Министерство образования РФ

Пермский национальный исследовательский политехнический университет

Кафедра ИТАС

ЛАБОРАТОРНАЯ РАБОТА № 11

ПО ОСНОВАМ АЛГОРИТМИЗАЦИИ ЗА I СЕМЕСТР

Вариант 15

Выполнил студент:

Сташков Арсений Дмитриевич

Группа РИС-20-1бз

Шифр 20-ЭТФ-644

Проверила:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

ПЕРМЬ 2020

# Лабораторная работа №11 "Поиск данных с помощью хэш-таблиц"

**Вариант №15**

**Цель:**

1. Изучить построение функции хеширования и алгоритмов хеширования данных
2. Научиться разрабатывать алгоритмы открытого и закрытого хеширования при решении задач на языке C++.
3. **Задача:** Создать динамический массив из записей (в соответствии с вариантом), содержащий не менее 100 элементов. Для заполнения элементов массива использовать ДСЧ.
4. Предусмотреть сохранение массива в файл и загрузку массива из файла.
5. Предусмотреть возможность добавления и удаления элементов из массива (файла).
6. Выполнить поиск элемента в массиве по ключу в соответствии с вариантом. Для поиска использовать хэш-таблицу.
7. Подсчитать количество коллизий при размере хэш-таблицы 40, 75 и 90 элементов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | ФИО, дата\_рождения, адрес | ФИО | H(k)=k mod M | Метод открытой адресации |

**Анализ задачи:** Сделаем основу программы по аналогии с предыдущей лабораторной работой. Для упрощения восприятия программы она была разбита на 3 файла. В 1 файле мы создаем меню и вызываем функции из 2 файла. 3 файл содержит описание этих функций.

**Решение:**

// Пишем основной файл

#include "Function.h"

using namespace std;

int stateMenu;

void Menu()

{

cout << "Веберите действие:\n\n"

<< "(0) Выйти\n"

<< "(1) Ввести данные\n"

<< "(2) Показать данные\n"

<< "(3) Измененить данные\n"

<< "(4) Удалить данные\n"

<< "(5) Добавить данные\n"

<< "(6) Сохранить данные\n\n"

<< "Ваш выбор: ";

cin >> stateMenu;

}

/\*

Данные:

Фамилия Имя Отчество

Дата рождения

Адрес

\*/

int main()

{

// Ввод и вывод на русском

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

DataInitialization();

Menu();

int actions;

string fileName;

while (stateMenu != 0)

{

switch (stateMenu)

{

case 1:

// Чистим консоль

system("cls");

cout << "Ввод вручную (1) или из файла (0)? ";

cin >> actions;

system("cls");

if (actions == 1)

{

// Ввод вручную

DataEntry();

}

else

{

cout << "Введите название файла: ";

cin >> fileName;

DataReading(fileName);

}

system("pause");

system("cls");

Menu();

break;

case 2:

system("cls");

Print();

system("pause");

system("cls");

Menu();

break;

case 3:

system("cls");

DataChange();

system("pause");

system("cls");

Menu();

break;

case 4:

system("cls");

DeleteData();

system("pause");

system("cls");

Menu();

break;

case 5:

system("cls");

AddData();

system("pause");

system("cls");

Menu();

break;

case 6:

system("cls");

cout << "Введите название файла: ";

cin >> fileName;

system("cls");

SaveData(fileName);

system("pause");

system("cls");

Menu();

break;

default:

cout << "Пункт меню выбран неверно!\n";

}

}

system("cls");

if (DataCleaning())

{

cout << "Данные очищены\n";

}

else

{

cout << "Данные не очищены!\n";

}

cout << "Работа завершена\n";

return 0;

}

// Пишем файл с прототипами функций

#pragma once

#include <iostream>

#include <fstream>

#include <string>

#include <Windows.h>

using namespace std;

