

Практическая работа №3

Хеш- таблица

Цель: Изучить хеш-таблицы, и выполняемые операции с ними

Хеш-таблица – это структура данных, реализующая интерфейс ассоциативного массива, то есть она позволяет хранить пары вида "ключ- значение" и выполнять три операции: операцию добавления новой пары, операцию поиска и операцию удаления пары по ключу. Хеш-таблица является массивом, формируемым в определенном порядке хеш-функцией.

Основной принцип работы хеш-таблицы заключается в том, что в качестве входных параметров она принимает пары ключ-значение. Затем с помощью специальной хеш функции получает короткий ключ на основе полученного ключа. И наконец добавляет данные в таблицу. Если в таблице уже существует значения с таким хешем, то объединяет их в коллекции. Внутри коллекции поиск выполняется по изначальному полученному ключу.

В качестве использования хеширования в повседневной жизни можно привести примеры распределение книг в библиотеке по тематическим каталогам, упорядочивание в словарях по первым буквам слов.

Код

```
//Хеш - функция
void displayHash() {
    for (int i = 0; i < MAXSIZE; i++) {
        cout << i << ":\t";
        if (isEmpty(root[i])) {
            cout << "----\n";
            continue;
        }
        Node* temp = root[i];
        while (temp != NULL) {
            cout << temp->data << " ";
            temp = temp->next;
        }
        cout << endl;
    }
}
```

					<i>AuCD.09.03.02.050000 ПР</i>			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.	Ермошина В.А				Практическая работа №3 Хэш-таблица	Лит.	Лист	Листов
Провер.	Берёза А.Н.						2	
Реценз						ИСОиП (филиал) ДГТУ в г.Шахты ИСТ-Тб21		
Н. Контр.								
Утверд.								

Диаграмма деятельности

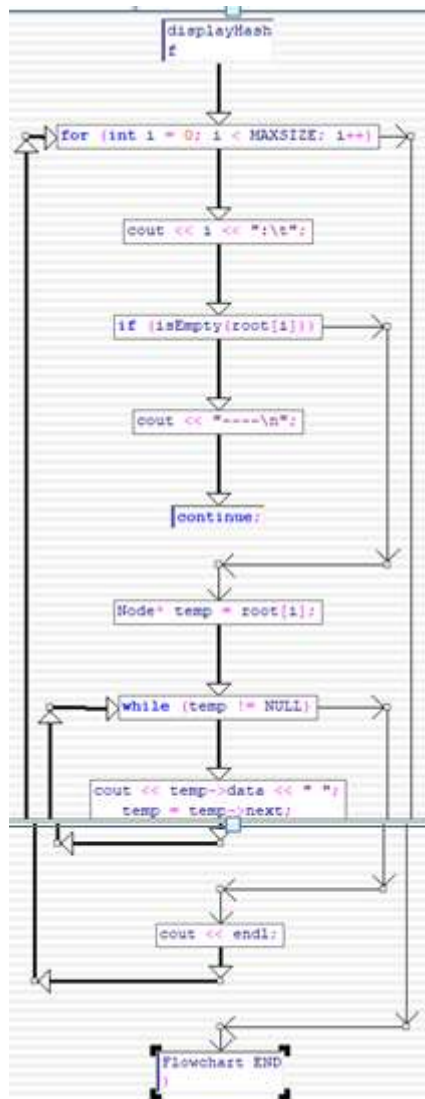


Рисунок 1 - Диаграмма деятельности для функции хеширования

Добавление элемента в хеш-таблицу

//Добавление элемента в хеш - таблицу

```

void Insert(int key) {
    int index = key % MAXSIZE;
    if (isEmpty(root[index])) {
        root[index] = new Node;
        root[index]->data = key;
        root[index]->next = NULL;
    }
    else {
        Node* temp = root[index];
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = new Node;
        temp->next->next = NULL;
        temp->next->data = key;
    }
}
  
