoClient object.
```

```
        db (Database): The Database object.

    Methods:
        connect: Connects to the database.
        disconnect: Disconnects from the database.
        insert_data: Inserts data into the database.
        delete_data: Deletes data from the database.
        update_data: Updates data in the database.
        select_data: Selects data from the database.
    """
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        """
        Constructor method.

        Args:
            host (str): The host address of the database.
            port (int): The port of the database.
            database_name (str): The name of the database.
            collection_name (str): The name of the collection.
            user (str): The user of the database.
            password (str): The password of the database.
        """
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None
        self.collection = None

    def connect(self):
        """
        Connect to the database.
        Raises:
            Exception: If already connected to the database.
        """
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        """
        Disconnect from the database.
        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, data):
        """
        Execute a database query.

        Args:
            query (str): The query to execute.

        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.collection.insert_one(data)

    def delete_data(self, data):
        """
        Delete data from the database.

        Args:
            data (dict): The data to delete.
```

```
        Raises:
            Exception: If not connected to the database.
        """
        self.collection.delete_one(data)

    def update_data(self, where, data):
        """
        Update data in the database.

        Args:
            where (dict): The data to update.
            data (dict): The data to update to.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.update_one(where, data)

    def select_data(self, query=None):
        """
        Fetch data from the database.

        Args:
            query (dict): The query to execute.

        Returns:
            list: The result of the query.
        """
        return [dict({"_id": "semente de bacon", "test": "test3"})]



""" ____________________________________________________________________________ """



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    """
    This class implements the DatabaseModule interface for MongoDB.

    Attributes:
        host (str): The host address of the database.
        port (int): The port of the database.
        user (str): The user of the database.
        password (str): The password of the database.
        database_name (str): The name of the database.
        collection_name (str): The name of the collection.
        client (MongoClient): The MongoClient object.
        db (Database): The Database object.

    Methods:
        connect: Connects to the database.
        disconnect: Disconnects from the database.
        insert_data: Inserts data into the database.
        delete_data: Deletes data from the database.
        update_data: Updates data in the database.
        select_data: Selects data from the database.
    """
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        """
        Constructor method.

        Args:
            host (str): The host address of the database.
            port (int): The port of the database.
            database_name (str): The name of the database.
            collection_name (str): The name of the collection.
            user (str): The user of the database.
            password (str): The password of the database.
        """
        self.host = host
        self.port = port
        self.user = user
```

```
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None
        self.collection = None

    def connect(self):
        """
        Connect to the database.
        Raises:
            Exception: If already connected to the database.
        """
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        """
        Disconnect from the database.
        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, data):
        """
        Execute a database query.

        Args:
            query (str): The query to execute.

        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.collection.insert_one(data)

    def delete_data(self, data):
        """
        Delete data from the database.

        Args:
            data (dict): The data to delete.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.delete_one(data)

    def update_data(self, where, data):
        """
        Update data in the database.

        Args:
            where (dict): The data to update.
            data (dict): The data to update to.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.update_one(where, data)

    def select_data(self, query=None):
        """
        Fetch data from the database.

        Args:
            query (dict): The query to execute.

        Returns:
            list: The result of the query.
        """
        return [dict({"_id": "semente de bacon", "test": "test3"})]
```

```
"""_____"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    """
    This class implements the DatabaseModule interface for MongoDB.

    Attributes:
        host (str): The host address of the database.
        port (int): The port of the database.
        user (str): The user of the database.
        password (str): The password of the database.
        database_name (str): The name of the database.
        collection_name (str): The name of the collection.
        client (MongoClient): The MongoClient object.
        db (Database): The Database object.

    Methods:
        connect: Connects to the database.
        disconnect: Disconnects from the database.
        insert_data: Inserts data into the database.
        delete_data: Deletes data from the database.
        update_data: Updates data in the database.
        select_data: Selects data from the database.
    """
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        """
        Constructor method.

        Args:
            host (str): The host address of the database.
            port (int): The port of the database.
            database_name (str): The name of the database.
            collection_name (str): The name of the collection.
            user (str): The user of the database.
            password (str): The password of the database.
        """
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None
        self.collection = None

    def connect(self):
        """
        Connect to the database.
        Raises:
            Exception: If already connected to the database.
        """
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        """
        Disconnect from the database.
        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None
```

```
    def insert_data(self, data):
        """
        Execute a database query.

        Args:
            query (str): The query to execute.

        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.collection.insert_one(data)

    def delete_data(self, data):
        """
        Delete data from the database.

        Args:
            data (dict): The data to delete.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.delete_one(data)

    def update_data(self, where, data):
        """
        Update data in the database.

