

*Поліщук В. С., здобувач 2 курсу спеціальності 122 Комп'ютерні науки, Потапова Н. А., канд. екон. наук, доцент, доцент кафедри інформаційних технологій*

## **ОСОБЛИВОСТІ ВИКОРИСТАННЯ СТРУКТУРИ ДАНИХ «ЧЕРГА» У ПРОГРАМУВАННІ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЯХ**

*Донецький національний університет імені Василя Стуса, м. Вінниця*

У світі комп'ютерних технологій черга – це важлива концепція, яка допомагає у впорядкуванні та управлінні набором елементів. Вона використовується для зберігання та обробки даних у такий спосіб, що перший елемент, який додається до черги, також є першим, що видаляється з неї. Цей принцип, відомий як «першим прийшов – першим пішов (FIFO)», дає змогу ефективно використовувати чергу в різних областях інформатики, включно з операційними системами, алгоритмами обробки даних та ін. [1].

Черги відіграють важливу роль у багатьох аспектах програмування, від управління завданнями до обробки подій. Вони є важливим інструментом для створення ефективних та надійних програм, які працюють із послідовністю дій або подій. Черги можуть мати різні реалізації, як-от черги на базі масивів або черги на базі зв'язних списків, і вони знаходять своє застосування в різних областях інформатики, від розробки програмного забезпечення до оптимізації роботи систем [2].

Черги також можна знайти в реальних сценаріях за межами цифрового світу. Уявіть, наприклад, чергу людей, які чекають на посадку в автобус. Перша людина, яка прибуває на зупинку, першою сідає в автобус, коли він прибуває. Аналогічна ситуація і з квитковою системою: хто перший у черзі, той обслуговується першим [3].

Розуміння того, як працюють черги та їх застосування, має вирішальне значення в інформатиці та розробці програмного забезпечення. Це дає змогу програмістам розробляти ефективні алгоритми та системи, які можуть виконувати завдання структуровано та організовано. Отже, наступного разу, коли ви стоятимете в черзі або працюватимете з комп'ютерною програмою, подумайте про концепцію черг і про те, як вони допомагають операціям працювати безперебійно [4]. А тепер давайте розглянемо на практиці чергу.

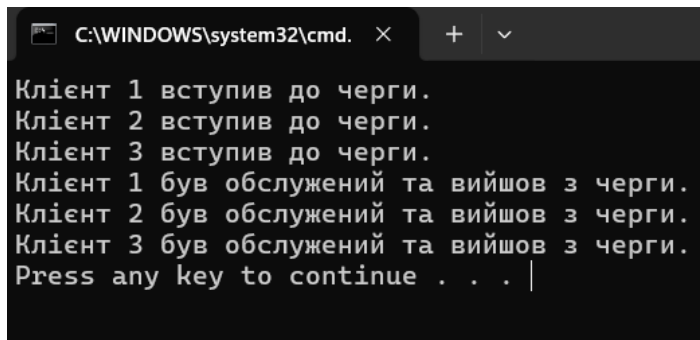
Програма нижче моделює роботу банківської черги, де клієнти заходять у чергу та обслуговуються у порядку їхнього приходу. Використовуючи мову програмування C#, код демонструє структуру черги, реалізованої за допомогою стандартного класу «Queue», а також механізм додавання клієнтів до черги та їх обслуговування.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using System.Threading;
5
6 class Program
7 {
8     0 references
9     static void Main()
10    {
11        // Встановлюємо кодування UTF-8 для консолі
12        Console.OutputEncoding = Encoding.UTF8;
13
14        // Створюємо чергу для банку
15        var bankQueue = new BankQueue();
16
17        // Додаємо клієнтів до черги
18        bankQueue.Enqueue("Клієнт 1");
19        bankQueue.Enqueue("Клієнт 2");
20        bankQueue.Enqueue("Клієнт 3");
21
22        // Обслуговуємо клієнтів та виводимо чергу
23        Thread.Sleep(2000); // Симулюємо час обслуговування
24        while (bankQueue.Count > 0)
25        {
26            Console.WriteLine($"{bankQueue.Dequeue()} був обслужений та вийшов з черги.");
27            Thread.Sleep(1000); // Симулюємо час обслуговування
28        }
29    }
30 }
```

Рис. 1. Код роботи банківської черги

```
29 }
30
31 1 reference
32 class BankQueue
33 {
34     private Queue<string> queue = new Queue<string>();
35
36     1 reference
37     public int Count => queue.Count;
38
39     3 references
40     public void Enqueue(string customer)
41     {
42         queue.Enqueue(customer);
43         Console.WriteLine($"{customer} вступив до черги.");
44     }
45
46     1 reference
47     public string Dequeue()
48     {
49         if (queue.Count > 0)
50             return queue.Dequeue();
51         else
52         {
53             Console.WriteLine("Черга порожня.");
54             return null;
55         }
56     }
57 }
```

Рис. 2. Продовження коду



```
C:\WINDOWS\system32\cmd.  ×  +  ▾
Клієнт 1 вступив до черги.
Клієнт 2 вступив до черги.
Клієнт 3 вступив до черги.
Клієнт 1 був обслужений та вийшов з черги.
Клієнт 2 був обслужений та вийшов з черги.
Клієнт 3 був обслужений та вийшов з черги.
Press any key to continue . . . |
```

Рис. 3. Результат коду

Цей код пропонує просту реалізацію черги у мові програмування C#, яка дає змогу ефективно додавати клієнтів до черги, обслуговувати їх у порядку черги та виводити повідомлення про їх обслуговування. Черги відіграють важливу роль у програмуванні та інформаційних технологіях, забезпечуючи правильне упорядкування та обробку даних у процесі виконання програм. Вони знаходять застосування у широкому спектрі задач, від управління завданнями до оптимізації роботи систем, завдяки принципу «першим прийшов – першим обслужований», який є ключовим для багатьох алгоритмів та програмних рішень.

#### Список використаних джерел

1. Черга. Способи реалізації черги. Представлення черги у вигляді динамічного масиву. *BestProg*: вебсайт. 2019. URL: <https://www.bestprog.net/uk/2019/09/26/c-queue-general-concepts-ways-to-implement-the-queue-implementing-a-queue-as-a-dynamic-array-ua/> (дата звернення: 30.04.2024).
2. Жмурко О. О. Урок: Роль інформаційних технологій у житті сучасної людини. *Vseosvita*: вебсайт. 2024. URL: <https://vseosvita.ua/lesson/rol-informatsiinykh-tekhnologii-u-zhyttisuchasnoi-liudyny-229384.html> (дата звернення 30.04.2024).
3. Основи інформатики та технологій програмування: навчальний посібник / М. Є. Рогоза, С. К. Рамазанов, А. В. Велігура, С. М. Танченко. Луганськ: Вид-во СНУ ім. В. Даля, 2012. 568 с.
4. Програмна реалізація та дослідження алгоритмів паралельного швидкого сортування / В. О. Денисюк, Н. А. Потапова, О. В. Зелінська, М. Б. Тарасюк. *Вісник Хмельницького національного університету. Технічні науки*. 2023. № 4. С. 95–105. URL: [http://journals.khnu.km.ua/vestnik/?page\\_id=41](http://journals.khnu.km.ua/vestnik/?page_id=41)