МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

По лабораторной работе №4

Выполнил:

Студент группы ИИ-22

Копанчук Евгений Романович

Проверил:

Войцехович О. Ю.

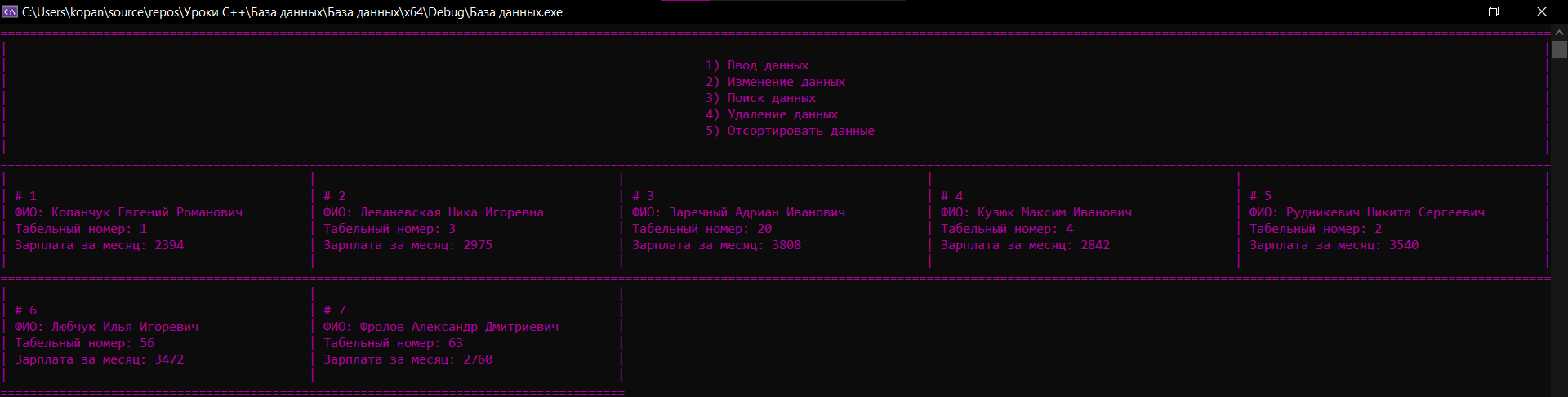
Брест 2022

**Вариант 6**

В программу разработанную в лабораторной работе 12 внести следующие изменения и дополнения:

1. 1. При запуске программы данные читаются в массив структур из файла, при добавлении новой записи в массив структур в файл должна дописываться новая запись, без изменения остальных записей.
2. 2. Все изменения (изменения полей записи, удаление записи) – сохраняются в файле при помощи перезаписи содержимого всего файла.
3. 3. Сортировка должна выполняться по двум полям на выбор при помощи создания индексных файлов. Содержимое индексного файла переписывается в случае изменения значения ключевого поля (поля, по которому выполняется сортировка) или в случае удаления, добавления записей.

Главное меню программы выглядит следующим образом. Данные располагаются таблицей в 5 колонок. Вывод происходи слева на право, сверху вниз.



Добавлен индексный файл, в который заносится последовательность из номеров элементов массива структур (данных). При каждом изменении массива данных (удаление, добавление и изменение) запускается перепись индексного файла. Далее в зависимости от выбранного варианта (переключается в пункте меню 5) осуществляется вывод данных с соответствующей последовательности. Реализовано это с помощью двух новых функций StoreIndex и ReadIndex и некоторыми поправками в функциях main и Menu.

Код функций для индексного файла:

inline void StoreIndex(Data\* d, int amount, int\* (&s1), int\* (&s2), int\* (&s3), string filename) {

if (amount != 0) {

Data\* \_d = new Data[amount];

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

\_d[i].FIO = d[i].FIO;

\_d[i].Number = d[i].Number;

\_d[i].FullWage.Wage = d[i].FullWage.Wage;

s1[i] = i;

s2[i] = i;

s3[i] = i;

}

ofstream INDEX;

INDEX.open(filename, ios::out);

if (!INDEX) {

cerr << "Ошибка. Файл INDEX.txt не может быть открыт!" << endl;

system("pause");

}

string sTemp;

int iTemp;

for (int i = 0; i < amount - 1; i++) {

for (int j = 0; j < amount - i - 1; j++) {

// Сортировка по Фамилии

if (\_d[j].FIO[0] > \_d[j + 1].FIO[0]) {

sTemp = \_d[j].FIO;

