МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА»

Институт радиоэлектроники и информационных технологий

Кафедра «Прикладная математика»

Лабораторная работа №4

Выполнил:

Студент группы 20-ПМ-1

Ермолаев Н.Д.

Проверил:

Доцент кафедры «Прикладная математика»

Чернов А.Г.

Нижний Новгород

2021

**Содержание**

[Постановка задачи…………………………………………………………………………..3](#_Toc20318830)

[Описание структуры………………………………………………………………………...4](#_Toc20318831)

[Код……………………………………………………………………………………………5](#_Toc20318832)

[Результат работы программы……………………………………………………………..11](#_Toc20318833)

**Постановка задачи**

Реализовать программу для работы с почтовым сервером в текстовом файле. Программа должна выполнять следующие функции:  
1) Просмотр всех элементов базы данных  
2) Поиск среди всех элементов   
3) Редактирование элементов   
4) Удаление элементов

**Описание классов**

Class Muzeum

Описание экспоната, автор, возраст.

Class List

Оболочка над Muzeum, соединяет соседние элементы массива

Class Menu

Класс для работы с массивом List. Включается в себя функции для взаимодействия.

**Код**

**CLASSES.H**

#pragma once

#include<string>

#include<iostream>

#include<fstream>

using namespace std;

class muzei {

private:

string exponat;

string author;

int Age;

public:

muzei\* next;

muzei\* prev;

muzei() {

next = NULL;

prev = NULL;

Age = 0;

}

muzei(string exponat, string author, int Age) : exponat(exponat), author(author), Age(Age) {

next = NULL;

prev = NULL;

}

friend std::ostream& operator<< (std::ostream& out, const muzei& \_muzei)

{

out << "Exponat: " << \_muzei.exponat << '\n';

out << "Author: " << \_muzei.author << '\n';

out << "Age: " << \_muzei.Age << '\n';

return out;

}

friend std::istream& operator>> (std::istream& in, muzei& \_muzei)

{

in >> \_muzei.exponat;

in >> \_muzei.author;

in >> \_muzei.Age;

return in;

}

void SetExponat(string \_exponat) {

this->exponat = \_exponat;

}

string GetExponat() {

return this->exponat;

}

void SetAuthor(string \_author) {

this->author = \_author;

}

string GetAuthor() {

return this->author;

}

void SetAge(int \_Age) {

this->Age = \_Age;

}

int GetAge() {

return this->Age;

}

~muzei() {

}

};

class List {

public:

muzei\* head;

List() {

}

List(muzei ml){

muzei\* tmp = new muzei(ml);

head = tmp;

}

void ChangeAuthor(int index, string newauthor);

muzei\* SearchExponatByAuthor(string name);

muzei\* Search(int index);

void Add(string exponat, string author, int Age);

void Print();

void PrintFile(string file);

void LoadFile(string file);

void Del(int index);

void Free();

~List() {

}

};

class Menu {

public:

List list;

Menu() {

}

Menu(List list) : list(list) {

}

void Print(void);

void Add(void);

void Del(void);

void Search(void);

void Cls(void);

void Change(void);

};

**CLASSES.CPP**

#include"list.h"

muzei\* List::Search(int index) {

muzei\* finded = NULL, \* tmp = head;

int i = 0;

while (tmp != NULL) {

if (i == index) {

finded = tmp;

break;

}

tmp = tmp->next;

i++;

}

return finded;

}

void List::Add(string exponat, string author, int Age) {

muzei\* tmp = new muzei(exponat, author, Age);

tmp->next = head;

tmp->prev = NULL;

head->prev = tmp;

head = tmp;

}

void List::Print() {

muzei\* tmp = head;

while (tmp != NULL) {

cout << "Exponat: " << tmp->GetExponat() << '\n';

cout << "Author: "<< tmp->GetAuthor() << '\n';

cout << "Age: "<< tmp->GetAge() << '\n';

tmp = tmp->next;

}

}

void List::PrintFile(string file) {

ofstream out(file);

if (out.is\_open())

{

muzei\* tmp = head;

while (tmp != NULL) {

out << tmp->GetExponat() << '\n' << tmp->GetAuthor() << '\n' << tmp->GetAge() << '\n';

tmp = tmp->next;