        Args:
            where (dict): The data to update.
            data (dict): The data to update to.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.update_one(where, data)

    def select_data(self, query=None):
        """
        Fetch data from the database.

        Args:
            query (dict): The query to execute.

        Returns:
            list: The result of the query.
        """
        if query == {"test": "test3"}:
            return [dict({"_id": "semente de bacon", "test": "test3"})]

        return [dict({"_id": "Puffle do cowboy beebop", "test": "test4"})]



"""_____"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    """
    This class implements the DatabaseModule interface for MongoDB.

    Attributes:
        host (str): The host address of the database.
        port (int): The port of the database.
        user (str): The user of the database.
        password (str): The password of the database.
        database_name (str): The name of the database.
        collection_name (str): The name of the collection.
        client (MongoClient): The MongoClient object.
        db (Database): The Database object.

    Methods:
        connect: Connects to the database.
        disconnect: Disconnects from the database.
        insert_data: Inserts data into the database.
        delete_data: Deletes data from the database.
```

```
        update_data: Updates data in the database.
        select_data: Selects data from the database.
    """
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        """
        Constructor method.

        Args:
            host (str): The host address of the database.
            port (int): The port of the database.
            database_name (str): The name of the database.
            collection_name (str): The name of the collection.
            user (str): The user of the database.
            password (str): The password of the database.
        """
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None
        self.collection = None

    def connect(self):
        """
        Connect to the database.
        Raises:
            Exception: If already connected to the database.
        """
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        """
        Disconnect from the database.
        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, data):
        """
        Execute a database query.

        Args:
            query (str): The query to execute.

        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.collection.insert_one(data)

    def delete_data(self, data):
        """
        Delete data from the database.

        Args:
            data (dict): The data to delete.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.delete_one(data)

    def update_data(self, where, data):
```

```
        """
        Update data in the database.

        Args:
            where (dict): The data to update.
            data (dict): The data to update to.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.update_one(where, data)

    def select_data(self, query=None):
        """
        Fetch data from the database.

        Args:
            query (dict): The query to execute.

        Returns:
            list: The result of the query.
        """
        if query == {"test": "test3"}:
            return [dict({"_id": "semente de bacon", "test": "test3"})]

        if query == {"test": "test4"}:
            return [dict({"_id": "Puffle do cowboy beebop", "test": "test4"})]

        return [dict({"_id": "KAHOOOOOOOOOT", "test": "test5"})]



"""_____________REFACTOR_____________________________________________________"""



import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    """
    This class implements the DatabaseModule interface for MongoDB.

    Attributes:
        host (str): The host address of the database.
        port (int): The port of the database.
        user (str): The user of the database.
        password (str): The password of the database.
        database_name (str): The name of the database.
        collection_name (str): The name of the collection.
        client (MongoClient): The MongoClient object.
        db (Database): The Database object.

    Methods:
        connect: Connects to the database.
        disconnect: Disconnects from the database.
        insert_data: Inserts data into the database.
        delete_data: Deletes data from the database.
        update_data: Updates data in the database.
        select_data: Selects data from the database.
    """
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 collection_name: str,
                 user: str = None,
                 password: str = None,):
        """
        Constructor method.

        Args:
            host (str): The host address of the database.
            port (int): The port of the database.
            database_name (str): The name of the database.
            collection_name (str): The name of the collection.
            user (str): The user of the database.
            password (str): The password of the database.
        """
        self.host = host
        self.port = port
        self.user = user
        self.password = password
```

```
        self.database_name = database_name
        self.collection_name = collection_name
        self.client = None
        self.db = None
        self.collection = None

    def connect(self):
        """
        Connect to the database.
        Raises:
            Exception: If already connected to the database.
        """
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]
        self.collection = self.db[self.collection_name]

    def disconnect(self):
        """
        Disconnect from the database.
        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.client = None
        self.db = None
        self.collection = None

    def insert_data(self, data):
        """
        Execute a database query.

        Args:
            query (str): The query to execute.

        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.collection.insert_one(data)

    def delete_data(self, data):
        """
        Delete data from the database.

        Args:
            data (dict): The data to delete.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.delete_one(data)

    def update_data(self, where, data):
        """
        Update data in the database.

        Args:
            where (dict): The data to update.
            data (dict): The data to update to.

        Raises:
            Exception: If not connected to the database.
        """
        self.collection.update_one(where, data)

    def select_data(self, query=None):
        """
        Fetch data from the database.

        Args:
            query (dict): The query to execute.

        Returns:
            list: The result of the query.
        """
        result = list(self.collection.find(query))

        return result
```

```
"""_____REFACTOR_____"""


import pymongo
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    """
    This class implements the DatabaseModule interface for MongoDB.

