**Державний вищий навчальний заклад**

**Ужгородський національний університет**

**Факультет інформаційних технологій**

**Практична робота № 8**

**Тема:** Хеш-таблиці.

Виконав студент І курсу

Спеціальності «Інженерія

програмного забезпечення»

Боднарчук Роман Романович

**Ужгород-2025**

**Мета:** познайомитися з хеш-функціями і хеш-таблицями та отримати навички програмування алгоритмів, що їх обробляють.

**Завдання до роботи:**

1. Написати код програми, яка реалізує роботу з хеш-таблицею, відповідно до обраного варіанту. Програма повинна підтримувати мінімальні методи хеш-таблиць: додавання, видалення, пошук. Алгоритм хешування ключа студент обирає самостійно.
2. Оформити звіт та завантажити його в системі електронного навчання ДВНЗ «УжНУ» в установлений термін.
3. Підготувати відповіді на контрольні питання.

# Хід роботи

**Варіант 2**

Створити хеш-таблицю для підрахунку кількості повторюваних слів у тексті. Розділіть текст на слова, додайте кожне слово до хеш-таблиці та виведіть кількість повторів кожного слова.

function HashTable() {

  this.table = {};

  this.add = function(word) {

    const lowerCaseWord = word.toLowerCase();

    if (!this.table[lowerCaseWord]) {

      this.table[lowerCaseWord] = 0;

    }

    this.table[lowerCaseWord] += 1;

  };

  this.remove = function(word) {

    const lowerCaseWord = word.toLowerCase();

    if (this.table[lowerCaseWord]) {

      this.table[lowerCaseWord] -= 1;

      if (this.table[lowerCaseWord] === 0) {

        delete this.table[lowerCaseWord];

      }

    }

  };

  this.search = function(word) {

    const lowerCaseWord = word.toLowerCase();

    return this.table[lowerCaseWord] || 0;

  };

}

function countWords(text) {

  const words = text.match(/[а-яіїєґa-z]+/gi);

  const hashTable = new HashTable();

  words.forEach((word) => {

    hashTable.add(word);

  });

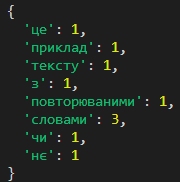
  return hashTable.table;

}

const text = "Це приклад тексту з повторюваними словами словами словами, чи нє?)";

const wordCounts = countWords(text);

console.log(wordCounts);



**Питання для самоконтролю**

1. Що таке асоціативний масив?

**Відповідь:** Асоціативний масив (також відомий як словник, словниковий масив або асоціативний масив ключ-значення) — це структура даних, що забезпечує відображення ключів на значення. У цій структурі доступ до значення здійснюється за допомогою ключа, а не за індексом, як у звичайному масиві. Ключі можуть бути рядками, числами або іншими об'єктами. Асоціативні масиви зазвичай використовуються для зберігання пар ключ-значення, де ключ відображається на відповідне значення.

1. Яка структура даних називається хеш-таблицею? Які основні операції вона має підтримувати?

**Відповідь:** Хеш-таблиця - це структура даних, що використовує хеш-функцію для відображення ключів на їх значення у масиві. Основні операції, які підтримує хеш-таблиця, включають вставку (додавання пари ключ-значення), вилучення (видалення пари ключ-значення) та пошук (знаходження значення за ключем).

1. У чому полягає процес розв'язання колізій?

**Відповідь:** Процес розв'язання колізій в хеш-таблиці полягає у вирішенні ситуацій, коли два або більше ключів вказують на одне і те ж місце у таблиці (колізія). Цей процес може бути реалізований за допомогою різних методів, таких як метод ланцюжків (зберігання всіх значень, що вказують на одне місце у таблиці, у вигляді списку) або метод відкритого хешування (пошук нового вільного місця для зберігання значення).

1. Які види хеш-таблиць бувають? Які відмінності між ними?

**Відповідь:** Існує кілька видів хеш-таблиць, основні з них:

* **Хеш-таблиці з відкритим хешуванням -** вони намагаються знайти інший вільний слот для зберігання колізійного ключа.
* **Хеш-таблиці зі зберіганням посилань -** вони використовують додаткову пам'ять для зберігання посилань на об'єкти, які зберігають дані.
* **Хеш-таблиці з використанням ланцюжків -** вони використовують масив списків для зберігання даних, що мають однаковий хеш.

Відмінності між ними полягають у способі вирішення колізій та управління пам'яттю. Хеш-таблиці з відкритим хешуванням намагаються знайти нове місце для зберігання даних, коли відбувається колізія. Хеш-таблиці зі зберіганням посилань використовують додаткову пам'ять для зберігання посилань на дійсні об'єкти. Хеш-таблиці з використанням ланцюжків використовують масив списків для зберігання даних, що мають однаковий хеш.

1. Яким чином реалізується процес видалення ключів у хеш-таблиці з відкритою адресацією?

**Відповідь:** У хеш-таблиці з відкритою адресацією процес видалення ключа може бути реалізований шляхом позначення слоту, який містить видаляєме значення, як вільний. Це означає, що в цьому слоті не міститься жодного значення. Після видалення значення зазвичай не видаляється фізично, а просто позначається як видалене. При подальших операціях додавання або пошуку цей слот може бути використаний для зберігання нового значення.

1. Поясніть концепцію методу ланцюжків для розв’язання колізій при реалізації хеш-таблиць.

**Відповідь:** Метод ланцюжків для розв'язання колізій в хеш-таблиці полягає в тому, що кожна ячейка хеш-таблиці представляє собою посилання на список (ланцюжок) ключів, які мають однаковий хеш-код. Коли відбувається колізія (тобто два ключі мають однаковий хеш-код), новий ключ додається до відповідного ланцюжка. При пошуку ключа спочатку обчислюється хеш-код, потім перевіряється ланцюжок, що відповідає цьому хеш-коду, на наявність шуканого ключа.

**Висновки:** У ході виконання Практичної роботи №8, я написав код програми, яка реалізує роботу з хеш-таблицею. Хеш-таблиці - це ефективна структура даних, яка використовує хеш-функцію для швидкого доступу до даних за ключем. Вони підтримують операції вставки, вилучення та пошуку за константний час в середньому (O(1)). Хеш-таблиці можуть вирішувати колізії за допомогою методів відкритої адресації або ланцюжків. Тим самим набув практичних навичок із виконанням Хеш-таблиць та зробив висновок.