}

}

else {

cout << "Can't open file " << file << '\n';

}

out.close();

}

void List::LoadFile(string file) {

ifstream in(file);

string \_exponat, \_author, \_age;

if (in.is\_open())

{

while (1)

{

if (getline(in, \_exponat) && getline(in, \_author) && getline(in, \_age)) {

int iage = stoi(\_age);

this->Add(\_exponat, \_author, iage);

}

else {

break;

}

}

in.close();

}

else {

cout << "Can't open file " << file << '\n';

}

}

void List::Del(int index) {

muzei\* finded = Search(index);

if (!finded)

return;

if (index == 0) {

this->head = this->head->next;

this->head->next->prev = NULL;

}

else {

if (finded->prev) {

finded->prev->next = finded->next;

}

if (finded->next) {

finded->next->prev = finded->prev;

}

}

delete finded;

}

void List::Free() {

muzei\* tmp = head;

while (tmp != NULL) {

muzei\* p = tmp;

tmp = tmp->next;

delete p;

}

}

void List::ChangeAuthor(int index, string newauthor) {

muzei\* finded = Search(index);

if (!finded) {

cout << "can't find index: " << index << '\n';

return;

}

finded->SetAuthor(newauthor);

}

muzei\* List::SearchExponatByAuthor(string author) {

muzei\* finded = NULL, \* tmp = head;

while (tmp != NULL) {

if (tmp->GetAuthor() == author) {

finded = tmp;

break;

}

tmp = tmp->next;

}

return finded;

}

void Menu::Cls(void)

{

system("cls");

}

void Menu::Print(void) {

list.Print();

}

void Menu::Add(void) {

string exponat, author;

int age;

cout << "Enter item: " << endl;

cout << "Exponat: ";

cin >> exponat;

cout << "Age: ";

cin >> age;

cout << "Author: ";

cin >> author;

list.Add(exponat, author, age);

}

void Menu::Del(void) {

int index;

cout << "enter index: " << '\n';

cin >> index;

list.Del(index);

}

void Menu::Search(void) {

string author;

cout << "Enter author: " << '\n';

cin >> author;

muzei\* tmp = list.SearchExponatByAuthor(author);

if (tmp)

cout << \*tmp << '\n';

else

cout << "Can't find exponat" << '\n';

}

void Menu::Change(void) {

int index;

cin >> index;

string author;

cout << "Enter new author: " << '\n';

cin >> author;

list.ChangeAuthor(index, author);

}

**MAIN.CPP**

#include<iostream>

#include<fstream>

#include<string>

#include"list.h"

using namespace std;

int main(int argc, char\*\* argv) {

if (argc < 2) {

cout << "Error: provide database filename\n";

return 1;

}

muzei monaliza("Mona\_Liza","Day\_Vinci",500);

List list(monaliza);

Menu menu(list);

menu.list.LoadFile(argv[1]);

char number = 0;

while (number != '6') {

cout << "\n";

cout << "Muzeum:\n";

cout << "1) Print all exponats \n";

cout << "2) Add exponats\n";

cout << "3) Search exponat \n";

cout << "4) Change author \n";

cout << "5) Delete exponat \n";

cout << "6) Exit \n";

cout << "\n";

cout << "Print number from 1 to 6:\n";

cin >> number;

menu.Cls();

switch (number) {

case '1':

menu.Print();

break;

case '2':

menu.Add();

break;

case '3':

menu.Search();

break;

case '4':

menu.Change();

break;

case '5':

menu.Del();

break;

case '6':

default:

break;

}

}

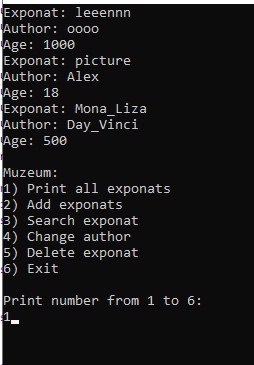
menu.list.PrintFile(argv[1]);

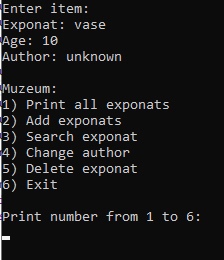
menu.list.Free();

return 0;

}

**Результат работы программы**

****

****

