

Практическая работа №3

Задание

Создать хеш-таблицу (массив) из пользовательских элементов в постоянной памяти. Таблица должна быть проинициализирована нулями. Создать хеш-функцию. Создать данные пользовательского типа. Записать пользовательские данные в хеш-таблицу, используя хеш-функцию.

Данные взять с количеством характеристик не менее 3.

При возникновении коллизии использовать любой метод: «метод цепочек» или «метод открытой адресации» (можно оба метода).

Написать программу, позволяющую продемонстрировать работу с хеш-таблицей.

Ответ

Консоль отладки Microsoft Visual Studio

```
Вывод хеш-таблицы
Индекс  Список
0       Калинин Д. П.      Mercedes E-class W124      A736AA75
1       Масленникова В. С.  Volvo 360                  A478AA75
2       Горячев В. М.       Volvo 460                  A123AA75
2       Сергеева Е. Е.      Lada 2104                 A754AA75
2       Попова С. А.        Mercedes 190 W201         A755AA75
3       Терентьева К. Д.    Lada 2115                 A678AA75
3       Сорокина Н. М.     Mercedes C-class W202     A947AA75
3       Киселев Д. И.      BMW M3 (E30)              A767AA75
4       Панфилова Т. М.    Volvo S40                  A325AA75
5       Кузнецова Т. К.     Volvo S70                  A241AA75
5       Виноградова Н. И.  Mercedes CL-class C215   A875AA75
5       Жуков Л. Г.        BMW M5 (E28)              A786AA75
5       Евдокимов К. А.    BMW M3 (E46)              A784AA75
5       Макаров Б. Д.      Volkswagen Tiguan         A878AA75
6       Горшков С. Д.      Lada 2107                 A794AA75
6       Петров И. С.       Volvo S90                  A8757B75
7       Кузнецов Л. А.     Lada 2114                 A716AA75
7       Андреев М. М.      Jaguar XJ                  A984AA75
7       Богомолова А. М.   Volkswagen Polo           A897AA75
8       Герасимова С. М.   Volvo S40                  A456AA75
```

Для поиска ведите номер интересующего
Вас автомобиля в формате A***AA**

Искомый номер >> A123AA75

Найдено: Горячев В. М. Volvo 460 A123AA75

Продолжить поиск?

- 1 - Да.
- 2 - Нет.

1

Искомый номер >> A666AA77

По запросу << A666AA77 >> не удалось ничего найти

Продолжить поиск?

- 1 - Да.
- 2 - Нет.

2

D:\Project C++\Практика номер 3\x64\Debug\Практика номер 3.exe

Мой код

```
#include <iostream>
#include <string>
#include <Windows.h>

using namespace std;

#define SizeData 20 // Размерность массива исходных данных
#define SizeHashTable 9 // Размерность хеш-таблицы

struct CRC // Car Registration Cards
{
    string model; // Марка автомобиля - ключ
    string owner; // Владелец автомобиля
    string number; // Номер автомобиля
    CRC* next;
};

int HashFunction(string brand)
{
    int sum = 0;

    for (int i = 0; i < 3; i++)
    {
        sum += abs(brand[i] - '0');
    }

    int cell_index = sum % SizeHashTable;

    return cell_index;
}

void PrintHashTable(CRC* HashTable[]) // Печать хеш-таблицы
{
    cout << " Вывод хеш-таблицы\n Индекс\t Список\n";
    for (int i = 0; i < SizeHashTable; i++)
    {
        if (HashTable[i] != NULL)
        {
            CRC* S = HashTable[i];
            while (S != NULL)
            {
                cout << " " << i << "\t";
                printf("%-20s", (S->owner).c_str());
                printf("%-25s", (S->model).c_str());
                printf("%-10s", (S->number).c_str());
                cout << "\n";

                S = S->next;
            }
            cout << "\n";
        }
    }
}

void SearchHashTable(CRC* HashTable[], string SearchQuery) // Поиск
{
    HANDLE h;
    h = GetStdHandle(STD_OUTPUT_HANDLE); // окно, в котором нужно изменить цвет

    int index = HashFunction(SearchQuery);

    if (!HashTable[index])
    {

