**Державний вищий навчальний заклад**

**Ужгородський національний університет**

**Факультет інформаційних технологій**

**Лабораторна робота №2**

**Односпрямовані та двоспрямовані списки**

Виконав студент 1 курсу

спеціальності “Інженерія

програмного забезпечення”

Дєлов В’ячеслав Едуардович

**Ужгород-2025**

**Мета:** набути практичних навичок створення та застосування односпрямованих та двоспрямованих списків.

Завдання до роботи:

1. Написати програму, у якій потрібно створити лінійний односпрямований список та виконати дії згідно з власним варіантом завдання табл. 1.1.
2. Написати програму, у якій потрібно створити двоспрямований список та виконати дії згідно з власним варіантом завдання табл 1.2.
3. Підготувати відповіді на питання до лекції з односпрямованого та двоспрямованого списку.

**Завдання 1.1:**

**Варіант 3**Створити лінійний односпрямований список з дійсних чисел. Видалити зі списку перший елемент менший за модулем 5

**Хід роботи:**1. Cтворено односпрямований список з функцією пошуку та видалення першого елемента, значення якого за модулем менше 5. Код програми (текстом):

class Node {

    constructor(value) {

      this.value = value;

      this.next = null;

    }

}

class LinkedList {

    constructor() {

      this.head = null;

    }

    append(value) {

      const newNode = new Node(value);

      if (!this.head) {

        this.head = newNode;

        return;

      }

      let current = this.head;

      while (current.next) {

        current = current.next;

      }

      current.next = newNode;

    }

    removeFirstLessThanFive() {

      if (!this.head) return; //якщо порожній

      if (Math.abs(this.head.value) < 5) {

        this.head = this.head.next;

        return;

      }

      let current = this.head;

      while (current.next) {

        if (Math.abs(current.next.value) < 5) {

          current.next = current.next.next;

          return;

        }

        current = current.next;

      }

      console.log("Елемент не знайдено!");

    }

    print() {

      let current = this.head;

      let result = "";

      while (current) {

        result += current.value + " -> ";

        current = current.next;

      }

      console.log(result + "null");

    }

}

Тест роботи:

const list = new LinkedList();

list.append(10);

list.append(3);

list.append(7);

list.append(-2);

list.append(15);

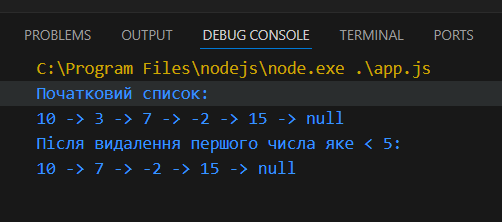
console.log("Початковий список:");

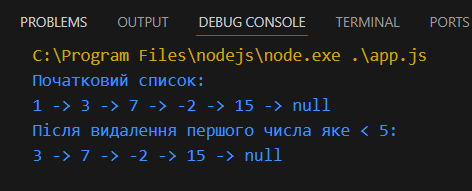
list.print();

list.removeFirstLessThanFive();

console.log("Після видалення першого числа яке < 5:");

list.print();

Результат:  
  
  
Тест роботи якщо елемент, який треба видалити буде головою списку:



Результат роботи правильний.

**Завдання 1.2**

**Варіант 3**Створити двоспрямований список із цілих чисел. Видалити зі списку останній елемент, який має значення менше 15.

2. Створено двоспрямований список із цілих чисел. Реалізовано функцію видалення останнього елемента < 15. (Перебір списку починається з кінця)

class Node {

    constructor(value) {

      this.value = value;

      this.next = null;

      this.prev = null;

    }

}

class DoublyLinkedList {

    constructor() {

      this.head = null;

      this.tail = null;

    }

    append(value) {

      const newNode = new Node(value);

      if (!this.head) {

        this.head = this.tail = newNode;

        return;

      }

      this.tail.next = newNode;

      newNode.prev = this.tail;

      this.tail = newNode;

    }

    removeLastLessThan15() {

      let current = this.tail;

      while (current) {

        if (current.value < 15) {

          if (current === this.head && current === this.tail) { //для одного ел.

            this.head = this.tail = null;

          } else if (current === this.tail) { //для остаанього

            this.tail = current.prev;

            this.tail.next = null;

          } else if (current === this.head) { //для першого

            this.head = current.next;

            this.head.prev = null;

          } else {

            current.prev.next = current.next;

            current.next.prev = current.prev;

          }

          return;

        }

        current = current.prev;

      }

    }

    print() {

      let current = this.head;

      let result = "";

      while (current) {

        result += current.value + " <-> ";

        current = current.next;

      }

      console.log(result + "null");

    }

}

Перевірка роботи тестовими значеннями:

const list = new DoublyLinkedList();

list.append(20);

list.append(10);

list.append(5);

list.append(15);

list.append(30);

console.log("Початковий список:");

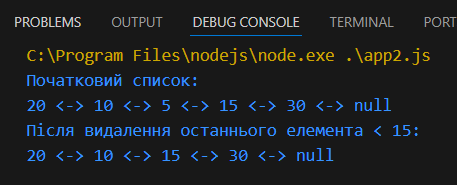
list.print();

list.removeLastLessThan15();

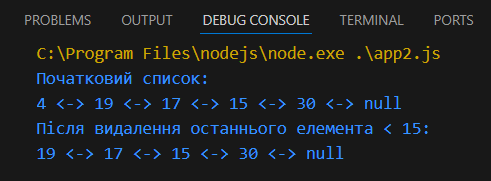
console.log("Після видалення останнього елемента < 15:");

list.print();

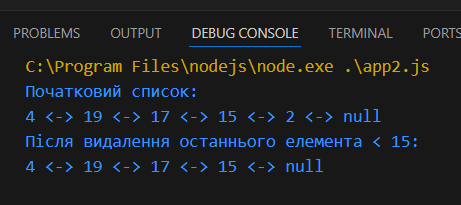
Результат виконання (елемент по-середині):



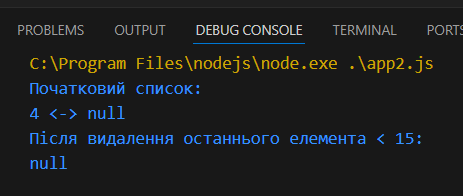
Елемент на початку:



Елемент в кінці:



Єдиний елемент:



Код програм запушено на гіт-хаб в репозиторій https://github.com/DelovSlava/Algoritmi\_-\_Strutures

**Висновок:**

У ході виконання роботи було розглянуто реалізацію **однозв’язного** та **двозв’язного** списків.

**Однозв’язний список** реалізований для роботи з дійсними числами, включаючи **видалення першого елемента, абсолютне значення якого менше 5.**

**Двозв’язний список** реалізовано для роботи з цілими числами, а також **видалення останнього елемента, значення якого менше 15.** Використання prev та tail дозволило здійснити **ефективний пошук у зворотному напрямку**, що є перевагою перед однозв’язним списком.