



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики  
Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 2**  
з дисципліни “Бази даних”

Виконав  
студент III курсу  
групи КП-81

Ладуда Данило Володимирович  
(прізвище, ім'я, по батькові)

варіант № 9

Зарахована  
“ \_\_\_\_ ” “ \_\_\_\_ ” 20\_\_ р.  
викладачем

Петрашенко Андрієм  
Васильовичем (прізвище, ім'я, по батькові)

*Мета роботи:* здобуття практичних навичок створення ефективних програм, орієнтованих на використання сервера Redis за допомогою мови Python.

*Завдання роботи* полягає у наступному: реалізувати можливості обміну повідомленнями між користувачами у оффлайн та онлайн режимах із можливістю фільтрації спам-повідомлень.

*Окремі програмні компоненти та вимоги до них*

1. Redis server (RS), що виконує наступні ролі:

1.1. *Сховище*, що містить: дані користувачів, їхні групи (звичайний користувач та адміністратор), а також повідомлення, що пересилаються між ними.

1.2. *Черга повідомлень*, які підлягають перевірці на спам та відправленню адресату.

1.3. Інструмент *Publish/Subscribe* для ведення та розсилання журналу активності користувачів (див. *Список активностей для журналювання*).

2. Інтерфейс користувача (User Interface)

2.1. *Звичайний користувач* має змогу виконувати вхід за ім'ям (без паролю), відправляти та отримувати (переглядати) повідомлення, отримувати дані про кількість **своїх** повідомлень, згрупованих за статусом (див. *Статуси повідомлень*).

2.2. *Адміністратор* має змогу переглядати журнал подій, що відбулись (див. *Список активностей для журналювання*), переглядати список користувачів, які знаходяться online, переглядати статистику (N найбільш активних відправників повідомлень із відповідною кількістю, N найактивніших “спамерів” із відповідною кількістю).

3. *Виконувач* (worker) призначений для:

перегляду черги повідомлень, відбору повідомлення, перевірки його вмісту на наявність спаму (у випадку наявності спаму -- додавання запису в журнал)

### *Інші вимоги*

1. Проаналізувавши матеріали ресурсів, наведений у пункті “Джерела”, обрати та обґрунтувати вибір структур даних Redis щодо реалізації наведених вище вимог, **обов’язково використати наступні структури даних** та інструменти Redis: List, Hash, Sorted List, Set, Pub/Sub.
2. Забезпечити роботу програмних засобів у режимі емуляції із можливістю генерації повідомлень від різних користувачів, налаштування кількості виконувачів та часу затримки обробки на спам з можливістю підключення адміністратора для перегляду подій, що відбуваються.
3. Перевірку на спам можна проемулювати за допомогою затримки на псевдовипадковий час та генерацію псевдовипадкового результату (Так/Ні).

### *Список активностей для журналювання*

Вхід/вихід користувача, наявність спаму у повідомленні.

### *Статуси повідомлень*

“Створено”, “У черзі”, “Перевіряється на спам”, “Заблоковано через спам”, “Відправлено адресату”, “Доставлено адресату”.

### *Вимоги до інтерфейсу користувача*

Використовувати консольний (текстовий) інтерфейс користувача.

Посилання на репозиторій: [https://github.com/ladudanil/db\\_2term](https://github.com/ladudanil/db_2term)

### *Обґрунтування вибору структур даних:*

Повідомлення зберігаються в хештаблиці, оскільки вона зберігає дані в парі ключ - значення. Це дозволє легко отримувати та обробляти повідомлення які відправили користувачі. В якості ключа виступає message\_id, а в якості значення структура, яка зберігає такі дані як текст, id відправника, id отримувача та статус повідомлення.

Користувачі зберігаються в множині, оскільки нам важлива швидкість.

Для черги повідомлень використовується List, так як нам є важливий порядок додавання та вилучення повідомлень. Також слід зауважити що в такому випадку час додавання та вилучення становитимме  $O(1)$ .