    Attributes:
        host (str): The host address of the database.
        port (int): The port of the database.
        user (str): The user of the database.
        password (str): The password of the database.
        database_name (str): The name of the database.
        client (MongoClient): The MongoClient object.
        db (Database): The Database object.

    Methods:
        connect: Connects to the database.
        disconnect: Disconnects from the database.
        insert_data: Inserts data into the database.
        delete_data: Deletes data from the database.
        update_data: Updates data in the database.
        select_data: Selects data from the database.
    """
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 user: str = None,
                 password: str = None,):
        """
        Constructor method.

        Args:
            host (str): The host address of the database.
            port (int): The port of the database.
            database_name (str): The name of the database.
            user (str): The user of the database.
            password (str): The password of the database.
        """
        self.host = host
        self.port = port
        self.user = user
        self.password = password
        self.database_name = database_name
        self.client = None
        self.db = None

    def connect(self):
        """
        Connect to the database.
        Raises:
            Exception: If already connected to the database.
        """
        if self.client:
            raise Exception("Already connected to the database.")

        self.client = pymongo.MongoClient(host=self.host, port=self.port, username=self.user, password=self.pas
        self.db = self.client[self.database_name]

    def disconnect(self):
        """
        Disconnect from the database.
        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        # disconnect from the database
        self.client.close()

        self.client = None
        self.db = None
        self.collection = None
```

```
    def insert_data(self, collection_name, data):
        """
        Execute a database query.

        Args:
            query (str): The query to execute.

        Raises:
            Exception: If not connected to the database.
        """
        if not self.client:
            raise Exception("Not connected to the database.")
        self.db[collection_name].insert_one(data)

    def delete_data(self, collection_name, condition):
        """
        Delete data from the database.

        Args:
            data (dict): The data to delete.

        Raises:
            Exception: If not connected to the database.
        """
        self.db[collection_name].delete_one(condition)

    def update_data(self, collection_name, condition, new_data):
        """
        Update data in the database.

        Args:
            where (dict): The data to update.
            data (dict): The data to update to.

        Raises:
            Exception: If not connected to the database.
        """
        self.db[collection_name].update_one(condition, new_data)

    def select_data(self, collection_name, condition):
        """
        Fetch data from the database.

        Args:
            query (dict): The query to execute.

        Returns:
            list: The result of the query.
        """
        result = list(self.db[collection_name].find(condition))

        return result



"""_____________REFACTOR_____________________________________________________"""



import pymongo
import sys
sys.path.append('/home/gustavo/ES/Engenharia_Software/')
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    """
    This class implements the DatabaseModule interface for MongoDB.

    Attributes:
        host (str): The host address of the database.
        port (int): The port of the database.
        user (str): The user of the database.
        password (str): The password of the database.
        database_name (str): The name of the database.
        client (MongoClient): The MongoClient object.
        db (Database): The Database object.

    Methods:
        connect: Connects to the database.
        disconnect: Disconnects from the database.
        insert_data: Inserts data into the database.
        delete_data: Deletes data from the database.
```

```
        update_data: Updates data in the database.
        select_data: Selects data from the database.
    """
    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 user: str = None,
                 password: str = None,):
        """
        Constructor method.

        Args:
            host (str): The host address of the database.
            port (int): The port of the database.
            database_name (str): The name of the database.
            user (str): The user of the database.
            password (str): The password of the database.
        """
        self._host = host
        self._port = port
        self._user = user
        self._password = password
        self._database_name = database_name
        self._client = None
        self._db = None

    def connect(self):
        """
        Connect to the database.
        Raises:
            Exception: If already connected to the database.
        """
        if self._client:
            raise Exception("Already connected to the database.")

        self._client = pymongo.MongoClient(host=self._host, port=self._port, username=self._user, password=self
        self._db = self._client[self._database_name]

    def disconnect(self):
        """
        Disconnect from the database.
        Raises:
            Exception: If not connected to the database.
        """
        if not self._client:
            raise Exception("Not connected to the database.")
        # disconnect from the database
        self._client.close()

        self._client = None
        self._db = None
        self.collection = None


    def insert_data(self, collection_name, data):
        """
        Execute a database query.

        Args:
            query (str): The query to execute.

        Raises:
            Exception: If not connected to the database.
        """
        if not self._client:
            raise Exception("Not connected to the database.")
        self._db[collection_name].insert_one(data)

    def delete_data(self, collection_name, condition):
        """
        Delete data from the database.

        Args:
            data (dict): The data to delete.

        Raises:
            Exception: If not connected to the database.
        """
        self._db[collection_name].delete_one(condition)

    def update_data(self, collection_name, condition, new_data):
        """
```

```
        Update data in the database.

        Args:
            where (dict): The data to update.
            data (dict): The data to update to.

