

**Київський національний університет
Імені Тараса Шевченка.**

Кафедра: Мережєвих та інтернет-технологій.

Лабораторна робота №7

з дисципліни: Бази даних та інформаційні системи

На тему: «Робота з Redis»

Студента 3 курсу:
Групи МІТ-31
Мулико Володимира

Київ - 2025р.

Мета: Закріпити розуміння роботи Redis та навчитися використовувати його основні можливості.

Хід Роботи

1. Встановлення Redis та запуск (Будемо використовувати Docker для розгортання redis на нашому ПК з ОС Windows).

```
C:\Users\Vovam>docker pull redis
Using default tag: latest
latest: Pulling from library/redis
8ae1ad8ce35e: Download complete
3ff860482ac5: Download complete
811af041d785: Download complete
4f4fb700ef54: Already exists
759939a29cb5: Download complete
cbb99c664e48: Download complete
8a628cdd7ccc: Download complete
a7e27cf18de4: Download complete
Digest: sha256:fbdbaea47b9ae4ecc2082ecdb4e1cea81e32176ffb1dcf643d422ad07427e5d9
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
```

Запускаємо контейнер:

```
C:\Users\Vovam>docker run --name redis-container -d -p 6379:6379 redis
41203cd4069311e95a99d5179a5fe844e2c18d7bd52f94fde4cee146728a0928
```

Підключаємося до redis cli:

```
C:\Users\Vovam>docker exec -it redis-container redis-cli
127.0.0.1:6379> |
```

2. Операції з рядками:

- Створіть ключ зі своїм ім'ям: **SET student "<Ваше ім'я>"**.

```
127.0.0.1:6379> SET student "Mylyko Volodymyr"
OK
```

- Отримайте значення командою **GET student**.

```
127.0.0.1:6379> GET student
"Mylyko Volodymyr"
```

- Виконайте **INCR mycounter** (двічі) та перевірте його значення.
(Команда INCR mycounter у Redis збільшує значення ключа mycounter на 1.)

```
127.0.0.1:6379> INCR mycounter
(integer) 1
127.0.0.1:6379> INCR mycounter
(integer) 2
127.0.0.1:6379> GET mycounter
"2"
```

3. Структури даних:

- Список (List):

- LPUSH tasks "Task1", LPUSH tasks "Task2"
- LRANGE tasks 0 -1, LPOP tasks

(Команда LPUSH tasks "Task1" додає "Task1" на початок списку з ключем tasks)

```
127.0.0.1:6379> LPUSH tasks "Task1"
(integer) 1
127.0.0.1:6379> LPUSH tasks "Task2"
(integer) 2
127.0.0.1:6379> LRANGE tasks 0 -1
1) "Task2"
2) "Task1"
127.0.0.1:6379> LPOP tasks
"Task2"
127.0.0.1:6379> |
```

- Множина (Set):

- SADD tech:set "Redis", SADD tech:set "PostgreSQL", SADD tech:set "MongoDB"
- SMEMBERS tech:set, SISMEMBER tech:set "Redis"

(команда SADD tech:set "Redis" додає елемент "Redis" до множини з ключем tech:set. Множина в Redis — це неупорядкована колекція унікальних значень.)

```
127.0.0.1:6379> SADD tech:set "Redis"
(integer) 1
127.0.0.1:6379> SADD tech:set "PostgreSQL"
(integer) 1
127.0.0.1:6379> SADD tech:set "MongoDB"
(integer) 1
127.0.0.1:6379> SMEMBERS tech:set
1) "Redis"
2) "PostgreSQL"
3) "MongoDB"
127.0.0.1:6379> SISMEMBER tech:set "Redis"
(integer) 1
```

- Хеш (Hash):

- HSET profile name "<Ваше ім'я>", HSET profile city "<Ваше місто>"
- HGET profile name, HGETALL profile

```
127.0.0.1:6379> HSET profile name "Mylyko Volodymyr"
(integer) 1
127.0.0.1:6379> HSET profile city "Kyiv"
(integer) 1
127.0.0.1:6379> HGET profile name
"Mylyko Volodymyr"
127.0.0.1:6379> HGETALL profile
```

