|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_***ИУК «Информатика и управление»*\_\_**\_\_\_\_\_\_\_\_\_\_\_\_

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА**

**«Обработка файлов произвольного доступа»**

**ДИСЦИПЛИНА: «Высокоуровневое программирование»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-12Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) (Ф.И.О.) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_Пчелинцева Н.И. )  (Подпись) (Ф.И.О.) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |
| Калуга, 2020 | | |

**Цель:** приобретение практических навыков создания и обработки файлов произвольного доступа.

**Задачи:**

1. Познакомиться с организацией файлов произвольного доступа
2. Изучить основные программные средства для произвольной обработки файлов указанного типа

**Вариант 2**

**Задача 1**

Запись имеет вид: название вуза, число студентов, количество факультетов. Cоздать файл из 10 записей, просмотреть файл, добавить в конец файла информацию о трех новых вузах и посчитать общее число студентов.

**Блок-схема:**





**Листинг:**

***Problem1.cpp***

#include <iostream>

#include <fstream>

#include <iomanip>

#include "MinusOne.h"

using namespace std;

const char path[]{"prob1.dat"};

const int MAXLEN = 20;

struct field

{

char HEI[MAXLEN]{};

int students{};

int faculties{};

};

void input();

void output();

void students();

char repeat();

void problem1()

{

char c = '0';

while (c == '0')

{

cout << "\tВУЗы" << endl << endl;

ofstream clr(path, ios::binary);

clr.close();

cout << "Введите 10 ВУЗов:" << endl << endl;

for (int i = 0; i < 10; i++)

{

cout << i + 1 << "-й ВУЗ" << endl;

input();

cout << endl;

}

output();

cout << endl;

cout << "Введите еще 3 ВУЗа:" << endl << endl;

for (int i = 0; i < 3; i++)

{

cout << i + 11 << "-й ВУЗ" << endl;

input();

cout << endl;

}

students();

cout << endl;

c = repeat();

system("cls");

}

}

void input()

{

ofstream out(path, ios::binary | ios::app);

field in{};

cout << "Введите название ВУЗа: ";

cin.getline(in.HEI, MAXLEN);

cout << "Введите число студентов: ";

cin >> in.students;

cout << "Введите количество факультетов: ";

cin >> in.faculties;

out.write((char\*)&in, sizeof(in));

out.close();

cin.ignore(32767, '\n');

}

void output()

{

ifstream in(path, ios::binary);

field out{};

cout << "\tСодержимое файла" << endl << endl;

cout << setw(20) << "Название ВУЗа" << " ";

cout << setw(10) << "Студенты" << " ";

cout << setw(10) << "Факультеты" << endl;

while (in.tellg() != -1)

{

in.read((char\*)&out, sizeof(out));

cout << setw(20) << out.HEI << " ";

cout << setw(10) << out.students << " ";

cout << setw(10) << out.faculties << endl;

minusone(in);

}

in.close();

}

void students()

{

ifstream in(path, ios::binary);

int sum = 0;

while (in.tellg() != -1)

{

field out{};

in.read((char\*)&out, sizeof(out));

sum += out.students;

minusone(in);

}

cout << "Всего студентов: " << sum << endl;

}

char repeat()

{

char res = '0';

cout << "Введите 0 для запуска программы еще раз: ";

cin.get(res);

if (res == '0')

{

cin.get(res);

if (res != '\n')

{

res = '1';

}

else

{

res = '0';

}

}

else if (res != '\n')

{

cin.ignore(32767, '\n');

}

return res;

}

***Problem1.h***

#pragma once

#ifndef PROBLEM1\_H

#define PROBLEM1\_H

void problem1();

#endif // !PROBLEM1\_H

**Тестирование:**

ВУЗы

Введите 10 ВУЗов:

1-й ВУЗ

Введите название ВУЗа: Univ1

Введите число студентов: 10000

Введите количество факультетов: 1

2-й ВУЗ

Введите название ВУЗа: Univ2

Введите число студентов: 20000

Введите количество факультетов: 2

3-й ВУЗ

Введите название ВУЗа: Univ3

Введите число студентов: 30000

Введите количество факультетов: 3

4-й ВУЗ

Введите название ВУЗа: Univ4

Введите число студентов: 40000

Введите количество факультетов: 4

5-й ВУЗ

Введите название ВУЗа: Univ5

Введите число студентов: 50000

Введите количество факультетов: 5

6-й ВУЗ

Введите название ВУЗа: Univ6

Введите число студентов: 60000

Введите количество факультетов: 6

7-й ВУЗ

Введите название ВУЗа: Univ7

Введите число студентов: 70000

Введите количество факультетов: 7

8-й ВУЗ

Введите название ВУЗа: Univ8

Введите число студентов: 80000

Введите количество факультетов: 8

9-й ВУЗ

Введите название ВУЗа: Univ9

Введите число студентов: 90000

Введите количество факультетов: 9

10-й ВУЗ

Введите название ВУЗа: Univ10

Введите число студентов: 100000

Введите количество факультетов: 10

Содержимое файла

Название ВУЗа Студенты Факультеты

Univ1 10000 1

Univ2 20000 2

Univ3 30000 3

Univ4 40000 4

Univ5 50000 5

Univ6 60000 6

Univ7 70000 7

Univ8 80000 8

Univ9 90000 9

Univ10 100000 10

Введите еще 3 ВУЗа:

11-й ВУЗ

Введите название ВУЗа: Univ11

Введите число студентов: 110000

Введите количество факультетов: 11

12-й ВУЗ

Введите название ВУЗа: Univ12

Введите число студентов: 120000

Введите количество факультетов: 12

13-й ВУЗ

Введите название ВУЗа: Univ13

Введите число студентов: 130000

Введите количество факультетов: 13

Всего студентов: 910000

Введите 0 для запуска программы еще раз:

**Задача 2**

Файл записей переменной длины перед каждой записью содержит целое, определяющее ее длину. Каждая запись – массив целых чисел. Написать функцию добавления нового массива в такой файл, функцию вывода размера выбранного массива и функцию вывода элементов выбранного массива.

**Блок-схема:**







**Листинг:**

***Problem2.cpp***

#include <iostream>

#include <fstream>

#include "MinusOne.h"

using namespace std;

const char path[]{"prob2.dat"};

void addarr();

void readarr(bool mode);

void problem2()

{

cout << "\tМассивы" << endl << endl;

int code = 0;

while (code != 3)

{

cout << " 0. Добавить массив" << endl

<< " 1. Узнать длину массива" << endl

<< " 2. Узнать содержимое массива" << endl

<< " 3. Выход" << endl << endl

<< "Введите код: ";

cin >> code;

system("cls");

cout << "\tМассивы" << endl << endl;

switch (code)

{

case 0:

addarr();

break;

case 1:

readarr(false);

break;

case 2:

readarr(true);

break;

}

}

}

void addarr()

{

int len{};

ofstream file(path, ios::binary | ios::app);

cout << "Введите длину массива: ";

cin >> len;

file.write((char\*)&len, sizeof(len));

cout << "Введите элементы:" << endl;

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

{

int num{};

cout << i << ". ";

cin >> num;

file.write((char\*)&num, sizeof(num));

}

cout << endl;

file.close();

}

void readarr(bool mode)

{

cout << "Введите порядковый номер массива: ";

int num{};

cin >> num;

if (num < 0)

{

cout << "Неправильный номер" << endl;

}

else

{

ifstream file(path, ios::binary);

int i = 0;

minusone(file);

while (file.tellg() != -1 && i < num)

{

int len{};

file.read((char\*)&len, sizeof(len));

file.seekg(len\*sizeof(int), ios\_base::cur);

i++;

minusone(file);

}

if (file.tellg() != -1 && i == num)

{

int len{};

file.read((char\*)&len, sizeof(len));

if (mode == false)

{

cout << "Длина " << i << "-го массива: " << len << endl;

}

else

{

for (int j = 0; j < len; j++)