## Результати виконання завдання:

### Приклад взаємодії з головним меню (user.py)

```
===== MAIN MENU =====
1) Register
2) Log in
3) Exit
=> 1
Enter your username: arcanit33
Sorry, but user with this username: "arcanit33" already exists
===== MAIN MENU =====
1) Register
2) Log in
3) Exit
=> 2
Enter your username: asdsacx
Sorry, there is no registered user with username: asdsacx
===== MAIN MENU =====
1) Register
2) Log in
3) Exit
=> 1
Enter your username: Xasturr
===== MAIN MENU =====
1) Register
2) Log in
3) Exit
=> 2
Enter your username: Xasturr
===== USER MENU =====
1) Send a message
2) Received messages
3) Message statistics
4) Log out
=>
```

### Приклад взаємодії з меню користувача (user.py)

```
===== USER MENU =====
1) Send a message
2) Received messages
3) Message statistics
4) Log out
=> 1
Enter your message: HI
Enter username of receiver: Sasha
Message is waiting to be processed
===== USER MENU =====
1) Send a message
2) Received messages
3) Message statistics
4) Log out
=> 3
In queue: 1
Need to check: 0
Blocked for spam: 0
Sent to user: 0
Delivered to user: 0
===== USER MENU =====
1) Send a message
2) Received messages
3) Message statistics
4) Log out
=> 4
===== MAIN MENU =====
1) Register
2) Log in
3) Exit
=> 3
```

```

===== MAIN MENU =====
1) Register
2) Log in
3) Exit
=> 2
Enter your username: Sasha
===== USER MENU =====
1) Send a message
2) Received messages
3) Message statistics
4) Log out
=> 2
From: arcanit33 - hi Sasha

```

Приклад взаємодії з меню адміністратора (admin.py)

```

===== ADMIN MENU =====
1) Show all online users
2) Show N top senders
3) Show N top spammers
4) Show last N lines of log
5) Exit
=> 1
Users online:
Sasha
===== ADMIN MENU =====
1) Show all online users
2) Show N top senders
3) Show N top spammers
4) Show last N lines of log
5) Exit
=> 2
Enter number of top senders: 3
Top 3 senders:
user:Sasha - 3 messages
user:arcanit33 - 2 messages
user:Xasturr - 1 messages
===== ADMIN MENU =====
1) Show all online users
2) Show N top senders
3) Show N top spammers
4) Show last N lines of log
5) Exit
=> 3
Enter number of top spammers: 3
Top 3 spammers:
user:Sasha - 3 spam messages
===== ADMIN MENU =====
1) Show all online users
2) Show N top senders
3) Show N top spammers
4) Show last N lines of log
5) Exit
=> _

```

```
===== ADMIN MENU =====
1) Show all online users
2) Show N top senders
3) Show N top spammers
4) Show last N lines of log
5) Exit
=> 4
Write number of last rows to show: 4
Last 3 lines of logs:
INFO:root:2021-05-18 16:38:38.410826: Log out: Log out successful. "Xasturr" logged in

INFO:root:2021-05-18 16:39:44.249873: Log in: Log in successful. "Sasha" logged in

===== ADMIN MENU =====
1) Show all online users
2) Show N top senders
3) Show N top spammers
4) Show last N lines of log
5) Exit
=> 5
```

Приклад роботи обробника повідомлень (worker.py)

```
PS C:\Users\arcanit33\Desktop\db\Lab2\src> python .\worker.py
```

Приклад роботи симуляції (simulation.py)

```
alexanderwilson
christywilliams
christopher86
hawkinsmichele
ljones
Ctrl+C to exit
```

## Приклад збережених логів (events.log)

```
INFO:root:2021-05-18 16:46:29.669836: Log out: Log out successful. "Sasha" logged in
INFO:root:2021-05-18 16:46:39.577174: Log in: Log in successful. "Xasturr" logged in
INFO:root:2021-05-18 16:46:39.577174: Log in: Log in successful. "Xasturr" logged in
INFO:root:2021-05-18 16:46:57.765808: Spam check: user "Xasturr" sent spam message: "test"
INFO:root:2021-05-18 16:48:41.875108: Registration: Registration successful. "alexanderwilson" registered
INFO:root:2021-05-18 16:48:41.875108: Registration: Registration successful. "alexanderwilson" registered
INFO:root:2021-05-18 16:48:41.876109: Log in: Log in successful. "alexanderwilson" logged in
INFO:root:2021-05-18 16:48:41.876109: Log in: Log in successful. "alexanderwilson" logged in
INFO:root:2021-05-18 16:48:41.877106: Registration: Registration successful. "christywilliams" registered
INFO:root:2021-05-18 16:48:41.877106: Registration: Registration successful. "christywilliams" registered
INFO:root:2021-05-18 16:48:41.877106: Log in: Log in successful. "christywilliams" logged in
INFO:root:2021-05-18 16:48:41.877106: Log in: Log in successful. "christywilliams" logged in
```