**Державний вищий навчальний заклад  
Ужгородський національний університет  
Факультет інформаційних технологій**

ПРАКТИЧНА РОБОТА № 1  
**Тема:** Стек та черга.

Виконав студент  
І курсу спеціальності «Інженерія програмного забезпечення»

Романюк Артем

**Ужгород-2025**

**Мета:** набути практичних навичок створення та застосування структури даних стек і черга та основні принципи роботи з ними.

Завдання до роботи:

1. Створити динамічну структуру згідно з варіантом таблиці 1.1. У програмі повинні бути передбачені наступні функції: «Додавання елемента»; «Видалення елемента»; «Перегляд»; «Очистка». Повинні бути передбачені аварійні ситуації (наприклад: не можна видалити елемент, якщо стек порожній).
2. Оформити звіт про виконання роботи та завантажити його в системі електронного навчання ДВНЗ «УжНУ» в установлений термін.
3. Підготувати відповіді на питання до лекції про стеки та черги.

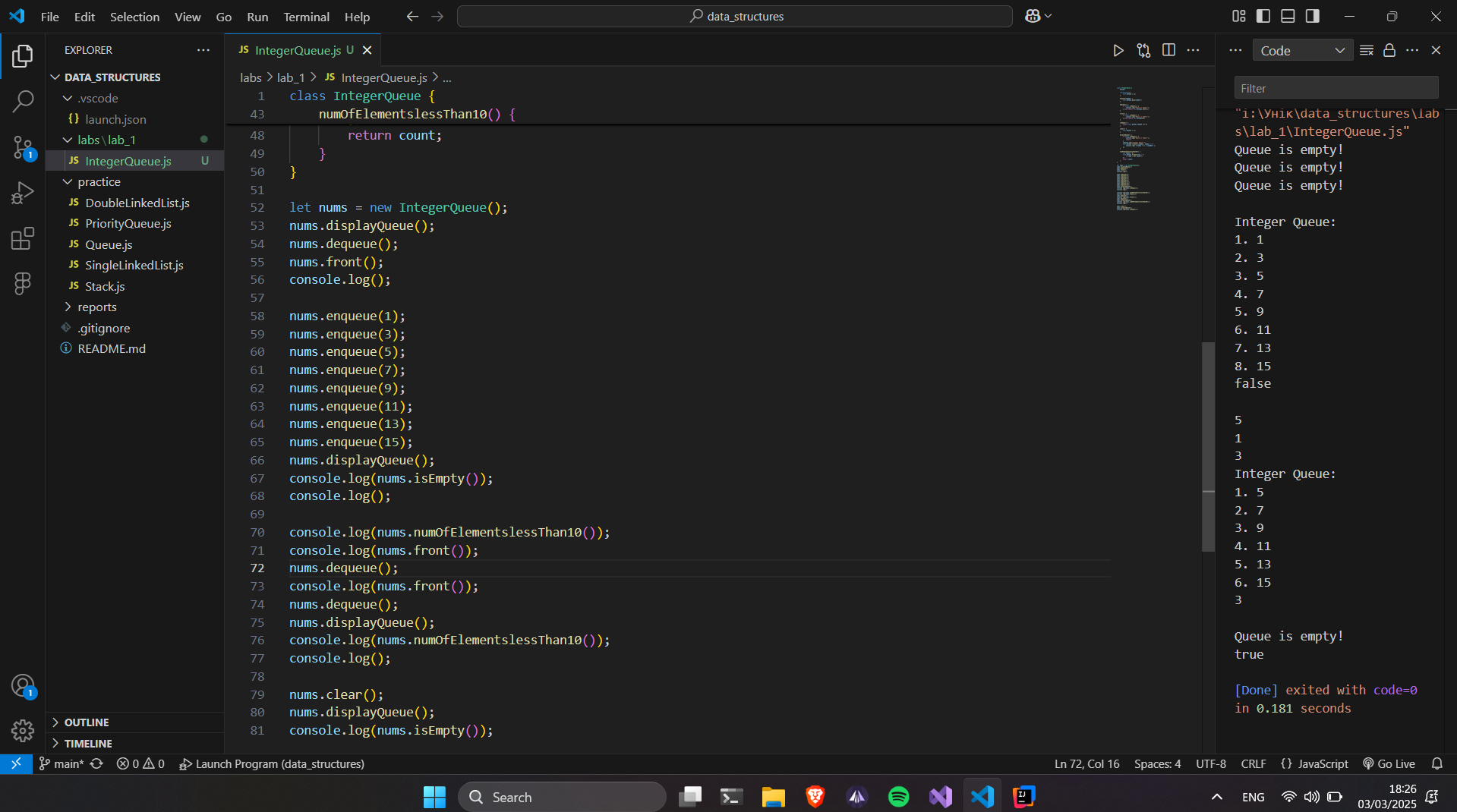
**Хід роботи:**

**Варіант 27**

1. Створити чергу з цілих чисел. Визначити кількість елементів черги менше 10. Організувати перегляд даних черги

[Реалізація в коді (github):](https://github.com/ArtemRomaniuk/data_structures/blob/main/labs/lab_1/IntegerQueue.js)

| class IntegerQueue {  #items;  constructor() {  this.#items = [];  }  enqueue(number) {  this.#items.push(number);  }  dequeue() {  if (this.isEmpty()) {  console.log("Queue is empty!");  } else return this.#items.shift();  }  front() {  if (this.isEmpty()) {  console.log("Queue is empty!");  } else return this.#items[0];  }  isEmpty() {  return this.#items.length === 0;  }  clear() {  this.#items = [];  }  displayQueue() {  if (this.isEmpty()) {  console.log("Queue is empty!");  return;  }  console.log("Integer Queue:");  this.#items.forEach((number, index) => {  console.log(`${index + 1}. ${number}`);  })  }  numOfElementslessThan10() {  let count = 0;  this.#items.forEach(num => {  if (num < 10) count++;  })  return count;  }  }  let nums = new IntegerQueue();  nums.displayQueue();  nums.dequeue();  nums.front();  console.log();  nums.enqueue(1);  nums.enqueue(3);  nums.enqueue(5);  nums.enqueue(7);  nums.enqueue(9);  nums.enqueue(11);  nums.enqueue(13);  nums.enqueue(15);  nums.displayQueue();  console.log(nums.isEmpty());  console.log();  console.log(nums.numOfElementslessThan10());  console.log(nums.front());  nums.dequeue();  console.log(nums.front());  nums.dequeue();  nums.displayQueue();  console.log(nums.numOfElementslessThan10());  console.log();  nums.clear();  nums.displayQueue();  console.log(nums.isEmpty()); |
| --- |



**Висновок:**

У ході виконання практичної роботи ми дослідили структури даних стек та черга, їхні принципи роботи, реалізацію та застосування.

Стек працює за принципом LIFO (Last In, First Out – останнім зайшов, першим вийшов). Це означає, що елементи додаються та видаляються лише з одного кінця – вершини стека. Такий підхід ефективно використовується в рекурсивних алгоритмах.

Черга діє за принципом FIFO (First In, First Out – першим зайшов, першим вийшов), де елементи додаються в один кінець (задню частину черги) і видаляються з іншого (передньої частини). Черги широко застосовуються в системах планування завдань та обробці потоків даних.

Практична реалізація цих структур допомогла краще зрозуміти їхню важливість і особливості використання в реальних завданнях програмування.