**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА (САМАРСКИЙ УНИВЕРСИТЕТ)»**

ИНСТИТУТ ИНФОРМАТИКИ И КИБЕРНЕТИКИ

Кафедра программных систем

Дисциплина

**Объектно-ориентированное программирование**

**ОТЧЕТ**по лабораторной работе № 3  
 **Объектно-ориентированное программирование обработки строк с использованием функций стандартной библиотеки языка С++ в стиле языка С.**  
Вариант № 3

Студент: Боряков Никита  
Группа: 6102-020302D  
  
Преподаватель: Баландин А.В.  
  
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2022

**Задание**

Реализовать информационную базу, состоящую из трех файлов: FIO.DAT - содержит в виде символьных строк, оканчивающихся , список фамилий; DOLGH.DAT - содержит в виде символьных строк наименования должностей, которые не повторяются; LINK.IDX - содержит записи вида struct {unsigned long name\_idx; //индекс записи в файле FIO.DAT; unsigned long post\_idx; //индекс записи в файле DOLGH.DAT}, связывающие между собой фамилию и должность. В качестве индексов используются значения позиций, с которых начинаются записи в соответствующих файлах. Записи в FIO.DAT и в LINK.IDX взаимно-однозначно соответствуют друг другу. Программа должна обеспечивать выполнение следующих функций: 1) включение новой фамилии с указанием 41 32 должности, 2) добавление новой должности, 3) по указанной фамилии вывод на экран занимаемой должности.

**Информационная модель предметной области**

Информационный объект База данных;

* Свойства:
* Поток к файлу с фамилиями FIO.DAT;
* Поток к файлу с должностями DOLGH.DAT;
* Поток к файлу со связками LINK.IDX;
* Структура Worker содержащая индекс фамилии и индекс должности в файле LINK.IDX;
* Процедуры:
* Конструктор без параметра;
* Добавление фамилии в базу с указанием существующей должности;
* Добавление новой должности;
* Вывод должности по фамилии (фамилия и должность);
* Процедура проверки существования фамилии или должности в файле;
* Деструктор.

**Структура алгоритма программы**

1. Создание файлов;
2. Ввод новой, несуществующей в базе, должности;
3. Добавление фамилии с указанием должности;
4. Вывод должности по указанной фамилии;
5. Завершение программы;

**Код программы**

**Database.h**

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

class Database

{

private:

FILE\* lastNameStream; // поток к файлу с фамилиями FIO.DAT

FILE\* postStream; // поток к файлу с должностями DOLGH.DAT

FILE\* linkStream; // поток к файлу со связками LINK.IDX

struct Worker

{

unsigned long lastname\_idx; //индекс фамилии в файле FIO.DAT;

unsigned long post\_idx; //индекс должности в файле DOLGH.DAT

};

public:

void AddLastname(char\* lastname, char\* post) // добавление фамилии и связывание с L

{

int postPosition;

bool postExist = ChekingRecExist(postStream, post, postPosition);

if (!postExist) {

puts("\nУказанной должности не существует, сначала добавьте её в базу.");

return;

}

if (strchr(lastname, '\n') == 0)

strcat(lastname, "\n");

int lastname\_index;

fputs(lastname, lastNameStream);

ChekingRecExist(lastNameStream, lastname, lastname\_index);

Worker worker;

worker.lastname\_idx = lastname\_index;

worker.post\_idx = postPosition;

fwrite(&worker, sizeof(worker), 1, linkStream);

}

void AddPost(char\* post) // добавление должности

{

if (strchr(post, '\n') == NULL)

strcat(post, "\n");

int postPos;

bool postExist = ChekingRecExist(postStream, post, postPos);

if (postExist) {

puts("\nТакая должность уже существует.");

return;

}

fputs(post, postStream);

}

void PrintPostByLastname(char\* lastname) // вывод должности по указанной фамилии

{

int lastnamePosition;

bool postExist = ChekingRecExist(lastNameStream, lastname, lastnamePosition);

if (!postExist) {

puts("\nУказанной фамилии не существует, сначала добавьте её в базу.");

return;

}

fseek(linkStream, 0, SEEK\_SET);

Worker worker;

int i = 0;

while (fread(&worker, sizeof(worker), 1, linkStream)) //проверить , что будеи если вместо worker вставить Worker

{

if (worker.lastname\_idx == lastnamePosition) {

fseek(postStream, worker.post\_idx, SEEK\_SET);

char\* reading = fgets((char\*)malloc(sizeof(char) \* BUFSIZ), BUFSIZ - 1, postStream);

puts(reading);

++i;

}

}

}

Database()

{

lastNameStream = fopen("FIO.DAT", "a+b");

postStream = fopen("DOLGH.DAT", "a+b");

linkStream = fopen("LINK.IDX", "a+b");

}

~Database()

{

fclose(lastNameStream);

fclose(postStream);

fclose(linkStream);

}

private:

// показывает есть ли в файле строка с заданной фамилией или должностью

// возвращает индекс в фамилии или должности в файле

bool ChekingRecExist(FILE\* filestream, char\* record, int& position)

{

fseek(filestream, 0, SEEK\_SET);

if (strchr(record, '\n') == 0)

strcat(record, "\n");

int currentPos = 0;

while (true)

{

currentPos = ftell(filestream);

char\* reading = fgets((char\*)malloc(sizeof(char) \* BUFSIZ), BUFSIZ - 1, filestream);

if (feof(filestream)) //если будет достигнут конец файла, то выйдем из цикла

break;

int res = \_stricmp(record, reading);

if (res == 0) {

position = currentPos;

return true;

}

}

return false;

}

};

**Main.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma once

#include <iostream>

#include <Windows.h>

#include "Database.h"

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Database database;

char buffer\_lastname[BUFSIZ];

char buffer\_post[BUFSIZ];

char menu;

do {

system("cls");

puts("Меню:");

puts("0 - Выход");

puts("1 - Добавить новую фамилию с указанием должности");

puts("2 - Добавить должность");

puts("3 - Вывести должность по фамилии");

menu = getchar();

rewind(stdin); //Устанавливает внутренний указатель положения файла в начальное положение(эквивалент fseek(fs, 0, SEEK\_SET)

putchar('\n');

switch (menu) {

case '1': {

puts("Введите новую фамилию: ");

gets\_s(buffer\_lastname);

puts("Введите существующую должность: ");

gets\_s(buffer\_post);

database.AddLastname(buffer\_lastname, buffer\_post);

break;

}

case '2': {

puts("Введите новую должность: ");

gets\_s(buffer\_lastname);

database.AddPost(buffer\_lastname);

break;

}

case '3': {

puts("Введите фамилию: ");

gets\_s(buffer\_lastname);

puts("\nДолжность :");

database.PrintPostByLastname(buffer\_lastname);

break;

}

}

putchar('\n');

system("pause");

} while (menu != '0');

return 0;

}

**Пример выполнения программы**

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание