|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра инструментального и прикладного программного обеспечения | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1** | |
| **по дисциплине** | |
| **«**Настройка и администрирование сервисного программного обеспечения**»**  **Тема: «Установка приложения по работе с контейнерами Docker»** | |
|  | |
| Выполнил студент группы ИКБО-10-23 | Харитонов А.Н. |
| Принял преподаватель | Зарипов Е.А. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2024

# **Цель работы**

Получить навыки по развертыванию приложения Docker.

# **Теоретическое введение**

Docker – это платформа для контейнеризации, которая позволяет упаковывать приложения и все необходимые зависимости в один контейнер.

Контейнер – изолированная среда, которая работает независимо от основной операционной системы и других приложений.

Docker Daemon – фоновый сервис, запущенный на хост-машине, который отвечает за создание, запуск и уничтожение Docker-контейнеров.

Docker Client – утилита командной строки, которая позволяет пользователю взаимодействовать с демоном.

Демон – это процесс, который запущен на операционной системе, взаимодействующий с клиентом.

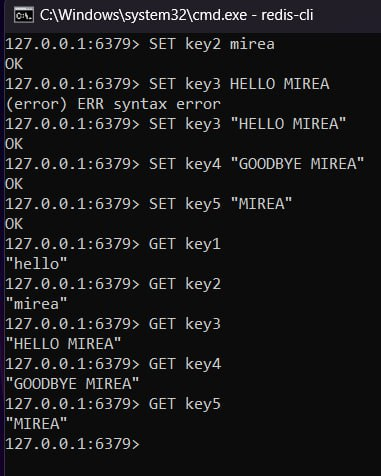
Redis – база данных с открытым исходным кодом. Она используется для хранения данных. Огромное преимущество данной БД – ее скорость, все данные хранятся в оперативной памяти, из-за чего Redis работает быстрее своих конкурентов.

# **Ход работы**

**Задание #4**

Для создания пары ключ-значение в базе Redis необходимо использовать команду *SET [КЛЮЧ] [ЗНАЧЕНИЕ].*

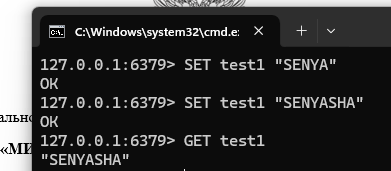
Для чтения значения из БД необходимо использовать команду *GET [КЛЮЧ].*



Изображение 1.1 – Решение задания 1.4

**Задание #5**

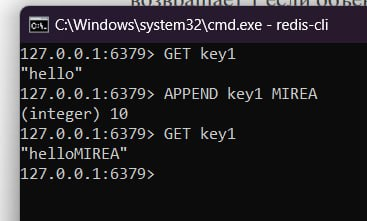
Для изменения значения в БД необходимо использовать команду *SET [КЛЮЧ] [ЗНАЧЕНИЕ]*

**

Изображение 1.2 – Решение задания 1.5

**Задание #6**

Для добавления строки к уже существующему значению необходимо использовать команду *APPEND [КЛЮЧ] [ЗНАЧЕНИЕ].*

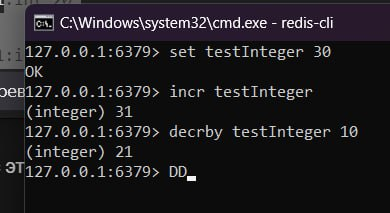


Изображение 1.3 – Решение задания 1.6

**Задание #7**

Для добавления числа используется команда *SET [КЛЮЧ] [ЗНАЧЕНИЕ].*

Для изменения значения числа можно использовать команды *INCR/DECR [КЛЮЧ]*, либо *INCRBY/DECRBY [КЛЮЧ] [ДЕЛЬТА].*



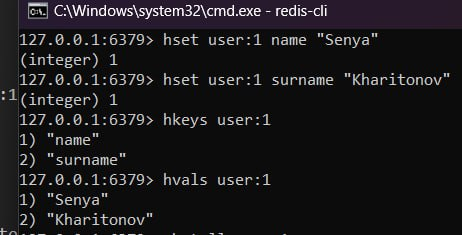
Изображение 1.4 – Решение задания 1.7

**Задание #8**

Для создания ключа со значением типа хеш-таблица необходимо использовать команду *HSET [КЛЮЧ] [ПОЛЕ] [ЗНАЧЕНИЕ].*