```

Изм.	Лист	№ докум.	Подпись	Дата

АуСД.09.03.02.050000 ПР

Лист

3

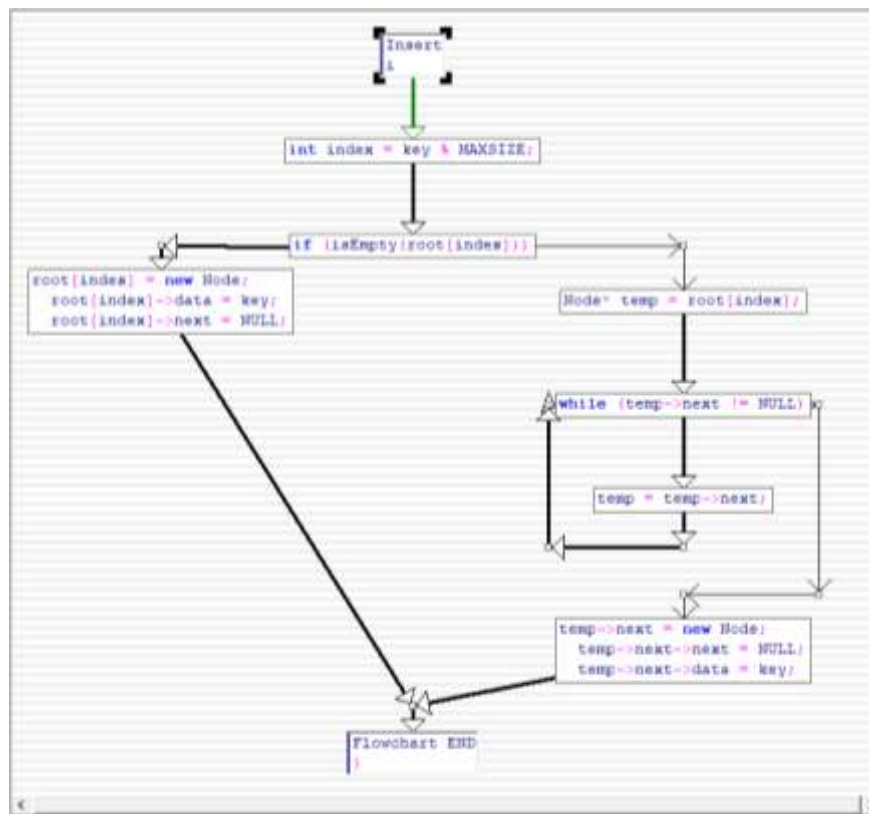


Рисунок 2 - Диаграмма деятельности для добавления элемента в таблицу

Поиск элемента

```

//Поиск элемента
bool Search(int key) {
    int index = key % MAXSIZE;
    if (root[index] == NULL) return false;
    Node* temp = root[index];
    if (temp->data == key) return true;

    while (temp != NULL) {
        if (temp->data == key)
            return true;
        temp = temp->next;
    }
    return false;
}