```

```

        SetConsoleTextAttribute(h, 4);
        cout << "    По запросу << " << SearchQuery << " >> не удалось ничего
найти" << endl;
    }
    else
    {
        bool stopSearch = false;
        CRC* S = HashTable[index];
        while (S != NULL)
        {
            if (S->number == SearchQuery)
            {
                SetConsoleTextAttribute(h, 2);
                cout << "    Найдено: " << S->owner << " " << S->model << "
" << S->number << "\t";
                stopSearch = true;
                break;
            }
            S = S->next;
        }

        SetConsoleTextAttribute(h, 4);
        if (!stopSearch) cout << " По запросу << " << SearchQuery << " >> не
удалось ничего найти" << endl;
    }

    SetConsoleTextAttribute(h, 7);
}

void main()
{
    setlocale(0, "Rus");
    HANDLE h;
    h = GetStdHandle(STD_OUTPUT_HANDLE);

    /* Исходные данные для работы */
    CRC CRC_list[SizeData] = {
        "Volvo S90", "Петров И. С.", "AB757B75", NULL,
        "Volkswagen Polo", "Богомолова А. М.", "A897AA75", NULL,
        "Volkswagen Tiguan", "Макаров Б. Д.", "A878AA75", NULL,
        "Jaguar XJ", "Андреев М. М.", "A984AA75", NULL,
        "BMW M3 (E30)", "Киселев Д. И.", "A767AA75", NULL,
        "BMW M3 (E46)", "Евдокимов К. А.", "A784AA75", NULL,
        "BMW M5 (E28)", "Жуков Л. Г.", "A786AA75", NULL,
        "Mercedes 190 W201", "Попова С. А.", "A755AA75", NULL,
        "Mercedes C-class W202", "Сорокина Н. М.", "A947AA75", NULL,
        "Mercedes CL-class C215", "Виноградова Н. И.", "A875AA75", NULL,
        "Mercedes E-class W124", "Калинин Д. П.", "A736AA75", NULL,
        "Lada 2104", "Сергеева Е. Е.", "A754AA75", NULL,
        "Lada 2114", "Кузнецов Л. А.", "A716AA75", NULL,
        "Lada 2107", "Горшков С. Д.", "A794AA75", NULL,
        "Lada 2115", "Терентьева К. Д.", "A678AA75", NULL,
        "Volvo S40", "Панфилова Т. М.", "A325AA75", NULL,
        "Volvo S70", "Кузнецова Т. К.", "A241AA75", NULL,
        "Volvo S40", "Герасимова С. М.", "A456AA75", NULL,
        "Volvo 460", "Горячев В. М.", "A123AA75", NULL,
        "Volvo 360", "Масленникова В. С.", "A478AA75", NULL,
    };

    /* Подготовка хеш-таблицы */
    CRC* HashTable[SizeHashTable] = { //хеш-таблица адресов
        NULL, NULL, NULL,
        NULL, NULL, NULL,
        NULL, NULL, NULL
    };

```

```

};

/* Заполнение хеш-таблицы */
for (int i = 0; i < SizeData; i++)
{
    int index = HashFunction(CRC_list[i].number);

    if (HashTable[index] == NULL)
    {
        HashTable[index] = &CRC_list[i];
    }
    else {
        CRC* S = HashTable[index];
        HashTable[index] = &CRC_list[i];
        HashTable[index]->next = S;
    }
}

/* Контрольная печать */
PrintHashTable(HashTable);

/* Поиск */
string SearchQuery;
SetConsoleTextAttribute(h, 6);
cout << " Для поиска ведите номер интересующего\n Вас автомобиля в формате
A***AA**\n";
SetConsoleTextAttribute(h, 7);

bool Search = true;
int goNext;

while (Search)
{
    cout << " Искомый номер >> ";
    cin >> SearchQuery;

    SearchHashTable(HashTable, SearchQuery);

    cout << "\n Продолжить поиск?\n    1 - Да.\n    2 - Нет.\n    "; cin >>
goNext;
    if (goNext == 1 ? Search = true : Search = false);
}

SetConsoleTextAttribute(h, 8);
}

```