Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

По дисциплине: «Языковые процессы интеллектуальных систем»

# Тема: «Хеш-таблицы»

Выполнил:

Студент 4 курса

Группы ИИ-21

Литвинюк Т. В.

Проверил:

Монтик Н.С.

Брест 2024

**Цель работы**: изучить основные методы организации хеш-таблиц, получить представление о преимуществах и недостатках, присущих различным методам организации таблиц.

3. Таблица упорядочивается в обратном алфавитном порядке, при этом все буквы преобразуются в заглавные. Поиск - логарифмический с подсчетом числа сравнений. import bisect

class HashTable:

    def \_\_init\_\_(self):

        self.table = []

    def insert(self, identifier):

        identifier = identifier.upper()

        table = self.table[::-1]

        bisect.insort(table, identifier)

        self.table = table[::-1]

    def search(self, identifier):

        identifier = identifier.upper()

        comparisons = 0

        low, high = 0, len(self.table) - 1

        while low <= high:

            comparisons += 1

            mid = (low + high) // 2

            if self.table[mid] == identifier:

                return True, comparisons

            elif self.table[mid] < identifier:

                high = mid - 1

            else:

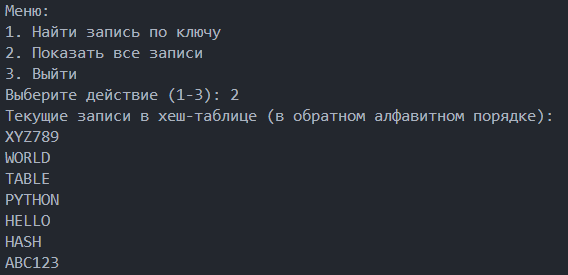
                low = mid + 1

        return False, comparisons

    def show(self):

        print(\*self.table, "\n", sep="\n")

def read\_identifiers\_from\_file(file\_path):

****    with open(file\_path, 'r') as file:

        return [line.strip() for line in file]

def main():

    file\_path = "APIS/lab1/ident.txt"

    identifiers = read\_identifiers\_from\_file(file\_path)

    hash\_table = HashTable()

    for identifier in identifiers:

        hash\_table.insert(identifier)

    while True:

        print("\nМеню:")

        print("1. Найти запись по ключу")

        print("2. Показать все записи")

        print("3. Выйти")

        choice = input("Выберите действие (1-3): ")

        if choice == '1':

            key = input("Введите ключ для поиска: ").strip()

            value, comparisons = hash\_table.search(key)

            if value:

                print(f"Найдена запись: {key}: {value}. Выполнено сравнений: {comparisons}")

            else:

                print(f"Запись с ключом '{key}' не найдена. Выполнено сравнений: {comparisons}")

        elif choice == '2':

            print("Текущие записи в хеш-таблице (в обратном алфавитном порядке):")

            hash\_table.show()

        elif choice == '3':

            print("Завершение программы.")

            break

        else:

            print("Неверный выбор. Пожалуйста, выберите действие от 1 до 4.")

    search\_id = input("Введите идентификатор для поиска: ").strip()

    found, comparisons = hash\_table.search(search\_id)

    if found:

        print(f"Идентификатор '{search\_id}' найден за {comparisons} сравнений.")

    else:

        print(f"Идентификатор '{search\_id}' не найден. Выполнено {comparisons} сравнений.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Вывод:** в ходе выполнения лабораторной работы изучил основные методы организации хеш-таблиц, получил представление о преимуществах и недостатках, присущих различным методам организации таблиц.