```

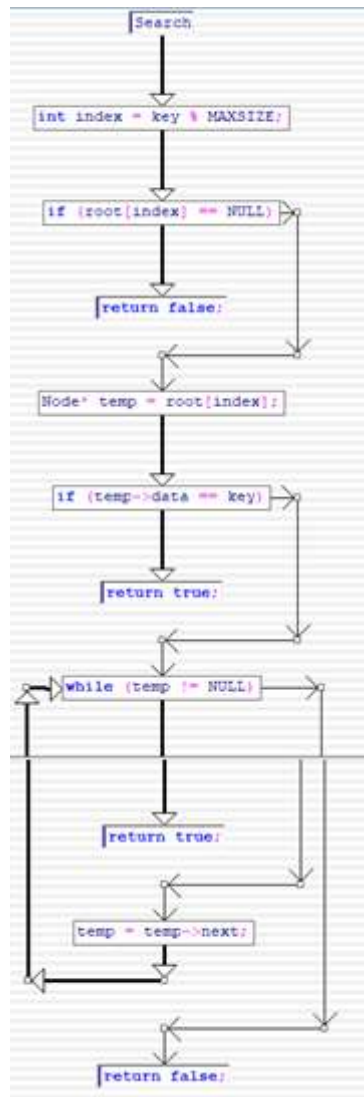


Рисунок 3 - Диаграмма деятельности поиска элемента

Удаление элемента

//Удаление элемента

```

void Delete(int key) {
    int index = key % MAXSIZE;
    if (root[index] == NULL) return;
    Node* temp = root[index];
    Node* prev = NULL;
    if (temp->data == key) {
        root[index] = root[index]->next;
        delete temp;
    }
    else {
        while (temp != NULL) {
            if (temp->data == key) {
                prev->next = temp->next;
                return;
            }
            prev = temp;
            temp = temp->next;
        }
    }
}
  
```

Изм.	Лист	№ докум.	Подпись	Дата

AuCD.09.03.02.050000 ПП

Лист

5

}
}

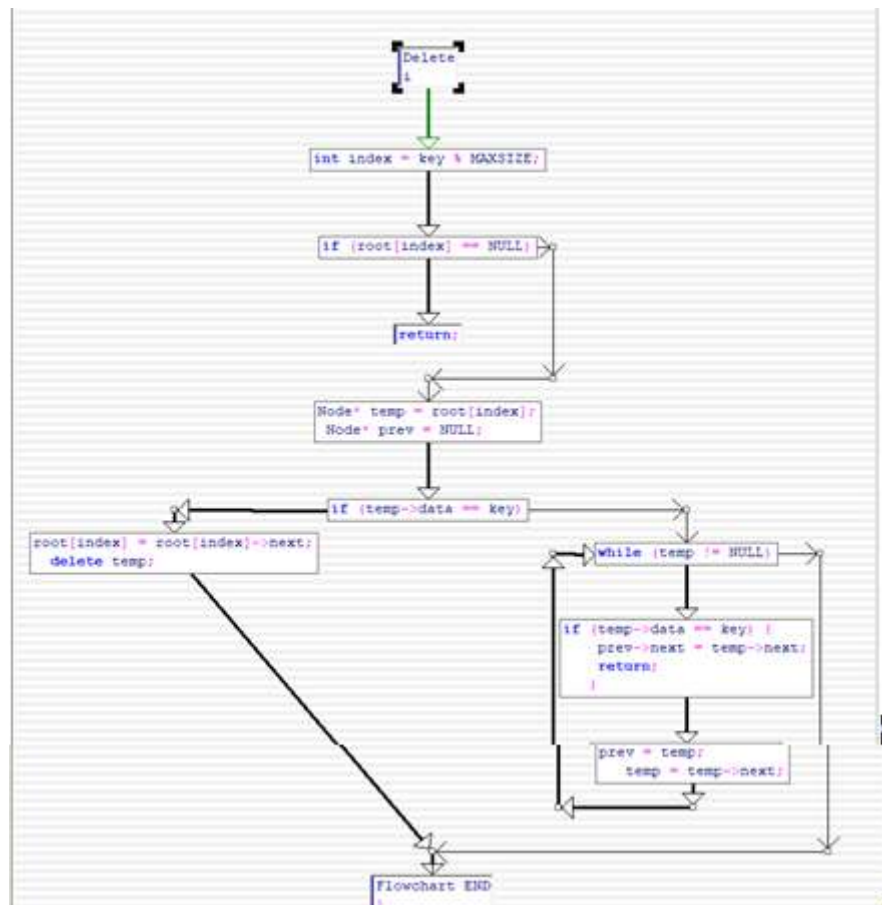


Рисунок 4 - Диаграмма деятельности удаления элемента

Вывод: В данной практической работе изучить хеш-таблицы, и написала программу для выполнения операций с ними на языке C++