- Sorted Set:
 - ZADD scores 85 "Student1", ZADD scores 92 "Student2", ZADD scores 74 "Student3"
 - ZREVRANGE scores 0 -1 WITHSCORES, ZRANK scores "Student1"

```
127.0.0.1:6379> ZADD scores 85 "Student1"
(integer) 1
127.0.0.1:6379> ZADD scores 92 "Student2"
(integer) 1
127.0.0.1:6379> ZADD scores 74 "Student3"
(integer) 1
127.0.0.1:6379> ZREVRANGE scores 0 -1 WITHSCORES
1) "Student2"
2) "92"
3) "Student1"
4) "85"
5) "Student3"
6) "74"
127.0.0.1:6379> ZRANK scores "Student1"
(integer) 1
```

4. Працюємо з TTL

- SET temp:data "Hello" EX 10
- GET temp:data, TTL temp:data

```
127.0.0.1:6379> SET temp:data "Hello" EX 10
OK
127.0.0.1:6379> GET temp:data
"Hello"
127.0.0.1:6379> TTL temp:data
(integer) -2
127.0.0.1:6379> GET temp:data
(nil)
```

5. Міні-програма з Redis (додатково)

Python або JavaScript-код для роботи з Redis.

```
import redis
import random

r = redis.Redis(host='localhost', port=6379,
decode_responses=True)

def start_game():
    secret = random.randint(1, 100)
    r.set('game:secret', secret)
    r.set('game:attempts', 0)
```

```

r.set('game:status', 'playing')
print("Я загадав число від 1 до 100. Спробуй вгадати!")

def play_game():
    while r.get('game:status') == 'playing':
        try:
            guess = int(input("Введи своє число: "))
        except ValueError:
            print("Це не число!")
            continue

        r.incr('game:attempts')
        secret = int(r.get('game:secret'))

        if guess < secret:
            print("Більше!")
        elif guess > secret:
            print("Менше!")
        else:
            attempts = r.get('game:attempts')
            print(f"Вітаю! Ти вгадав число за {attempts} спроб!")
            r.set('game:status', 'won')

if __name__ == "__main__":
    start_game()
    play_game()

```

(Використання redis для написання міні-гри вгадай число, яка використовує Redis як "мозок" гри)

6. Аналіз даних (теоретичне завдання)

- Де можна використати Redis у реальних проєктах.

⇒ **Кешування**: для прискорення доступу до часто використовуваних даних (наприклад, сторінки сайту).

⇒ **Черги повідомлень**: Redis List або Pub/Sub використовуються для обробки подій.

⇒ **Сесії користувачів**: зберігання даних сесій у веб-застосунках.

⇒ **Лічильники та рейтинги**: використання INCR/ZINCRBY/ZADD для підрахунку або ранжування.

⇒ **Часові дані**: ключі з TTL — тимчасові токени, коди підтвердження тощо.

Висновок: У ході лабораторної роботи було вивчено базові можливості Redis за допомогою Docker, зокрема встановлення, запуск Redis-сервера та підключення до нього через redis-cli, що підтверджувалось успішним виконанням команди PING. Було опрацьовано основні типи даних Redis: рядки (команди SET, GET, INCR), списки (LPUSH, LRANGE, LPOP), множини (SADD, SMEMBERS, SISMEMBER), хеші (HSET, HGET, HGETALL) і відсортовані множини (ZADD, ZREVRANGE, ZRANK), а також роботу з TTL (SET з параметром EX та команда TTL). Окрім того, реалізовано міні-програму на Python для демонстрації взаємодії з Redis: інкрементування лічильника, додавання задач у список та їх зчитування. У результаті роботи було закріплено навички використання Redis як швидкої, вбудованої у пам'ять бази даних, що може застосовуватись у реальних проєктах для кешування, обробки черг, тимчасового зберігання та керування сесіями.