

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«Национальный исследовательский университет ИТМО»**

**(Университет ИТМО)**

**Факультет прикладной информатики**

**Образовательная программа Мобильные и сетевые технологии**

**Направление подготовки 09.03.03 Мобильные и сетевые технологии**

**Дисциплина: Проектирование и реализация баз данных**

**Практическая работа №6.1**

**“ВВЕДЕНИЕ В СУБД MONGODB. УСТАНОВКА MONGODB. НАЧАЛО РАБОТЫ  
С БД”**

**Обучающийся: Скоблилова Виктория Васильевна K3241**

**Проверил: Говорова Марина Михайловна**

Санкт-Петербург,

2025

**Цель:** овладеть практическими навыками установки СУБД MongoDB.

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД MongoDB 4+, 8.0.4 (последняя)

**Практическое задание:**

1. Установите MongoDB для обеих типов систем (32/64 бита).
2. Проверьте работоспособность системы запуском клиента mongo.
3. Выполните методы:
  - a) db.help()
  - b) db.help
  - c) db.stats()
4. Создайте БД learn.
5. Получите список доступных БД
6. СОздайте коллекцию unicorns, вставив в нее документ {name: 'Aurora', gender:'f', weight: 450}.
7. Просмотрите список текущих коллекций.
8. Переименуйте коллекцию unicorns.
9. Просмотрите статистику коллекции.
- 10.Удалите коллекцию.
- 11.Удалите БД learn.

## Выполнение:

1. Мы установили MongoDB и MongoDB Shell
2. Запускаем клиента:

```
mongosh mongodb://localhost/
2025-05-28T21:00:04.282+03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
test> db.version
[Function: version] AsyncFunction {
  apiVersions: [ 0, 0 ],
  returnsPromise: true,
  serverVersions: [ '0.0.0', '999.999.999' ],
  topologies: [ 'ReplSet', 'Sharded', 'LoadBalanced', 'Standalone' ],
  returnType: { type: 'unknown', attributes: {} },
  deprecated: false,
  platforms: [ 'Compass', 'Browser', 'CLI' ],
  isDirectShellCommand: false,
  acceptsRawInput: false,
  shellCommandCompleter: undefined,
  help: [Function (anonymous)] Help
}
test> db.version()
8.0.9
test> show dbs
admin    40.00 KiB
config  12.00 KiB
local   72.00 KiB
test> mongodb://localhost/
...
test> db.system.namespaces.find();
```

3. Выполняем методы:

a)db.help()

```
mongosh mongodb://localhost/
test> db.help()

Database Class:

  getMongo           Returns the current database connection
  getName            Returns the name of the DB
  getCollectionNames Returns an array containing the names of all collections in the current database.
  getCollectionInfos Returns an array of documents with collection information, i.e. collection name and options, for the current database.
  runCommand         Runs an arbitrary command on the database.
  adminCommand       Runs an arbitrary command against the admin database.
  aggregate          Runs a specified admin/diagnostic pipeline which does not require an underlying collection.
  getSiblingDB       Returns another database without modifying the db variable in the shell environment.
  getCollection      Returns a collection or a view object that is functionally equivalent to using the db.<collectionName>.
  dropDatabase       Removes the current database, deleting the associated data files.
  createUser         Creates a new user for the database on which the method is run. db.createUser() returns a duplicate user error if the user already exists on the database.
  updateUser         Updates the user's profile on the database on which you run the method. An update to a field completely replaces the previous field's values. This includes updates to the user's roles array.
  changeUserPassword Updates a user's password. Run the method in the database where the user is defined, i.e. the database you created the user.
  logout             Ends the current authentication session. This function has no effect if the current session is not authenticated.
```

