

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский
политехнический университет»**

Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
направление подготовки: 09.03.01 – «Информатика и вычислительная
техника»

**Отчет по
лабораторной работе 18**

Выполнил студент гр. ИВТ-23-16
Давыдов Андрей Юрьевич

Проверил:

ст. преп. каф. ИТАС

Яруллин Денис Владимирович

(оценка)

(подпись)

(дата)

г. Пермь, 2023

Постановка задачи

Создание хеш-таблицы

Код программы:

```
#include <iostream>
#include <string>
#include <ctime>

using namespace std;

const int TABLE_SIZE = 50;
const int NAME_COUNT = 30;

string rand_name[NAME_COUNT] = { "Александр", "Мария", "Иван", "Екатерина", "Максим",
    "Дарья", "Артём", "Анастасия", "Дмитрий", "Софья",
    "Михаил", "Ксения", "Владимир", "Елизавета", "Андрей",
    "Марина", "Игорь", "Юлия", "Алексей", "Яна",
    "Тимофей", "Полина", "Роман", "Валерия", "Никита",
    "Татьяна", "Арсений", "Вероника", "Евгений", "Кристина" };

int get_hash(string str) {
    unsigned key = 0;
    for (char ch : str) {
        key += ch;
    }
    return static_cast<int>(key * 0.2) % TABLE_SIZE;
}

struct Node {
    string name;
    int series;
    int number;
    Node* next;

    Node() : name(" "), series(0), number(0), next(nullptr) {}
    Node(string n, int s, int num) : name(n), series(s), number(num), next(nullptr) {}
};
```

```

class HashTable {
private:
    Node** table;
public:
    HashTable() {
        table = new Node * [TABLE_SIZE];
        for (int i = 0; i < TABLE_SIZE; ++i) {
            table[i] = nullptr;
        }
    }

    ~HashTable() {
        for (int i = 0; i < TABLE_SIZE; ++i) {
            Node* entry = table[i];
            while (entry != nullptr) {
                Node* prev = entry;
                entry = entry->next;
                delete prev;
            }
        }
        delete[] table;
    }

    void set_node(string name, int series, int number) {
        int hash = get_hash(name);
        Node* prev = nullptr;
        Node* entry = table[hash];

        while (entry != nullptr && entry->name != " ") {
            prev = entry;
            entry = entry->next;
        }

        if (entry == nullptr) {
            entry = new Node(name, series, number);
            if (prev == nullptr) {
                table[hash] = entry;
            }
            else {
                prev->next = entry;
            }
        }
        else {
            entry->name = name;
            entry->series = series;
            entry->number = number;
        }
    }

    void print_table() {
        for (int i = 0; i < TABLE_SIZE; ++i) {
            Node* entry = table[i];
            while (entry != nullptr) {
                cout << "Name: " << entry->name << endl;
                cout << "Series: " << entry->series << endl;
                cout << "Number: " << entry->number << endl;
                cout << "=====\n";
                entry = entry->next;
            }
        }
    }

    void set_table(int count) {
        srand(static_cast<unsigned int>(time(nullptr)));
        for (int i = 0; i < count; ++i) {
            int r = rand() % NAME_COUNT;
            int series = rand() % 8999 + 1000;
            int number = rand() % 899999 + 100000;
            set_node(rand_name[r], series, number);
        }
    }
};

int main() {
    system("chcp 1251>0");
    HashTable* people = new HashTable;
    people->set_table(80);
    people->print_table();
    delete people;

    return 0;
}

```

Тест программы

```
Name: Елизавета
Series: 6348
Number: 111141
=====
Name: Елизавета
Series: 9909
Number: 132504
=====
Name: Елизавета
Series: 9624
Number: 130928
=====
Name: Екатерина
Series: 6067
Number: 130273
=====
Name: Александр
Series: 2757
Number: 116044
=====
Name: Екатерина
Series: 6226
Number: 108663
=====
Name: Александр
Series: 4397
Number: 109731
=====
Name: Екатерина
Series: 4816
Number: 123975
=====
Name: Александр
Series: 3041
Number: 104912
=====
Name: Екатерина
Series: 6669
Number: 111113
=====
Name: Александр
Series: 9676
Number: 100435
=====
Name: Владимир
Series: 3858
Number: 112618
=====
```

=====

Name: Владимир
Series: 4848
Number: 119433

=====

Name: Анастасия
Series: 4803
Number: 108500

=====

Name: Евгений
Series: 1646
Number: 116902

=====

Name: Кристина
Series: 2285
Number: 106157

=====

Name: Кристина
Series: 6263
Number: 108785

=====

Name: Кристина
Series: 2637
Number: 109323

=====

Name: Алексей
Series: 9625
Number: 109236

=====

Name: Алексей
Series: 6517
Number: 131018

=====

Name: Алексей
Series: 2034
Number: 129252

=====

Name: Алексей
Series: 1209
Number: 122285

=====

Name: Кристина
Series: 3489
Number: 101283

=====

Name: Арсений
Series: 6875
Number: 100073

=====