Для вывода ключей хеш-таблицы используется команда *HKEYS [КЛЮЧ].*

Для вывода значений хэш-таблицы используется команда *HVALS [КЛЮЧ].*

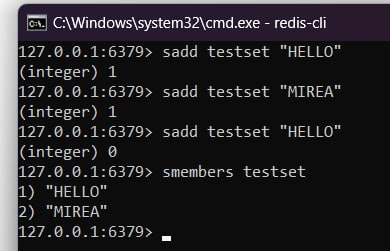


Изображение 1.5 – Решение задания 1.8

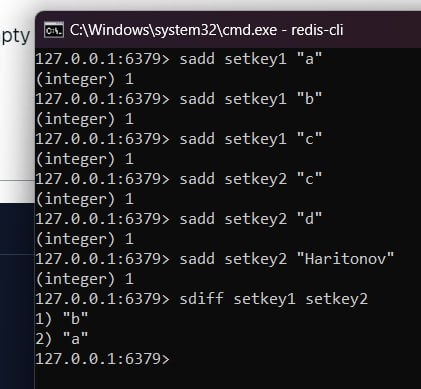
**Задание #9**

Для работы со множествами используются команды:

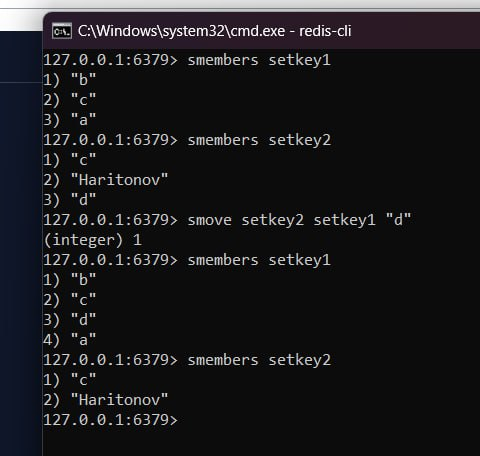
* *SADD [МНОЖЕСТВО] [ЗНАЧЕНИЕ]* – Добавить в множество
* *SDIFF [МНОЖЕСТВО] [МНОЖЕСТВО]* – Пересечь множества
* *SMOVE [МНОЖЕСТВО] [МНОЖЕСТВО] [ЗНАЧЕНИЕ]* – Перенести значение из одного множества в другое
* *SPOP [МНОЖЕСТВО]* – Вывести и удалить первое значение множества
* *SUNION [МНОЖЕСТВО] [МНОЖЕСТВО]* – Объединить множеств
* *SREM [МНОЖЕСТВО] [ЗНАЧЕНИЕ]* – Удалить значение из множества
* *SMEMBERS [МНОЖЕСТВО]* – Вывести значения множества



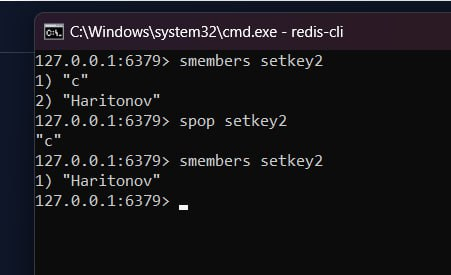
Изображение 1.6 – Демонстрация работы SADD



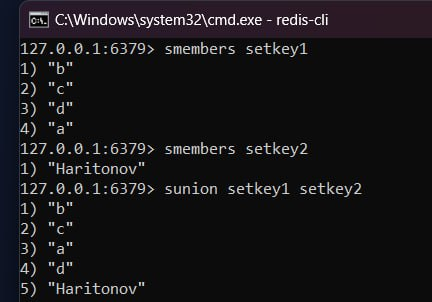
Изображение 1.7 – Демонстрация работы SDIFF



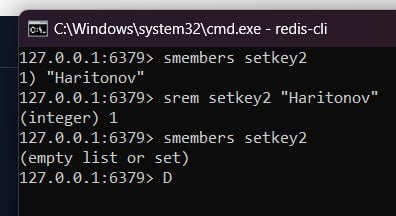
Изображение 1.8 – Демонстрация работы SMOVE



Изображение 1.9 – Демонстрация работы SPOP



Изображение 1.10 – Демонстрация работы SUNION

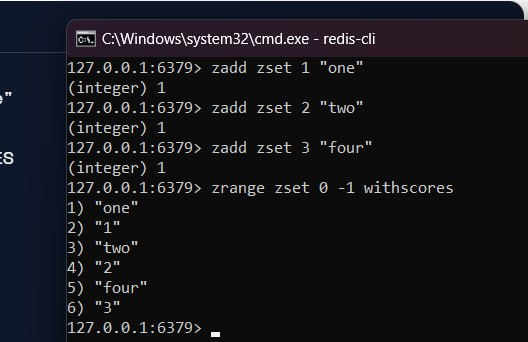


Изображение 1.11 – Демонстрация работы SREM

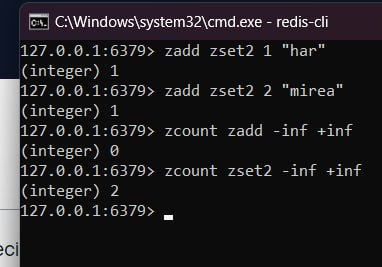
**Задание #10**

Для работы с упорядоченными множествами используются команды:

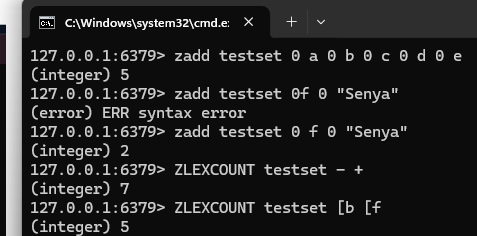
* *ZADD [МНОЖЕСТВО] [КЛЮЧ] [ЗНАЧЕНИЕ]* – Добавить в множество
* *ZCOUNT [МНОЖЕСТВО] [МИН] [МАКС]* – Посчитать кол-во элементов
* *ZDIFF [МНОЖЕСТВО] [КЛЮЧ] [МНОЖЕСТВО] [МНОЖЕСТВО]* – Вычислить разницу между множествами
* *ZPOPMAX [МНОЖЕСТВО] [КОЛ-ВО]* – Вернуть N наибольших значений
* *ZPOPMIN [МНОЖЕСТВО] [КОЛ-ВО]* – Вернуть N наименьших значений
* *ZUNION [КЛЮЧ] [МНОЖЕСТВО] [МНОЖЕСТВО]* – объединить множества
* *ZMSCORE [МНОЖЕСТВО] [ЗНАЧЕНИЕ]* – Получение ключа(оценки) по значению
* *ZLEXCOUNT [МНОЖЕСТВО] [МИН] [МАКС]* – Подсчет кол-во элементов в отсортированном множестве.
* *ZRANGE [МНОЖЕСТВО] [НАЧАЛО] [КОНЕЦ]* – Вывод значений множества



Изображение 1.12 – Демонстрация работы ZADD

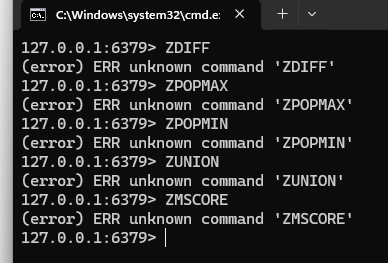


Изображение 1.13 – Демонстрация работы ZCOUNT



Изображение 1.14 – Демонстрация работы ZLEXCOUNT

**Команды ZDIFF-ZMSCORE почему-то не работают.**

****

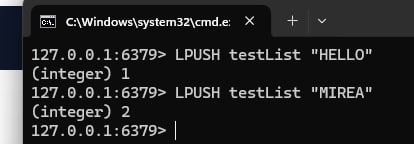
Изображение 1.15 – Неработающие команды

**Задание #11**

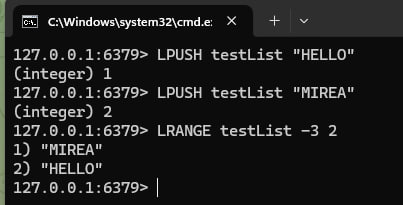
Для 11 задания я решил выбрать команды для работы с листом в Redis.

А именно:

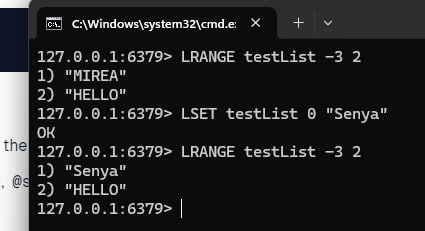
* *LPUSH [СПИСОК] [ЗНАЧЕНИЕ]* – Добавить элемент в список
* *LRANGE [СПИСОК] [МИН] [МАКС]* – Вывести элементы списка
* *LSET [СПИСОК] [ИНДЕКС] [ЗНАЧЕНИЕ]* – Изменение элемента списка
* *LREM [СПИСОК] [КОЛ-ВО] [ЭЛЕМЕНТ]* – Удалить значение из списка
* *LLEN [СПИСОК]* – Вывести длину списка



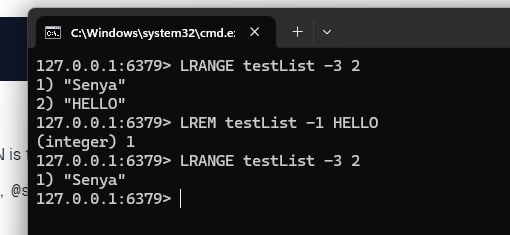
Изображение 1.16 – Демонстрация работы LPUSH



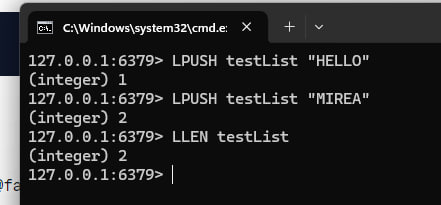
Изображение 1.17 – Демонстрация работы LRANGE



Изображение 1.18 – Демонстрация работы LSET



Изображение 1.19 – Демонстрация работы LREM



Изображение 1.20 – Демонстрация работы LLEN

# **Вывод**

В ходе работы я познакомился с Docker, научился развертывать приложения Docker, а также научился работать с базой данных Redis.