{

int elem{};

file.read((char\*)&elem, sizeof(elem));

cout << j << ". " << elem << endl;

}

}

}

else

{

cout << "Массив не найден" << endl;

}

cout << endl;

file.close();

}

}

***Problem2.h***

#pragma once

#ifndef PROBLEM2\_H

#define PROBLEM2\_H

void problem2();

#endif // !PROBLEM2\_H

**Тестирование:**

Массивы

0. Добавить массив

1. Узнать длину массива

2. Узнать содержимое массива

3. Выход

Введите код:

***При вводе 0:***

Массивы

Введите длину массива: 3

Введите элементы:

0. 1

1. 2

2. 3

0. Добавить массив

1. Узнать длину массива

2. Узнать содержимое массива

3. Выход

Введите код:

***При вводе 1:***

Массивы

Введите порядковый номер массива: 0

Длина 0-го массива: 3

0. Добавить массив

1. Узнать длину массива

2. Узнать содержимое массива

3. Выход

Введите код:

***При вводе 2:***

Массивы

Введите порядковый номер массива: 0

0. 1

1. 2

2. 3

0. Добавить массив

1. Узнать длину массива

2. Узнать содержимое массива

3. Выход

Введите код:

**Основная программа:**

#include <iostream>

#include "Menu.h"

#include "Problem1.h"

#include "Problem2.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

char\* progname = new char[39] {"Обработка файлов произвольного доступа"};

int probs = 2;

const char\*\* probname = new const char\*[13] { "ВУЗы", "Массивы" };

void (\*func[])() { problem1, problem2 };

menu(progname, probs, probname, func);

delete[] progname;

delete[] probname;

return 0;

}

**Меню:**

***Menu.cpp:***

#include <iostream>

using namespace std;

const int MAXLEN = 255;

void display(char progname[], int probs, const char\* probname[]);

int digcheck(char str[]);

void menu(char progname[], int probs, const char\* probname[], void (\*prog[])())

{

display(progname, probs, probname);

char\* input = new char[MAXLEN] {};

int num{};

while (num != probs)

{

cout << "Введите код: ";

do

{

cin >> input;

cin.ignore(32767, '\n');

num = digcheck(input);

if (num > probs || num < 0)

{

display(progname, probs, probname);

cout << "Неправильный код, попробуйте еще раз: ";

}

} while (num > probs || num < 0);

if (num != probs)

{

system("cls");

prog[num]();

display(progname, probs, probname);

}

}

delete[] input;

}

void display(char progname[],int probs, const char\* probname[])

{

system("cls");

cout << "\t" << progname << endl << endl;

cout << "Задания:" << endl;

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

{

cout << " " << i << ". " << probname[i] << endl;

}

cout << " " << probs << ". Выход" << endl;

}

int digcheck(char str[])

{

bool alldigit = true;

for (int i = 0; i < strlen(str); i++)

{

if (!isdigit(str[i]))

{

alldigit = false;

}

}

if (alldigit)

{

return atoi(str);

}

else

{

return -1;

}

}

***Menu.h:***

#pragma once

#ifndef MENU\_H

#define MENU\_H

void menu(char progname[], int probs, const char\* probname[], void (\*prog[])());

#endif // !MENU\_H

**Тестирование:**

Обработка файлов произвольного доступа

Задания:

0. ВУЗы

1. Массивы

2. Выход

Введите код:

**Дополнительные программы**

***MinusOne.cpp:***

#include <fstream>

using namespace std;

void minusone(ifstream& in)

{

char c;

in.read((char\*)&c, sizeof(c));

if (in.tellg() != -1)

{

int x = sizeof(c);

in.seekg(-x, ios::cur);

}

}

***MinusOne.h:***

#include <fstream>

#pragma once

#ifndef MINUSONE\_H

#define MINUSONE\_H

void minusone(std::ifstream& in);

#endif // !MINUSONE\_H

**Вывод:** были получены практические навыки создания, просмотра и редактирования файлов произвольного доступа с записями определенной и произвольной длины.