\_d[j].FIO = \_d[j + 1].FIO;

\_d[j + 1].FIO = sTemp;

iTemp = s1[j];

s1[j] = s1[j + 1];

s1[j + 1] = iTemp;

}

// Сортировка по табельному номеру

if (Parsing(\_d[j].Number) > Parsing(\_d[j + 1].Number)) {

sTemp = \_d[j].Number;

\_d[j].Number = \_d[j + 1].Number;

\_d[j + 1].Number = sTemp;

iTemp = s2[j];

s2[j] = s2[j + 1];

s2[j + 1] = iTemp;

}

// Сортировка по зарабной плате

if (\_d[j].FullWage.Wage > \_d[j + 1].FullWage.Wage) {

iTemp = \_d[j].FullWage.Wage;

\_d[j].FullWage.Wage = \_d[j + 1].FullWage.Wage;

\_d[j + 1].FullWage.Wage = iTemp;

iTemp = s3[j];

s3[j] = s3[j + 1];

s3[j + 1] = iTemp;

}

}

}

delete[] \_d;

for (int i = 0; i < amount; i++) INDEX << s1[i] << endl;

for (int i = 0; i < amount; i++) INDEX << s2[i] << endl;

for (int i = 0; i < amount; i++) INDEX << s3[i] << endl;

INDEX.close();

}

}

inline void ReadIndex(int amount, int\* (&s1), int\* (&s2), int\* (&s3), string filename) {

ifstream INDEX;

INDEX.open(filename, ios::out);

if (!INDEX) {

cerr << "Ошибка. Файл INDEX.txt не может быть открыт!" << endl;

system("pause");

}

for (int i = 0; i < amount; i++) INDEX >> s1[i];

for (int i = 0; i < amount; i++) INDEX >> s2[i];

for (int i = 0; i < amount; i++) INDEX >> s3[i];

INDEX.close();

}

В функции main создаём массивы для хранения считаных последовательностей индексов (s1, s2, s3) и последовательности, используемой для вывода (\_s).

// Создаём массив для индексов

int\* s1 = new int[amount];

int\* s2 = new int[amount];

int\* s3 = new int[amount];

int\* \_s = new int[amount];

for (int i = 0; i < amount; i++) \_s[i] = i;

Изменения в выводе прогаммы: теперь мы обращаемся к элементам массива структур не по порядку, а к элементу с индексом [\_s[j]].

// Вывод базы данных

int NumOfData = 1;

if (amount != 0) {

for (int i = 0; i < amount; i += 5) {

string spacebar1 = "|", spacebar2 = "|", buffer1 = "|", buffer2 = "|", buffer3 = "|", buffer4 = "|", linebar = "=";

for (int j = i; j < i + 5 && j < amount; j++) {

NumOfData++;

spacebar1 += RightSewt("|", 42);

buffer1 += LeftSewt(" # " + ConvertToString(j + 1), 41) + "|";

buffer2 += LeftSewt(" ФИО: " + d[\_s[j]].FIO, 41) + "|";

buffer3 += LeftSewt(" Табельный номер: " + d[\_s[j]].Number, 41) + "|";

buffer4 += LeftSewt(" Зарплата за месяц: " + ConvertToString(d[\_s[j]].FullWage.Wage), 41) + "|";

spacebar2 += RightSewt("|", 42);

for (int l = 0; l < 42; l++) linebar += '=';

}

cout << LeftSewt(spacebar1, 211);

cout << LeftSewt(buffer1, 211);

cout << LeftSewt(buffer2, 211);

cout << LeftSewt(buffer3, 211);

cout << LeftSewt(buffer4, 211);

cout << LeftSewt(spacebar2, 211);

cout << linebar;

}

cout << endl;

}

Перезапись массива \_s происходит следующим образом:

case 5:

sort = ++sort % 4;

switch (sort) {

case 0:

for (int i = 0; i < amount; i++) \_s[i] = s1[i];

break;

case 1:

for (int i = 0; i < amount; i++) \_s[i] = s2[i];

break;

case 2:

for (int i = 0; i < amount; i++) \_s[i] = s3[i];

break;

default:

for (int i = 0; i < amount; i++) \_s[i] = i;

}

system("cls");

Menu(d, amount, s1, s2, s3, \_s, sort);

break;