**Домашня робота №66 Черга – структура даних**

**Завдання 1**

Необхідно реалізувати систему обробки завдань, яка буде приймати та виконувати задачі в заданому порядку. Кожна задача (як вузол) має номер *id*, назву *name* та свій пріоритет виконання *priority*. Задачі можуть бути додані до черги, очікувати на обробку та видалятись з черги по одній. Задачі повинні бути виконані (тобто видалятись) у порядку зростання пріоритету, при цьому задачі з однаковим пріоритетом повинні виконуватись у порядку проходження. У разі, якщо дві задачі мають однаковий пріоритет, то першою виконується та, яка прийшла раніше. Система повинна забезпечувати додавання нових задач до черги, видалення задач для їх виконання та виведення статистики про стан поточний стан черги та виконані завдання.

Основні операції для реалізації черги:

1. enqueue(id, name, priority): додати задачу до черги.
2. dequeue(): видалити та повернути задачу з черги із першим найвищим рівнем пріоритету.
3. is\_empty(): перевірити чи черга порожня.
4. size(): повернути поточну кількість задач у черзі.
5. completed tasks(): повернути поточну кількість виконаних задач

*Приклад*

|  |  |  |
| --- | --- | --- |
| **№ =id** | **Завдання** | **Пріоритет** |
| 1 | Підготувати звіт про продажі | 3 |
| 2 | Відправити заказ клієнту A | 1 |
| 3 | Сформувати презентацію для команди. | 3 |
| 4 | Зателефонувати постачальнику щодо поставки товару. | 2 |
| 5 | Відправити заказ клієнту B | 1 |
| 6 | Замовити нове обладнання для офісу. | 2 |

*1 – найвищий пріоритет, 2 – середній пріоритет, 3 – низький пріоритет*

У хвіст черги додаємо послідовно завдання від №1 до №6.

Задачі повинні виконатись (видалитись) із черги задачі у такій послідовності:

№2, №5, №4, №6, №3, №1

(це при умові, якщо в чергу більше завдань не було додано) .