// Прототипы функций

void DataInitialization();

void DataEntry();

void DataReading(string fileName);

void Print();

bool DataCleaning();

void DataChange();

void Copy();

int AmountOfData();

void DeleteData();

void AddData();

void SaveData(string fileName);

// Пишем файл с описанием функций

#include "Function.h"

// Инициализация данных

void DataInitialization()

{

ofstream buf("Buffer.txt");

if (!buf)

cout << "Ошибка создания буферного файла!/n";

buf.close();

}

// Ввод данных вручную

void DataEntry()

{

string name;

string dateOfBirth;

string address;

int n;

cout << "Введите количество данных: ";

cin >> n;

// Открываем буферный файл и записываем данные в него

ofstream record("Buffer.txt", ios::app);

if (record)

{

for (int i = 0; i < n; i++)

{

cout << "ФИО (без пробелов): ";

cin >> name;

cout << "Дата рождения: ";

cin >> dateOfBirth;

cout << "Адрес: ";

cin >> address;

record << name << "\n";

record << dateOfBirth << "\n";

if (i < n - 1)

{

record << address << " мес.\n";

}

else

{

record << address << " мес.";

}

cout << "----------------------\n";

}

}

else

{

cout << "Ошибка открытия файла!\n";

}

record.close();

}

// Чтение данных

void DataReading(string fileName)

{

ifstream reading(fileName);

ofstream record("Buffer.txt", ios::out);

if (reading)

{

if (record)

{

string name;

string dateOfBirth;

string address;

int n;

reading >> n;

record << n << "\n";

for (int i = 0; i < n; i++)

{

reading >> name;

reading >> dateOfBirth;

reading >> address;

record << name << "\n";

record << dateOfBirth << "\n";

if (i < n - 1)

{

record << address << "\n";

}

else

{

record << address;

}

}

cout << "Данные считаны!\n";

}

else

cout << "Данные считаны!\n";

}

else

cout << "Ошибка открытия файла!\n";

// Закрытие файла

reading.close();

record.close();

}

// Вывод данных

void Print()

{

ifstream reading("Buffer.txt");

if (reading)

{

string name;

string dateOfBirth;

string address;

int n;

reading >> n;

cout << "Количество данных: " << n << "\n\n";

for (int i = 0; i < n; i++)

{

cout << "Данные о №" << i + 1 << "\n";

reading >> name;

cout << "ФИО: " << name << "\n";

reading >> dateOfBirth;

cout << "Дата рождения: " << dateOfBirth << "\n";

reading >> address;

cout << "Адрес: " << address << "\n";

cout << "----------------------\n";

}

}

else

{

cout << "Ошибка открытия файла!\n";

}

reading.close();

}

// Очистка данных

bool DataCleaning()

{

bool clear = false;

fstream buf("Buffer.txt", ios::out);

if (!buf)

{

cout << "Ошибка открытия буферного файла!\n";

}

buf.clear();

if (buf.peek() == EOF)

{

clear = true;

}

else

{

clear = false;

}

buf.close();

return clear;

}

// Изменение данных

void DataChange()

{

Copy();

ifstream reading("Buffer.txt");

ofstream record("Buffer.txt", ios::out);

if (reading)

{

if (record)

{

string name;

string dateOfBirth;

string address;

int n, n2;

reading >> n;

cout << "Выберите номер изменяемого элемента (от 1 до " << n << "): ";

cin >> n2;

n2--;

system("cls");

record << n << "\n";

if (n2 >= 0 && n2 < n)

{

for (int i = 0; i < n; i++)

{

if (i != n2)

{

reading >> name;

record << name << "\n";

reading >> dateOfBirth;

record << dateOfBirth << "\n";

reading >> address;

if (i < n - 1)

{

record << address << "\n";

}

else

{

record << address;

}

}

else

{

cout << "Введите данные №1 (ФИО): ";

cin >> name;

cout << "Введите данные №2 (Дата рождения): ";

cin >> dateOfBirth;

cout << "Введите данные №3 (Адрес): ";

cin >> address;

record << name << "\n";

record << dateOfBirth << "\n";

if (i < n - 1)

