**Державний вищий навчальний заклад**

**Ужгородський національний університет**

**Факультет інформаційних технологій**

**ЛАБОРАТОРНА РОБОТА № 1**

**Тема:** **Стек та черга**

Виконала студентка 1 курсу спеціальності «Інженерія програмного забезпечення» Росада Марина Іванівна

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 28**

Створити стек, інформаційними полями якого є: прізвище робітника та його оклад. Додати до стека відомості про нового робітника. Організувати перегляд даних та обчислити середній оклад.

**Код програми:**

class EmployeeStack {

constructor() {

this.items = [];

}

push(surname, salary) {

this.items.push({ surname, salary });

}

pop() {

if (this.isEmpty()) {

throw new Error("Стек порожній!");

}

this.items.pop();

}

isEmpty() {

return this.items.length === 0;

}

view() {

if (this.isEmpty()) {

console.log("Стек порожній!");

return;

}

console.log("--- Працівники ---");

for (let i = this.items.length - 1; i >= 0; i--) {

console.log(`Прізвище: ${this.items[i].surname}, Оклад: ${this.items[i].salary}`);

}

console.log("-------------------");

}

peek() {

if (this.isEmpty()) {

return "Стек порожній";

}

console.log(`Прізвище: ${this.items[this.items.length - 1].surname}, Оклад: ${this.items[this.items.length - 1].salary}`);

return;

}

averageSalary() {

if (this.isEmpty()) return 0;

const totalSalary = this.items.reduce((sum, employee) => sum + employee.salary, 0);

return totalSalary / this.items.length;

}

clear() {

this.items = [];

}

}

//приклад використання

const stack = new EmployeeStack();

stack.push("Шевченко", 19000);

stack.push("Безгрішна", 17000);

stack.push("Мельник", 18000);

stack.push("Ковальчук", 20000);

stack.push("Вишнивецька", 21000);

stack.view();

console.log("Середній оклад:", stack.averageSalary());

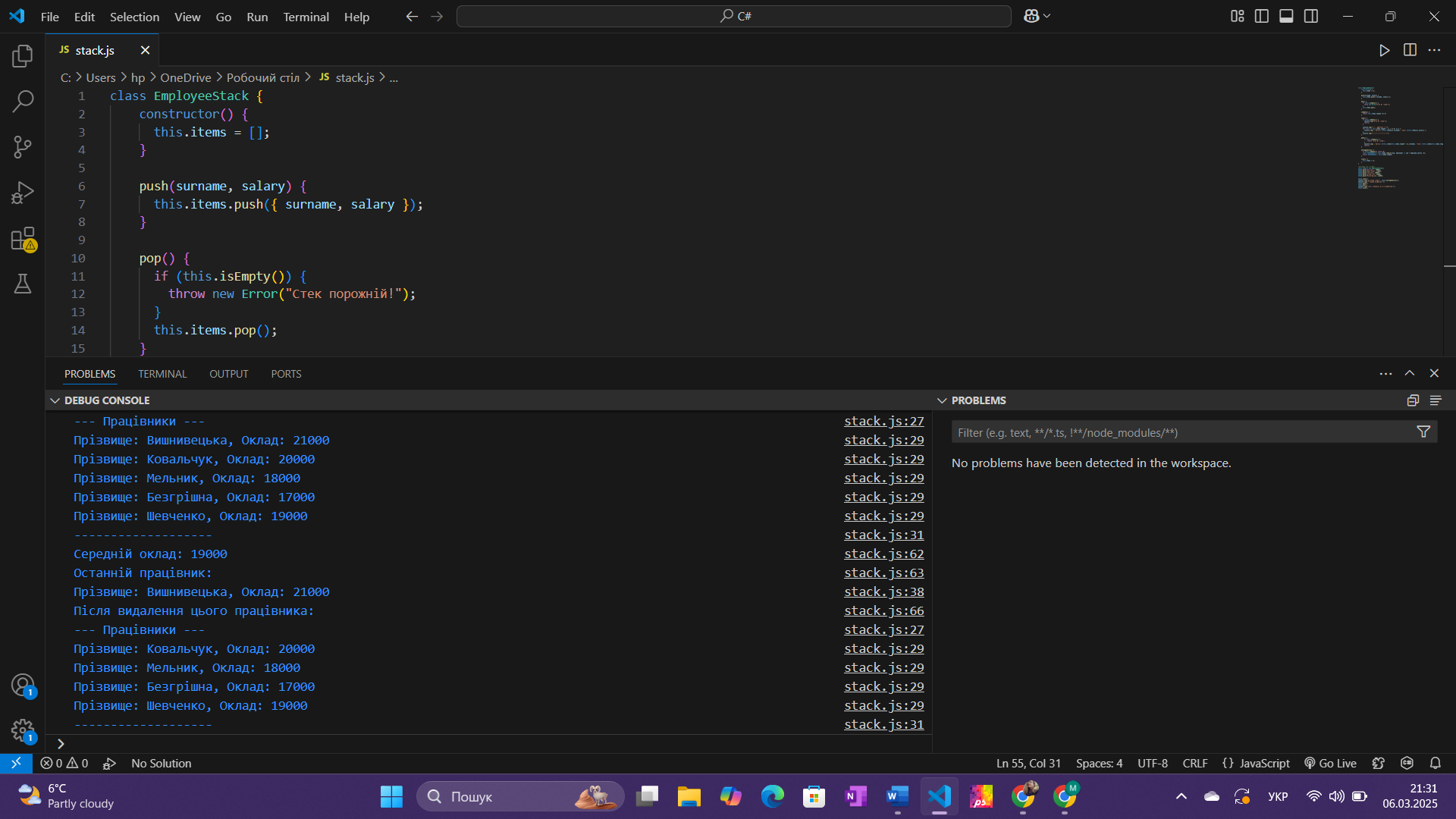
console.log("Останній працівник:");

stack.peek();

stack.pop();

console.log("Після видалення цього працівника:");

stack.view();

**Скріншот роботи програми:**

**Висновок:**

У ході виконання лабораторної роботи було розглянуто та реалізовано динамічну структуру даних – стек, інформаційними полями якого є прізвище робітника та його оклад. Було створено відповідні функції для додавання нового робітника до стека, перегляду його вмісту, а також обчислення середнього окладу всіх робітників, що знаходяться в стеку.

У результаті роботи було успішно створено та протестовано стек, реалізовано всі необхідні функції та перевірено їхню коректність. Було визначено середній оклад робітників, що містяться у структурі.

Під час виконання роботи було проаналізовано основні переваги та недоліки використання стека. Основними перевагами є простота реалізації, швидкий доступ до останнього доданого елемента та ефективність використання пам’яті при динамічному розширенні структури. Недоліками є обмежений доступ до елементів (тільки через вершину) та можливе перевитрачання пам’яті у випадку некоректного керування видаленням елементів.