Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №13

По дисциплине: «Основы алгоритмизации и программирования»

**Тема:** **«Модульное программирование»**

Вариант №6

Выполнил:

Студент 1-го курса

Группы ПО-7

Комиссаров А.Е.

Проверила:

Войцехович О. Ю.

Брест 2021

**Цель работы:** изучить принципы модульного программирования в Си; ознакомиться с основными возможностями межмодульного взаимодействия.

**Текст программы:**

**Файл Source.cpp:**

#include "Header.h"

int main(){

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n = 0, choice;

worker\* work = new worker[100];

do

{

menu();

cin >> choice;

switch (choice)

{

case 1: {

input\_all(work, n);

}

break;

case 2: {

output\_all(work, n);

}

break;

case 3: {

index\_sort(work, n);

} break;

case 4: {

select\_delete(work, n);

}break;

case 5: {

revenue\_all(work, n);

}break;

case 6: {

cout << "Выход из программы..." << endl;

}break;

default: {cout << "Неверный ввод!" << endl; }

}

} while (choice != 6);

delete[]work;

return 0;

}

**Файл Header.h:**

#pragma once

#include <iostream>

#include <string>

#include <fstream>

#include <Windows.h>

using namespace std;

struct worker

{

string fio;

union { int tabnum; }u\_tabnum;

int tarif;

float time;

int index;

};

enum tarif\_type { low = 1, avg = 2, high = 3 };

void menu();

void input\_all(worker\* work, int& N);

void output\_all(worker\* work, int& n);

void revenue\_all(worker\* work, int& n);

void select\_delete(worker\* work, int& n);

int gettarif(int i, worker\* work);

void cout\_info(worker\* work, int& N);

void index\_sort(worker work[], int N);

**Файл Header.cpp:**

#include "Header.h"

void menu() {

cout << endl << "Меню: " << endl;

cout << "1-Ввод данных;" << endl;

cout << "2-Вывод данных;" << endl;

cout << "3-Сортировка по кол-ву часов;" << endl;

cout << "4-Удаление по заданному значению поля;" << endl;

cout << "5-Зароботная плата сотрудников;" << endl;

cout << "6-Выход." << endl;

cout << endl << "Ваш выбор: ";

}

void input\_all(worker\* work, int& N) {

ofstream out;

out.open("D:\Workers.txt", ios::out);

if (!out.is\_open()) { cout << "Не удалось открыть файл" << endl; }

else {

cout << "Введите количество рабочих: ";

cin >> N;

worker\* work2 = new worker[N];

for (int i = 0; i < N; i++) {

cout << "[" << i + 1 << "] - - - - - - - - - - - - - - -" << endl;

cout << "Введите ФИО рабочего: "; cin.ignore(); getline(cin, work2[i].fio);

cout << "Введите табельный номер рабочего: "; cin >> work2[i].u\_tabnum.tabnum;

cout << "Введите тариф (1 - 5руб/ч, 2 - 25руб/ч, 3 - 50руб/ч): "; cin >> work2[i].tarif;

cout << "Введите время работы: "; cin >> work2[i].time;

work[i] = work2[i];

}

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

for (int i = 0; i < N; i++) { out.write((char\*)&(work[i]), sizeof(worker)); }

}

out.close();

}

void output\_all(worker\* work, int& n) {

int n2;

worker\* work2 = new worker[n];

ifstream in("D:\Workers.txt", ios::binary);

if (!in.is\_open()) { cout << "Файл отсутствует" << endl; }

else {

in.read((char\*)&n2, sizeof(int));

for (int i = 0; i < n2; i++) { in.read((char\*)&(work[i]), sizeof(worker)); }

in.close();

cout << "Содержимое файла: " << endl;

for (int i = 0; i < n2; i++)

{

cout << "Рабочий №" << i + 1 << " - ";

cout << "ФИО:" << work[i].fio << ", тариф:" << gettarif(i, work) << ", время работы: " << work[i].time << ", табельный №: " << work[i].u\_tabnum.tabnum << ";" << endl;

}

}

n = n2;

delete[] work2;

}

void revenue\_all(worker\* work, int& n) {

cout << "Зарплата сотрудников фирмы:" << endl;

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

cout << "Рабочий №" << i + 1 << " - ";

cout << "ФИО:" << work[i].fio << ", зарплата:" << gettarif(i, work) \* work[i].time << ";" << endl;

}

}

void select\_delete(worker\* work, int& n) {

int del;

cout << "Введите кол-во часов при котором стоит удалить сотрудника(-ов):"; cin >> del;

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

if (work[i].time == del) { for (int j = i; j < n; j++) { work[j] = work[j + 1]; } i--; n--; }

}

}

int gettarif(int i, worker\* work) {

int val = work[i].tarif;

if (val == 1 || val == 2 || val == 3) {

if (val == low) { return 5; }

if (val == avg) { return 25; }

if (val == high) { return 50; }

}

else return val;