{

record << address << "\n";

}

else

{

record << address;

}

reading >> name;

reading >> dateOfBirth;

reading >> address;

}

}

cout << "Данные изменены!\n";

}

else

"Номер введен неверно!";

}

else

cout << "Ошибка открытия файла!\n";

}

else

cout << "Ошибка открытия буферного файла!\n";

record.close();

reading.close();

remove("Buffer.txt");

}

// Копирование данных

void Copy()

{

ifstream reading("Buffer.txt");

ofstream record("Buffer.txt", ios::out);

if (reading)

{

if (record)

{

string name;

string dateOfBirth;

string address;

int n;

reading >> n;

record << n << "\n";

for (int i = 0; i < n; i++)

{

reading >> name;

record << name << "\n";

reading >> dateOfBirth;

record << dateOfBirth << "\n";

reading >> address;

if (i < n - 1)

{

record << address << "\n";

}

else

{

record << address;

}

}

}

else

cout << "Ошибка открытия файла!\n";

}

else

cout << "Ошибка открытия буферного файла!\n";

}

// Кол-во данных из файла

int AmountOfData()

{

ifstream buf("Buffer.txt");

int n;

if (buf)

{

buf >> n;

}

else

{

cout << "Ошибка открытия буферного файла!\n";

}

buf.close();

return n;

}

// Удаление данных

void DeleteData()

{

Copy();

ifstream reading("Buffer.txt");

ofstream record("Buffer.txt", ios::out);

if (reading)

{

if (record)

{

string name;

string dateOfBirth;

string address;

int n, n2;

reading >> n;

int b = n - 1;

cout << "Выберите номер изменяемого элемента от 1 до " << n << "): ";

cin >> n2;

n2--;

system("cls");

if (n2 >= 0 && n2 < n)

{

for (int i = 0; 1 < n; i++)

{

if (i != n2)

{

reading >> name;

record << name << "\n";

reading >> dateOfBirth;

record << dateOfBirth << "\n";

reading >> address;

if (i < n - 1)

{

record << address << "\n";

}

else

{

record << address;

}

}

cout << "Данные удалены!\n";

}

}

else

cout << "Номер введен неверно!\n";

}

else

cout << "Ошибка открытия файла!\n";

}

else

cout << "Ошибка открытия буферного файла!\n";

record.close();

reading.close();

}

// Добавление данных

void AddData()

{

string name;

string dateOfBirth;

string address;

int n = AmountOfData() + 1;

ofstream record("Buffer.txt", ios::app);

ofstream record2("Buffer.txt", ios::out | ios::in);

if (record2)

{

record2 << n << "\n";

}

else

{

cout << "Ошибка открытия буферного файла!\n";

}

if (record)

{

record << "\n";

cout << "ФИО (без пробелов): ";

cin >> name;

cout << "Дата рождения: ";

cin >> dateOfBirth;

cout << "Адрес: ";

cin >> address;

record << name << "\n";

record << dateOfBirth << " руб.\n";

record << address << "\n";

}

else

cout << "Ошибка открытия файла!\n";

record.close();

record2.close();

}

// Сохранение данных

void SaveData(string fileName)

{

ifstream reading("Buffer.txt");

ofstream record(fileName, ios::out);

if (reading)

{

if (record)

{

string name;

string dateOfBirth;

string address;

int n;

reading >> n;

record << n << "\n";

for (int i = 0; i < n; i++)

{

reading >> name;

record << name << "\n";

reading >> dateOfBirth;

record << dateOfBirth << "\n";

reading >> address;

if (i < n - 1)

{

record << address << "\n";

}

else

{

record << address;

}

cout << "Данные сохранены в файле" << fileName << "\n";

}

}

else

cout << "Ошибка открытия буферного файла!\n";

}

else

cout << "Ошибка открытия файла!\n";

}



# 

