

## Лабораторна робота №8

**З дисципліни:** Базы даних та інформаційні системи

**Студента групи МІТ-31:** Заяць Діани

**Тема:** Робота з Redis

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

**Хід роботи:**

### 1. Встановлення та запуск Redis

Для встановлення та запуску Redis було обрано використання Docker, що дозволило ізолювати Redis в контейнері.

**Команди для запуску Redis в Docker:**

```
PS C:\Users\Lenovo Legion> docker run --name redis-container -d -p 6379:6379 redis
Unable to find image 'redis:latest' locally
latest: Pulling from library/redis
c70d7dc4bd70: Pull complete
63df650ee4e0: Pull complete
c175c1c9487d: Pull complete
Status: Downloaded newer image for redis:latest
8c2822e63616e74ff56cf6e985b95d1ba562e00e8f64a496675916b38898a6ee
PS C:\Users\Lenovo Legion> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
NAMES
8c2822e63616   redis    "docker-entrypoint.s..." 12 seconds ago Up 11 seconds  0.0.0.0:6379->6379/tcp
redis-container
PS C:\Users\Lenovo Legion> docker exec -it redis-container redis-cli
127.0.0.1:6379> PING
PONG
```

### 2. Операції з рядками (Strings)

Після успішного запуску Redis, були виконані операції з рядками. Створено ключ зі значенням та перевірено, чи збережено значення в Redis:

**Команди:**

```
127.0.0.1:6379> SET student "Zayats Diana"
OK
127.0.0.1:6379> GET student
"Zayats Diana"
```

```
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. Операції зі списками (List)

Далі було продемонстровано використання списків. Було додано кілька елементів до списку та видалено перший елемент:

**Команди:**

```
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> LRANGE tasks 0 -1
1) "Task1"
127.0.0.1:6379>
```

### 4. Операції з множинами (Set)

Було виконано операції з множинами, додано елементи та перевірено наявність елементів у множині:

- **Команди:**

```
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
127.0.0.1:6379> █
```

## 5. Операції з хешами (Hash)

Далі було продемонстровано використання хешів. Було додано пару полів до хешу і отримано значення полів:

```
127.0.0.1:6379> HSET profile name "Zayats Diana"
(integer) 1
127.0.0.1:6379> HSET profile city "Kyiv"
(integer) 1
127.0.0.1:6379> HGET profile name
"Zayats Diana"
127.0.0.1:6379> HGETALL profile
1) "name"
2) "Zayats Diana"
3) "city"
4) "Kyiv"
127.0.0.1:6379>
```

## 6. Операції з відсортованими множинами (Sorted Set)

Було виконано операції з відсортованими множинами, додано кілька елементів і отримано елементи в зворотньому порядку:

**Команди:**

```
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
127.0.0.1:6379> █
```

## 7. Робота з TTL (Time to Live)

Створено ключ з обмеженим часом життя, після чого було перевірено час життя ключа та його значення:

```
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) 1
127.0.0.1:6379>
```

## 8. Додаткове завдання

Програма на Python, яка працює з Redis, була реалізована для зберігання та управління списком завдань. Програма дозволяє додавати, видаляти та переглядати завдання:

**Код програми:**

```
import redis

class TodoApp:
    def __init__(self):
        # Підключення до Redis
        self.r = redis.Redis(host='localhost', port=6379, db=0)
        self.key = 'tasks' # Ключ для списку завдань у Redis

    def show_tasks(self):
        # Отримуємо всі завдання
        tasks = self.r.lrange(self.key, 0, -1)
        if tasks:
            print("\nВаші завдання:")
            for idx, task in enumerate(tasks, start=1):
                print(f"{idx}. {task.decode('utf-8')}")
        else:
```

```

        print("Список завдань порожній.")

    def add_task(self, task):
        # Додаємо нове завдання
        self.r.lpush(self.key, task)
        print(f"\nЗавдання '{task}' додано.")

    def delete_task(self, task_index):
        # Видаляємо завдання за індексом
        tasks = self.r.lrange(self.key, 0, -1)
        if 0 < task_index <= len(tasks):
            task_to_delete = tasks[task_index - 1]
            self.r.lrem(self.key, 1, task_to_delete)
            print(f"\nЗавдання '{task_to_delete.decode('utf-8')}'
видалено.")
        else:
            print("\nНевірний індекс завдання.")

    def clear_tasks(self):
        # Очищаємо всі завдання
        self.r.delete(self.key)
        print("\nВсі завдання видалено.")

def main():
    app = TodoApp()
    while True:
        print("\nМеню:")
        print("1. Показати завдання")
        print("2. Додати завдання")
        print("3. Видалити завдання")
        print("4. Очистити список завдань")
        print("5. Вийти")

        choice = input("\nВиберіть опцію (1/2/3/4/5): ")

        if choice == '1':
            app.show_tasks()
        elif choice == '2':
            task = input("\nВведіть нове завдання: ")
            app.add_task(task)

```

```

        elif choice == '3':
            task_index = int(input("\nВведіть номер завдання для
видалення: "))
            app.delete_task(task_index)
        elif choice == '4':
            app.clear_tasks()
        elif choice == '5':
            print("\nДо побачення!")
            break
        else:
            print("\nНевірна опція. Спробуйте знову.")

if __name__ == "__main__":
    main()

```

Як працює:

Меню:

1. Показати завдання
2. Додати завдання
3. Видалити завдання
4. Очистити список завдань
5. Вийти

Виберіть опцію (1/2/3/4/5): 1

Ваші завдання:

1. Task1

1. Task1

Меню:

1. Показати завдання
2. Додати завдання
3. Видалити завдання
4. Очистити список завдань
5. Вийти

Виберіть опцію (1/2/3/4/5): 2

Введіть нове завдання: Task 2

Завдання 'Task 2' додано.

Меню:

1. Показати завдання
2. Додати завдання
3. Видалити завдання
4. Очистити список завдань
5. Вийти

Виберіть опцію (1/2/3/4/5): 3

Введіть номер завдання для видалення: 2

Завдання 'Task1' видалено.

Меню:

1. Показати завдання
2. Додати завдання
3. Видалити завдання
4. Очистити список завдань
5. Вийти
3. Видалити завдання
4. Очистити список завдань
5. Вийти
4. Очистити список завдань
5. Вийти
5. Вийти

Виберіть опцію (1/2/3/4/5): 4

Всі завдання видалено.