**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»

Боднарчук Роман Романович

**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 2**

1. Створити чергу із дійсних чисел. Визначити кількість додатних значень елементів черги

class Queue {

    constructor() {

      this.queue = [];

    }

    enqueue(item) {

      this.queue.push(parseFloat(item));

    }

    shift() {

      return this.queue.shift();

    }

    show(){

      console.log(this.queue.join(' '))

    }

    clear(){

      this.queue = []

    }

    showPositive (){

      return this.queue.filter(e=>e>0).length

    }

  }

  const queue = new Queue();

  for(let i = 0;i<10;i++){

    queue.enqueue((Math.random() \*10 -   5 ).toFixed(2))

  }

  queue.show()

  console.log(queue.showPositive())

  queue.clear()

  queue.show()

**Питання для самоконтролю**

1. У чому переваги і недоліки організації структур у вигляді стека?

**Відповідь:**

**Переваги стеку:**

-Простота в реалізації і використанні.

-Ефективність для задач, де потрібно використовувати дані у порядку "останнім -прийшов - першим вийшов" (LIFO).

-Можливість рекурсивного використання.

**Недоліки стеку:**

-Обмеженість розміру, що може призвести до переповнення (stack overflow).

-Неефективність для задач, де потрібно доступатися до середини даних.

-Відсутність ітерації (проходу) по всіх елементах.

1. У чому переваги і недоліки організації структур у вигляді черги?

**Відповідь:**

**Переваги черги:**

-Ефективність для задач, де обробка даних відбувається за принципом "перший прийшов - перший вийшов" (FIFO).

-Підтримка паралельних операцій додавання і видалення елементів.

-Використання у алгоритмах обробки даних та управління задачами.

**Недоліки черги:**

-Обмеженість розміру, можливість переповнення.

-Неефективність для задач, де потрібен доступ до середини даних.

-Відсутність можливості вставки елементів у середину черги без видалення інших елементів.

1. Для моделювання яких реальних завдань зручно використовувати стек? А для яких чергу?

**Відповідь:**

**Стек підходить для моделювання таких завдань:**

-Відстеження викликів функцій у рекурсивних алгоритмах.

-Управління відкритими діалоговими вікнами у програмах.

-Відміна дій у текстовому редакторі.

-Історія перегляду веб-сторінок у веб-браузерах.

**Черга підходить для моделювання таких завдань:**

-Керування задачами на обробку в комп'ютерних системах.

-Очікування у черзі на обслуговування (наприклад, у банку або у касі супермаркету).

-Обробка запитів у мережах передачі даних.

-Управління відправленням та отриманням повідомлень у мережевих протоколах.

1. Яке значення зберігає покажчик на стек?

**Відповідь:**

Покажчик на стек зберігає адресу пам'яті останнього доданого елемента стеку. Коли додається новий елемент, покажчик зсувається вгору, вказуючи на новий елемент. При видаленні елемента покажчик зсувається вниз, вказуючи на попередній елемент у стеці.

1. Яке значення зберігає покажчик на чергу?

**Відповідь:**

Покажчик на чергу зберігає адресу пам'яті першого елемента в черзі. При додаванні нового елемента покажчик залишається на місці, але змінюється адреса останнього елемента. При видаленні елемента покажчик зсувається до наступного елемента в черзі.

1. Які існують обмеження на тип інформаційного поля стека і черги?

**Відповідь:**

Обмеження на тип інформаційного поля стека і черги зазвичай визначається типом даних, які можна зберігати в них. Зазвичай це можуть бути будь-які примітивні типи даних (наприклад, цілі числа, дійсні числа, символи) або вказівники на складні типи даних (наприклад, вказівники на структури або класи). Однак, обмеження можуть відрізнятися в залежності від мови програмування або бібліотеки, що використовується

1. З якою метою в програмах виконується перевірка на порожнину стека і

черги?

**Відповідь:**

Перевірка на порожнину стека і черги виконується для визначення, чи містить структура даних жодного елемента. Це важливо для уникнення помилок при спробі доступу до неіснуючих елементів. Наприклад, перевірка на порожнину допомагає уникнути вилучення елементів з порожньої черги або стеку, що може призвести до помилки в програмі.

1. При роботі зі стеком або чергою доступні позиції обмеженого числа елементів. Чи можлива ситуація запису нових елементів стека або черги на вже зайняті власними елементами ділянки пам'яті? Відповідь обґрунтуйте.

**Відповідь:**

Ні, така ситуація неможлива. Робота зі стеком або чергою передбачає доступ лише до кінця структури даних (вершини стека або початку черги) для додавання нових елементів або видалення існуючих. Нові елементи додаються лише відповідно до логіки "останнім прийшов - першим вийшов" (стек) або "першим прийшов - першим вийшов" (черга), тобто не можна записати новий елемент на вже заїняту ділянку пам'яті, яка належить іншому елементу.

1. З якою метою в програмах виконується видалення стека і черги після закінчення роботи з ними? Як зміниться робота програми, якщо операцію видалення не виконувати?

**Відповідь:**

Видалення стека і черги після закінчення роботи з ними виконується для звільнення пам'яті, яку вони займають. Якщо не виконувати операцію видалення, програма може продовжувати використовувати цю пам'ять, навіть якщо структура даних більше не потрібна. Це може призвести до зайвого використання пам'яті і, в кінцевому рахунку, до вичерпання ресурсів пам'яті в системі.

**Висновки:** У ході виконання Практичної роботи №1, я успішно освоїв роботу із стеками та чергами на мові js. Я створив чергу із дійсних чисел. Визначив кількість додатніх значень елементів черги. Заповнив чергу через метод Math.Random і тим самим кожен запуск програми дає нам рандомні значення.