        Raises:
            Exception: If not connected to the database.
        """
        self._db[collection_name].update_one(condition, new_data)

    def select_data(self, collection_name, condition):
        """
        Fetch data from the database.

        Args:
            query (dict): The query to execute.

        Returns:
            list: The result of the query.
        """
        result = list(self._db[collection_name].find(condition))

        return result



"""_____________SINGLETON_____________________________________________________"""

import pymongo
import sys
from src.database.database_module import DatabaseModule

class MongoModule(DatabaseModule):
    """
    This class implements the DatabaseModule interface for MongoDB.

    Attributes:
        host (str): The host address of the database.
        port (int): The port of the database.
        user (str): The user of the database.
        password (str): The password of the database.
        database_name (str): The name of the database.
        client (MongoClient): The MongoClient object.
        db (Database): The Database object.

    Methods:
        connect: Connects to the database.
        disconnect: Disconnects from the database.
        insert_data: Inserts data into the database.
        delete_data: Deletes data from the database.
        update_data: Updates data in the database.
        select_data: Selects data from the database.
    """
    _instance = None

    def __init__(self,
                 host: str,
                 port: int,
                 database_name: str,
                 user: str = None,
                 password: str = None,):
        """
        Constructor method.

        Args:
            host (str): The host address of the database.
            port (int): The port of the database.
            database_name (str): The name of the database.
            user (str): The user of the database.
            password (str): The password of the database.
        """
        self._host = host
        self._port = port
        self._user = user
        self._password = password
        self._database_name = database_name
        self._client = None
        self._db = None

    def __new__(cls, *args, **kwargs):
        """
        Singleton constructor method.
```

```
        Returns:
            MongoModule: The MongoModule instance.
        """
        if not cls._instance:
            cls._instance = super(MongoModule, cls).__new__(cls)
        return cls._instance

    def connect(self):
        """
        Connect to the database.
        Raises:
            Exception: If already connected to the database.
        """
        if self._client:
            raise Exception("Already connected to the database.")

        self._client = pymongo.MongoClient(host=self._host,
                                           port=self._port,
                                           username=self._user,
                                           password=self._password)
        self._db = self._client[self._database_name]

    def disconnect(self):
        """
        Disconnect from the database.
        Raises:
            Exception: If not connected to the database.
        """
        if not self._client:
            raise Exception("Not connected to the database.")
        # disconnect from the database
        self._client.close()

        self._client = None
        self._db = None
        self.collection = None


    def insert_data(self, collection_name, data):
        """
        Execute a database query.

        Args:
            query (str): The query to execute.

        Raises:
            Exception: If not connected to the database.
        """
        if not self._client:
            raise Exception("Not connected to the database.")
        self._db[collection_name].insert_one(data)

    def delete_data(self, collection_name, condition):
        """
        Delete data from the database.

        Args:
            data (dict): The data to delete.

        Raises:
            Exception: If not connected to the database.
        """
        self._db[collection_name].delete_one(condition)

    def update_data(self, collection_name, condition, new_data):
        """
        Update data in the database.

        Args:
            where (dict): The data to update.
            data (dict): The data to update to.

        Raises:
            Exception: If not connected to the database.
        """
        self._db[collection_name].update_one(condition, new_data)

    def select_data(self, collection_name, condition):
        """
        Fetch data from the database.

        Args:
```

```
            query (dict): The query to execute.

        Returns:
            list: The result of the query.
        """
        result = list(self._db[collection_name].find(condition))

        return result

if __name__ == "__main__":
    mongo_module = MongoModule(host="localhost",
                               port=27017,
                               database_name="test")

    mongo_module2 = MongoModule(host="localhost",
                               port=27017,
                               database_name="test")

    if mongo_module == mongo_module2:
        print("Singleton works, both variables contain the same instance.")
    else:
        print("Singleton failed, variables contain different instances.")
```

In [ ]:

```
import unittest
import unittest.mock
# from src.database.mongo_module import MongoModule
import sys
sys.path.append('/home/gustavo/ES/Engenharia_Software/')

from src.database.mongo_module import MongoModule


class TestMongoModule(unittest.TestCase):

    def setUp(self):
        self.mongo_module = MongoModule(host="localhost", collection_name="test_collection")

    def test_connect(self):
        with unittest.mock.patch('pymongo.MongoClient') as mock_mongo:
            self.mongo_module.connect()
            mock_mongo.assert_called_once_with(host="localhost")

    def test_disconnect(self):
        pass

    def test_insert_data(self):
        pass

    def test_delete_data(self):
        pass

    def test_update_data(self):
        pass

    def test_select_data(self):
        pass

if __name__ == '__main__':
    unittest.main()
```