b)db.help

```
test> db.help

Database Class:

  getMongo           Returns the current database connection
  getName            Returns the name of the DB
  getCollectionNames Returns an array containing the names of all collections in the current database.
  getCollectionInfos Returns an array of documents with collection information, i.e. collection name and options, for the current database.
  runCommand         Runs an arbitrary command on the database.
  adminCommand       Runs an arbitrary command against the admin database.
  aggregate          Runs a specified admin/diagnostic pipeline which does not require an underlying collection.
  getSiblingDB       Returns another database without modifying the db variable in the shell environment.
  getCollection      Returns a collection or a view object that is functionally equivalent to using the db.<collectionName>.
  dropDatabase       Removes the current database, deleting the associated data files.
  createUser         Creates a new user for the database on which the method is run. db.createUser() returns a duplicate user error if the user already exists on the database.
  updateUser         Updates the user's profile on the database on which you run the method. An update to a field completely replaces the previous field's values. This includes updates to the user's roles array.
  changeUserPassword Updates a user's password. Run the method in the database where the user is defined, i.e. the database you created the user.
```

c)db.stats()

```
test> db.stats()
{
  db: 'test',
  collections: Long('0'),
  views: Long('0'),
  objects: Long('0'),
  avgObjSize: 0,
  dataSize: 0,
  storageSize: 0,
  indexes: Long('0'),
  indexSize: 0,
  totalSize: 0,
  scaleFactor: Long('1'),
  fsUsedSize: 0,
  fsTotalSize: 0,
  ok: 1
}
test> |
```

4. Введем команду test> use learn, switched to db learn и получим список доступных БД:

```
switched to db learn
learn> show dbs
admin      40.00 KiB
config     60.00 KiB
local      72.00 KiB
```

Для отображения базы данных learn нам понадобится создать коллекцию с документами.

5. Создадим коллекцию unicorns и далее проверим список текущих коллекций:

```
learn> db.unicorns.insertOne({name: 'Aurora', gender: 'f', weight: 450})
{
  acknowledged: true,
  insertedId: ObjectId('683775fbac02869fda6c4bd0')
}
learn> |

learn> show collections
unicorns
learn> |
```

6. Изменим название коллекции unicorns на sweet unicorns:

```
learn> db.unicorns.renameCollection("sweet_unicorns")
{ ok: 1 }
learn> show collections
sweet_unicorns
```

## 7. Посмотрим статистику коллекции

```
learn> db.sweet_unicorns.stats()
{
  ok: 1,
  capped: false,
  wiredTiger: {
    metadata: { formatVersion: 1 },
    creationString: 'access_pattern_hint=none,allocation_size=none,durable_timestamp=none,read_timestamp=none,write_timestamp=none,checksum=on,collator=,collator=,columns=,format=btree,huffman_key=,huffman_value=,ignore_in_metadata=,format=btree,huffman_key=,huffman_value=,ignore_in_metadata=oldest_timestamp,enabled=false,file_metadata=,metadata_format=0,internal_key_max=0,internal_key_truncate=true,internal_page_max=32KB,leaf_page_max=32KB,leaf_value_max=64MB,log=(enabled=true),log_flush=,bloom_hash_count=8,bloom_oldest=false,chunk_count_limit=0,generation=0,suffix=),merge_max=15,merge_min=0),memory_page_max=0,prefix_compression=false,prefix_compression_min=4,lit_pct=90,tiered_storage=(auth_token=,bucket=,bucket_prefix_size=0),type=file,value_format=u,verbose=[],write_timestamp=0)',
    type: 'file',
    uri: 'statistics/table/collection=0-1838126561231811137'
  },
  rollbackToStableUpdate: 0,
  'updateConflicts': 0
},
sharded: false,
size: 65,
count: 1,
numOrphanDocs: 0,
storageSize: 20480,
totalIndexSize: 20480,
totalSize: 40960,
indexSizes: { _id_: 20480 },
avgObjSize: 65,
ns: 'learn.sweet_unicorns',
nindexes: 1,
scaleFactor: 1
```

## 8. И в заключение удалим sweet unicorns и learn

```
learn> db.sweet_unicorns.drop()
true
learn> show collections

learn> |

learn> db.dropDatabase()
{ ok: 1, dropped: 'learn' }
learn> |
```

## Вывод по проделанной работе:

В процессе выполнения задания мы приобрели практические навыки по установке MongoDB, установили программное обеспечение и настроили запуск сервера баз данных. Также проверили функциональность системы управления базами данных.