}

void cout\_info(worker\* work, int& N) { //вывод элементов по индексам из файла

int i;

ifstream fin("D:\index.txt", ios::in);

if (!fin.is\_open()) {

cout << "Ошибка открытия файла. (1)" << endl; return;

}

else {

for (int j = 0; j < N; j++) {

fin >> i;

cout << "Рабочий №" << j + 1 << " - ";

cout << "ФИО:" << work[i].fio << ", тариф:" << gettarif(i, work) << ", время работы: " << work[i].time << ", табельный №: " << work[i].u\_tabnum.tabnum << ";" << endl;

}

}

}

void index\_sort(worker work[], int N) { //сортировка элементов и запись индексов в файл

if (N == 0) { cout << "Нечего сортировать!" << endl; }

else {

for (int i = 0; i < N; i++) { work[i].index = i; }//присваивание индексов элементов в соответствующее поле

//далее сортировка

worker\* tmp = new worker[N];

worker max;

for (int i = 0; i < N; i++)//копирование во временный массив

{

tmp[i] = work[i];

}

for (int i = N - 1; i >= 0; i--)//сортировка

{

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

{

if (tmp[j].time > tmp[j + 1].time)

{

worker temp = tmp[j];

\*(tmp + j) = \*(tmp + j + 1);

\*(tmp + j + 1) = temp;

}

}

}

//конец сортировки

//далее вывод индексов в файл

ofstream fout;

fout.open("D:\index.txt", ofstream::trunc);

if (!fout.is\_open())

{

cout << "Ошибка открытия файла.";

return;

}

else

{

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

{

fout << tmp[i].index << '\n';

}

}

fout.close();

delete[] tmp;

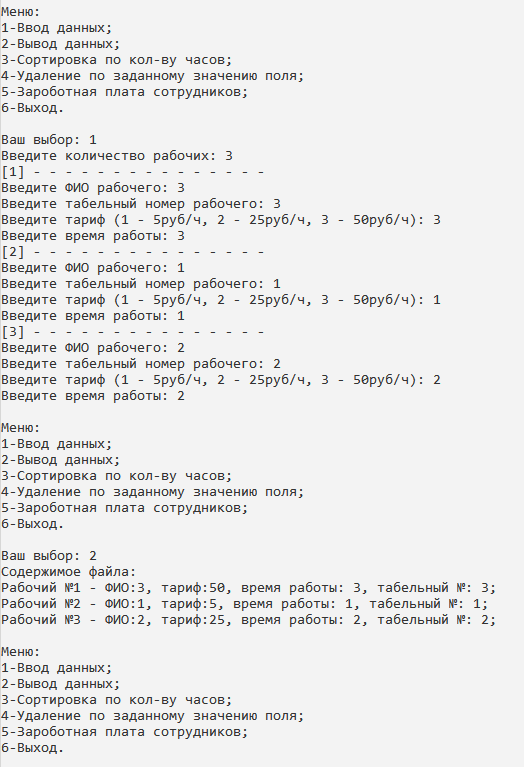
//конец вывода

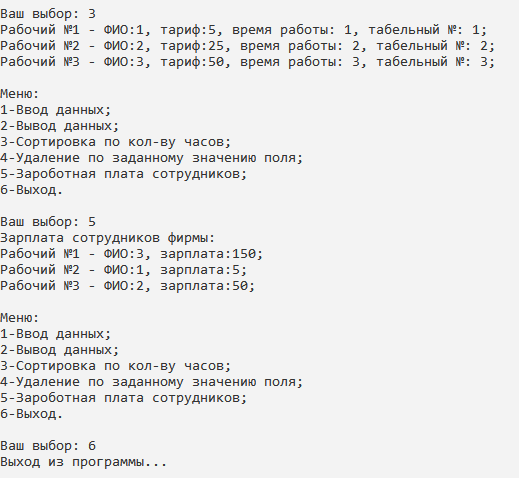
}

cout\_info(work, N);//пользовательская функция описанная ранее (вывод по индексам)

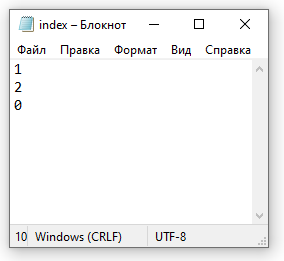
}

**Результаты:**

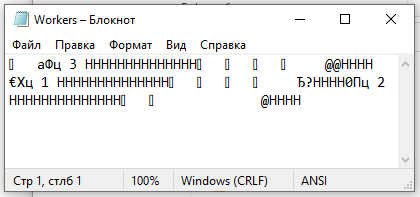
****



В индексном файле записалось как:



Файл workers.txt с информацией массива:



**Вывод:** изучил принципы модульного программирования в Си; ознакомился с основными возможностями межмодульного взаимодействия.