МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УО «ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра технологий программирования

КУРСОВАЯ РАБОТА

По дисциплине «Основы алгоритмизации и программирования»

Тема: «База данных сотрудников дошкольного учреждения образования»

|  |  |
| --- | --- |
| Выполнила: | студентка гр.15-ИТ-1, Рутковская Н.Ю. |
| Проверил: | преподаватель-стажер, А.В. Бойченко |

Новополоцк,

2016 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc421140012)

[1 АНАЛИЗ ЗАДАНИЯ И ПОСТАНОВКА ЗАДАЧИ 7](#_Toc421140013)

[2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 8](#_Toc421140014)

[3 ПРОЕКТИРОВАНИЕ ПРОГРАММЫ 11](#_Toc421140015)

[4 РЕАЛИЗАЦИЯ ПРОГРАММЫ 13](#_Toc421140016)

[5 ТЕСТИРОВАНИЕ ПРОГРАММЫ 15](#_Toc421140017)

[6 ЗАКЛЮЧЕНИЕ 19](#_Toc421140018)

[СПИСОК ЛИТЕРАТУРЫ 29](#_Toc421140019)

[ПРИЛОЖЕНИЕ А 30](#_Toc421140020)

[ПРИЛОЖЕНИЕ Б 31](#_Toc421140021)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

Рутковская Н.Ю. 1510647

Разраб.

Рутковская Н.Ю.

Провер.

Бойченко А.В.

Реценз.

Н. Контр.

Утверд.

.

Информационные системы

«Магазин игрушек»

Лит.

Листов

48

УО «ПГУ» 15-ИТ-1

**ВВЕДЕНИЕ**

**Язык программирования C** (рус. *Си*) — компилируемый статически типизированный язык программирования общего назначения, разработанный в 1969—1973 годах как развитие языка Би, когда Кен Томпсон и Дэннис Ритчи из Bell Labs разрабатывали операционную систему UNDC. Первоначально был разработан для реализации операционной системы UNIX, но, впоследствии, был перенесён на множество других платформ. Согласно дизайну языка Си, его конструкции близко сопоставляются типичным машинным инструкциям, благодаря чему он нашёл применение в проектах, для которых был свойственен язык ассемблера, в том числе как в операционных системах, так и в различном прикладном ПО для множества устройств — от суперкомпьютеров до встраиваемых систем. Язык программирования Си оказал существенное влияние на развитие индустрии программного обеспечения, а его синтаксис стал основой для таких языков программирования, как C++, C#, Java и Objective-C.

Сначала они создали часть компилятора С, затем использовали ее для компиляции остальной части компилятора С и, наконец, применили полученный в результате компилятор для компиляции UNIX. Операционная система UNIX первоначально распространялась в исходных кодах на С среди университетов и лабораторий, а получатель мог откомпилировать исходный код на С в машинный код с помощью подходящего компилятора С.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

4

Курсовая работа

Распространение исходного кода сделало операционную систему UNIX уникальной; программист мог изменить операционную систему, а исходный код мог быть перенесен с одной аппаратной платформы на другую. Сегодня стандарт POSIX определяет стандартный набор системных вызовов UNIX, доступных в С, которые должны быть реализованы в версиях UNIX, являющихся POSIX-совместимыми. С был третьим языком, который разработали Томсон и Ритчи в процессе создания UNIX; первыми двумя были, разумеется, А и В.

По сравнению с более ранним языком — BCPL, С был улучшен путем добавления типов данных определенной длины. Например, тип данных int мог применяться для создания переменной с определенным числом битов (обычно 16), в то время как тип данных long мог использоваться для создания целой переменной с большим числом битов (обычно 32). В отличие от других языков высокого уровня, С мог работать с адресами памяти напрямую с помощью указателей и ссылок. Поскольку С сохранил способность прямого доступа к аппаратному обеспечению, его часто относят к языкам среднего уровня или в шутку называют "мобильным языком ассемблера".

Что касается грамматики и синтаксиса, то С является структурным языком программирования. В то время как многие современные программисты мыслят в категориях классов и объектов, программисты на С думают в категориях процедур и функций. В С можно определить собственные абстрактные типы данных, используя ключевое слово struct. Аналогично можно описывать собственные целые типы (перечисления) и давать другие названия существующим типам данных при помощи ключевого слова typedef. В этом смысле С является структурным языком с зародышами объектно-ориентированного программирования.

Широкое распространение языка C на различных типах компьютеров (иногда называемых аппаратными платформами) привело, к сожалению, ко многим вариациям языка. Они были похожи, но несовместимы друг с другом. Это было серьезной проблемой для разработчиков программ, нуждавшихся в написании совместимых программ, которые можно было бы выполнять на нескольких платформах. Стало ясно, что необходима стандартная версия C. В 1983г. ANSI (Американский Национальный Комитет Стандартов) сформировал технический комитет для создания стандарта языка C (чтобы "обеспечить недвусмысленное и машинно-независимое определение языка"). В 1989 стандарт был утвержден. ANSI скооперировался с ISO (Международной Организацией Стандартов), чтобы стандартизовать C в международном масштабе; совместный стандарт был опубликован в 1990 году и назван ANSI/ISO 9899:1990. Этот стандарт совершенствуется до сих пор и поддерживается большинством фирм разработчиков компиляторов.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

5

Курсовая работа

В наше время, компьютер стал частью повседневной жизни. Он используется как для отдыха, так и для работы; заменяет собой бумагу, калькулятор и еще множество мелких и не очень вещей. Компьютеры окружают нас везде: в банках, магазинах, на почте. Компьютер позволяет производить сложные расчёты с минимальными затратами времени, а так же быстро находить нужную нам информацию, тем самым облегчая труд и повышая производительность.

В рамках данного курсового проекта попыталась организовать базу данных магазина мягких игрушек.

***Цель изучения и задачи курса «Основы алгоритмизации и программирования»***

Овладеть знаниями: теоретические основы алгоритмизации задач и проектирования программ; основы организации вычислительных процессов на ЭВМ; приемы проектирования программ на современном языке высокого уровня.

Научиться: характеризовать исходные данные решения задач; характеризовать выходные данные решения задач; уметь анализировать решения задач, а также возможные методы решения задач и осуществлять обоснованный выбор.

Приобрести навыки: формальной математической постановки задачи; алгоритмизации задач и программирования, отладки и выполнения на ЭВМ конкретных задач с использованием современных методов программирования.

Изучаемые темы: структуры, объединения, указатели и динамическая память, объявление указателей; процедуры и функции для работы с динамической памятью; файлы: понятие файла, виды файлов, функции для работы с файлами; алгоритмы сортировки и поиска; списки с разными видами хранения.

Задачи курса сводятся к изучению:

* истории языков программирования и их развития;
* языка программирования высокого уровня С;
* системы простых (скалярных) и сложных (векторных) типов данных языков программирования высокого уровня на примере языка С;
* методов реализации разветвляющихся и циклических алгоритмов с использованием операторов управления языков программирования высокого уровня на примере языка С;
* указателей, принципов адресной арифметики на примере языка С;
* принципов «нисходящего» и «восходящего» проектирования программ на примере реализации функций на языке С;
* алгоритмов сортировки и поиска;
* динамической памяти и сложноорганизованных динамических структур данных;
* принципов модульного программирования;

Изм.

Лист

№ докум.

Подпись

Дата

Лист

6

Курсовая работа

* принципов «правильного» программирования.

**1 АНАЛИЗ ЗАДАНИЯ И ПОСТАНОВКА ЗАДАЧИ**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

7

Курсовая работа

Цель курсовой работы является создание программы, которая содержала бы базу данных сотрудников дошкольного учреждения образования. Основная функция программы – работа с одно файловой базой данных. Каждый сотрудник должен быть описан по следующим критериям (которые более подробно характеризуются в скобках, т.к. для работы с программой, нужно лучше знать, что должно содержаться в каждом из них):

1. Фамилия, имя, отчество сотрудника (например: Иванов Иван Иванович).
2. Дата рождения (пишется в формате дд.мм.гггг).
3. Отдел (имеется в виду отдел, к которому относится сотрудник. В дошкольном учреждении образования есть два отдела: педагог и технический персонал).
4. Должность (вводится должность сотрудника).
5. Стаж работы (количество полных лет, которые отработал человек в данной сфере за всю жизнь до сего момента).

Программа должна иметь дружественный интерфейс и проводить проверку на правильность ввода данных. Программа должна осуществлять:

* Запись и загрузку файла базы данных (текстовый файл).
* Добавление новых записей, удаление старых.
* Сортировать записи по любому из полей базы данных.
* Перемещать запись из старой позиции в новую.
* Отображать весь список записей в консоль.
* Отображать информацию по одной записи.

**2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**Стек** (англ. ***stack*** — стопка; читается стэк) — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»).

Чаще всего принцип работы стека сравнивают со стопкой тарелок: чтобы взять вторую сверху, нужно снять верхнюю. В цифровом вычислительном комплексе стек называется магазином — по аналогии с магазином в огнестрельном оружии (стрельба начнётся с патрона, заряженного последним).

В 1946 Алан Тьюринг ввёл понятие стека. А в 1957 году немцы Клаус Самельсон и Фридрих Л. Бауэр запатентовали идею Тьюринга.

Зачастую стек реализуется в виде однонаправленного списка (каждый элемент в списке содержит помимо хранимой информации в стеке указатель на следующий элемент стека).

Но также часто стек располагается в одномерном массиве с упорядоченными адресами. Такая организация стека удобна, если элемент информации занимает в памяти фиксированное количество слов, например, 1 слово. При этом отпадает необходимость хранения в элементе стека явного указателя на следующий элемент стека, что экономит память. При этом указатель стека (*Stack Pointer*, — SP) обычно является регистром процессора и указывает на адрес головы стека.

Предположим для примера, что голова стека расположена по меньшему адресу, следующие элементы располагаются по нарастающим адресам. При каждом вталкивании слова в стек, SP сначала уменьшается на 1 и затем по адресу из SP производится запись в память. При каждом извлечении слова из стека (выталкивании) сначала производится чтение по текущему адресу из SP и последующее увеличение содержимого SP на 1.

При организации стека в виде однонаправленного списка значением переменной стека является указатель на его вершину — адрес вершины. Если стек пуст, то значение указателя равно NULL.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

8

Курсовая работа

**Указатель –** этопеременная, значением которой является адрес, по которому располагаются данные.

**Типизированный указатель** – указатель, содержащий адрес данных определенного типа (системного или пользовательского).

**Не типизированный указатель** – указатель, содержащий адрес данных неопределенного типа (просто адрес).

Классификация по области доступа определяется методом адресации принятой для семейства процессоров x86: адрес состоит из двух элементов: сегмент и смещение.

**Адрес** – это номер ячейки памяти, в которой или с которой располагаются данные.

Классифицировать указатели можно:

* по типу данных (типизированные и не типизированные указатели);
* по области доступа (ближние и дальние указатели).

**Динамическая память** – это область (блок) памяти выделенный для нужд программы в процессе работы программы (а не заранее).

Основными двумя действиями над динамической памятью являются: выделение и освобождение. В языке С функции для осуществления этих действий описаны в библиотеке **stdlib.h**.

**Структура** – это сложный тип данных представляющий собой упорядоченное в памяти множество элементов различного типа. Каждый элемент в структуре имеет свое имя и называется полем. Размер структуры определяется суммой размеров всех элементов.

**Файл** – именованная область данных на каком-либо носителе информации (жесткий диск, дискета, компакт-диск и т.д. и т.п.).

Язык С, как и другие языки программирования высокого уровня, позволяет осуществлять операции файлового ввода и вывода. Основной алгоритм обработки файлов выполняется в три действия в следующем порядке: открытие файла; чтение и/или запись данных в файл; закрытие файла.

В языке С все файлы делятся на два вида:

* бинарные;
* текстовые.

**Бинарный файл** – файл, содержащий структурированную или не структурированную информацию, представленную в двоичном (бинарном) виде.

**Текстовый файл** – файл, содержащий структурированную или не структурированную информацию, представленную в текстовом (ASCII символы) виде.

Операции чтения и записи для текстовых и бинарных файлов отличаются друг от друга (в языке С реализованы в виде различных функций). В языке С для работы с файлами набор функций реализован в библиотеке stdio.h.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

9

Курсовая работа

**Функция** – это синтаксически выделенный именованный программный модуль, выполняющий определенное действие или группу действий. Каждая функция имеет свой интерфейс и реализацию.

**Реализация функции** – тело функции, содержащее внутренние (локальные) данные функции и программный код, выполняющий действия согласно переданным в функцию параметрам и возвращающий значение, соответствующего интерфейсу функции типа.

**3 ПРОЕКТИРОВАНИЕ ПРОГРАММЫ**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

11

Курсовая работа

Для реализации программы будем использовать стек.

При открытии программы, пользователь увидит меню с пунктами, содержащими предлагаемые к выполнению действия:

1. Отображение всего списка;
2. Добавление:
   1. Добавление в начало;
   2. Добавление в конец;
   3. Добавление в позицию.
3. Удаление:
   1. Удаление с начала;
   2. Удаление с конца;
   3. Удаление в позиции;
   4. Очистка списка.
4. Отображение элемента:
   1. Отображение первого;
   2. Отображение последнего;
   3. Отображение в позиции.
5. Перемещение;
6. Сортировка по всем полям;
7. Сохранение в текстовый файл и выход.

Если пользователь набирает на клавиатуре 1, то ему открывается таблица, содержащая всех сотрудников и данные по ним.

При нажатии 2 - пользователю открывается дополнительное меню для выбора места добавления нового сотрудника в список с дальнейшим его добавлением.

* При нажатии 1 – пользователь производит добавление сотрудника в начало списка с заполнением всех данных.
* При нажатии 2 – пользователь производит добавление сотрудника в конец списка с заполнением всех данных.
* При нажатии 3 – пользователь вводит позицию, в которую производит добавление сотрудника с заполнением всех данных.

При нажатии 3 – пользователю открывается дополнительное меню для выбора элемента для удаления из списка с дальнейшим его удалением.

* При нажатии 1 – пользователь производит удаление сотрудника с начала списка.
* При нажатии 2 – пользователь производит удаление сотрудника с конца списка.
* При нажатии 3 – пользователь вводит позицию, из которой производит удаление сотрудника.
* При нажатии 4 – пользователь производит очистку списка.

При вводе 4 – пользователю открывается дополнительное меню для выбора сотрудника, по которому желает просмотреть данные.

* При нажатии 1 – пользователь открывает все данные по первому в списке сотруднике.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

12

Курсовая работа

* При нажатии 2 – пользователь открывает все данные по последнему в списке сотруднику.
* При нажатии 3 – пользователь вводит позицию сотрудника в списке, по которому желает просмотреть данные.

При выборе 5 – пользователь производит перемещение элемента списка из исходной позиции в новую.

При выборе 6 – пользователю открывается дополнительное меню, для выбора поля сортировки и дальнейшей сортировкой.

При выборе 7-ого пункта меню, пользователю открывается дополнительное меню с выбором сохранения в файл и выходом.

**4 РЕАЛИЗАЦИЯ ПРОГРАММЫ**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

13

Курсовая работа

Для реализации программы были создана следующая структура данных:

* Структура **HUMAN**, характеризирующая своими полями каждый элемент и содержащая указатель на следующий элемент:

typedef struct HUMAN

{

struct FullName

{

char surname[15];

char name[15];

char patronymic[15];

} FN;

struct Data

{

int day;

int month;

int year;

} DATA;

char office[50];

char job[50];

int experience;

struct HUMAN \*next;

} HUMAN;

Листинг 1 – структура HUMAN

int add(void);//Создание

int add\_l(void);//Ввод в конец

int add\_p(int);//Ввод в позицию

int del\_last(void);//Удаление

int del\_last\_buf(void);//Удаление из буфера

int in\_buf\_stack(void);//Ввод в буфер

int out\_buf\_stack(void);//Вывод из буфера

int print\_all(void);//Вывод всех

int print\_f(void);//Вывод первого

int print\_l(void);//вывод последнего

int print\_p(int);//Вывод по позиции

int del\_f(void);//Удаление последнего

int del\_p(int);//Удаление из позиции

int del\_full(void);//Полная очистка

int replace(int, int);//Перемещение

int sort\_fn(void);//Сортировка по ФИО

int sort\_date(void);//Сортировка по дате

int sort\_office(void);//Сортировка по отделу

int sort\_job(void);//Сортировка по должности

int sort\_experience(void);//Сортировка по стажу

int copy(HUMAN\*, HUMAN\*);//Копирование

int out\_file(void);//Вывод в файл с псевдографикой

int out\_file2(void);//Вывод в файл

int file\_pop(void);//Чтение из файла

int in\_buf\_buf(HUMAN\*, int);

int out\_buf\_buf(HUMAN\*);

Листинг 2 – список функций модуля **stack.c**

Модуль **main.c** :

*main* – содержит структуру данных и запускает функцию *menu();* из модуля **menu.c**

int main(int argc, char \*argv[])

{

setlocale(LC\_ALL,"RU");

menu();

}

Листинг 3 – модуль **main.c**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

14

Курсовая работа

Следующий модуль **menu.c** содержит в себе следующие функции:

int CheckINT(void);//Проверка на число

int add(void);//Создание

int add\_l(void);//Ввод в конец

int add\_p(int);//Ввод в позицию

int del\_last(void);//Удаление

int del\_last\_buf(void);//Удаление из буфера

int in\_buf\_stack(void);//Ввод в буфер

int out\_buf\_stack(void);//Вывод из буфера

int print\_all(void);//Вывод всех

int print\_f(void);//Вывод первого

int print\_l(void);//вывод последнего

int print\_p(int);//Вывод по позиции

int del\_f(void);//Удаление последнего

int del\_p(int);//Удаление из позиции

int del\_full(void);//Полная очистка

int replace(int, int);//Перемещение

int sort\_fn(void);//Сортировка по ФИО

int sort\_date(void);//Сортировка по дате

int sort\_office(void);//Сортировка по отделу

int sort\_job(void);//Сортировка по должности

int sort\_experience(void);//Сортировка по стажу

int copy(HUMAN\*, HUMAN\*);//Копирование

int out\_file(void);//Вывод в файл с псевдографикой

Изм.

Лист

№ докум.

Подпись

Дата

Лист

15

Курсовая работа

int out\_file2(void);//Вывод в файл

int file\_pop(void);//Чтение из файла

int in\_buf\_buf(HUMAN\*, int);

int out\_buf\_buf(HUMAN\*);

Листинг 4 – функции модуля **menu.c**

Модуль **menu.c**:

*menu* – содержит структуру данных и состоит из функций, которые осуществляют работу программы и проверку на корректность ввода данных.

Модуль check.c содержит следующие функции:

void CheckFN(char \*, char \*, char \*);//Проверка

// на фамилию имя отчество

void CheckDATA(int \*, int \*, int \*);//Проверка

//на дату

void CheckWORD(char \*);//Проверка на слова

int CheckINT(void);//Проверка на число

Листинг 5 – функции проверки из модуля **check.c**

Модуль **check.c**:

*check* – содержит структуру данных и функции проверки.

Модуль **stack.c** содержит структуру данных и функции обеспечивающие работу программы:

int add(void);//Создание

int add\_l(void);//Ввод в конец

int add\_p(int);//Ввод в позицию

int del\_last(void);//Удаление

int del\_last\_buf(void);//Удаление из буфера

int in\_buf\_stack(void);//Ввод в буфер

int out\_buf\_stack(void);//Вывод из буфера

int print\_all(void);//Вывод всех

int print\_f(void);//Вывод первого

int print\_l(void);//вывод последнего

int print\_p(int);//Вывод по позиции

int del\_f(void);//Удаление последнего

int del\_p(int);//Удаление из позиции

int del\_full(void);//Полная очистка

int replace(int, int);//Перемещение

int sort\_fn(void);//Сортировка по ФИО

int sort\_date(void);//Сортировка по дате

int sort\_office(void);//Сортировка по отделу

int sort\_job(void);//Сортировка по должности

int sort\_experience(void);//Сортировка по стажу

int copy(HUMAN\*, HUMAN\*);//Копирование

int out\_file(void);//Вывод в файл с псевдографикой

Изм.

Лист

№ докум.

Подпись

Дата

Лист

16

Курсовая работа

int out\_file2(void);//Вывод в файл

int file\_pop(void);//Чтение из файла

int in\_buf\_buf(HUMAN\*, int);

int out\_buf\_buf(HUMAN\*);

Листинг 6 – функции модуля **stack.c**

**5 ТЕСТИРОВАНИЕ ПРОГРАММЫ**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

17

Курсовая работа

При запуске программы пользователь может наблюдать красивый цветной интерфейс и основное меню, содержащее пункты для работы с базой данных. Данное меню представлено на рисунке 5.1.

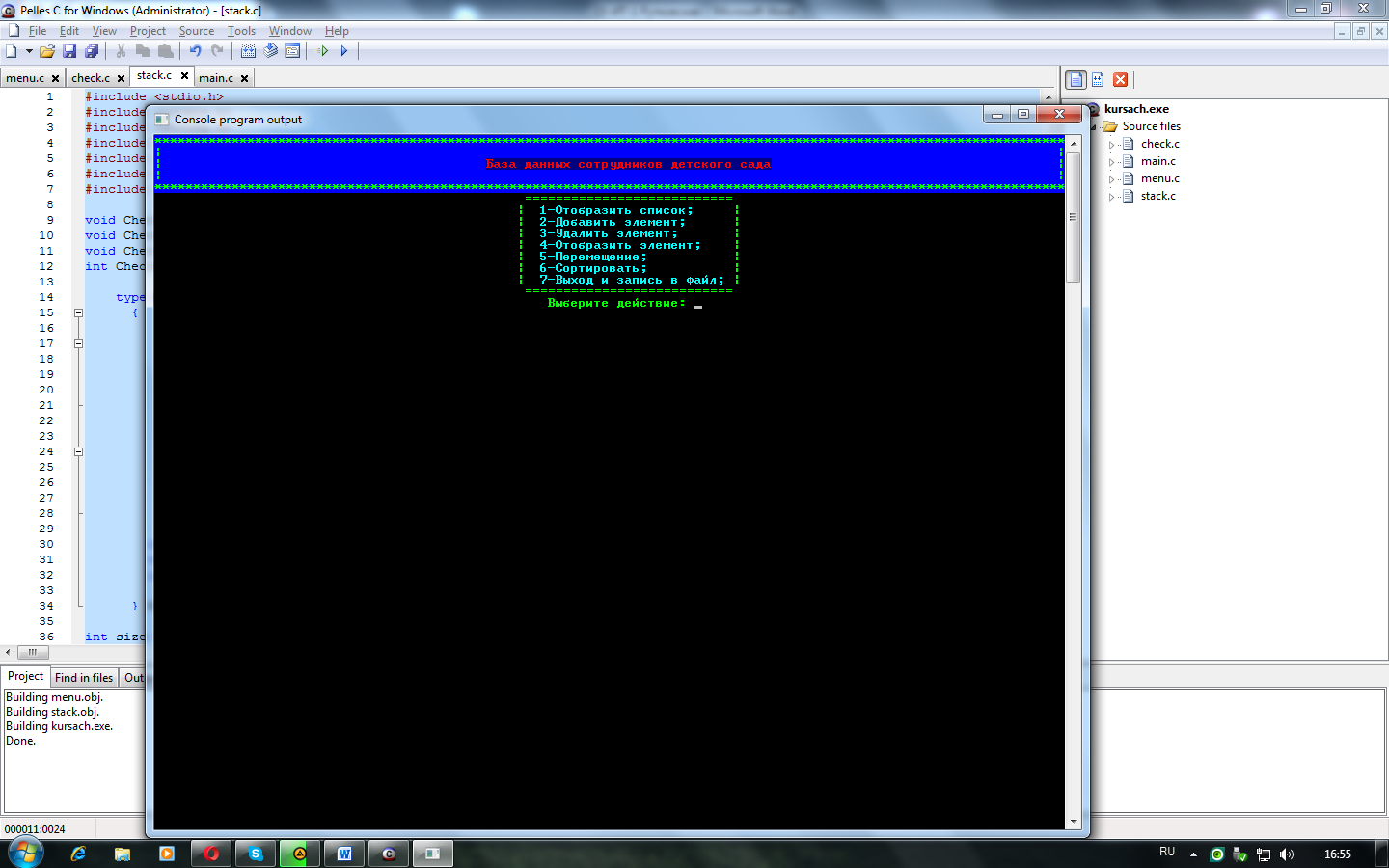


Рисунок 1 – меню

Изначально база данных содержит в себе информацию о сотрудниках дошкольного учреждения образования, чтобы увидеть список уже имеющихся сотрудников, от пользователя требуется нажать клавишу 1. В результате он увидит на экране отображение данных из текстового файла в виде псевдографической таблицы. Таблица представлена на рисунке 5.2.

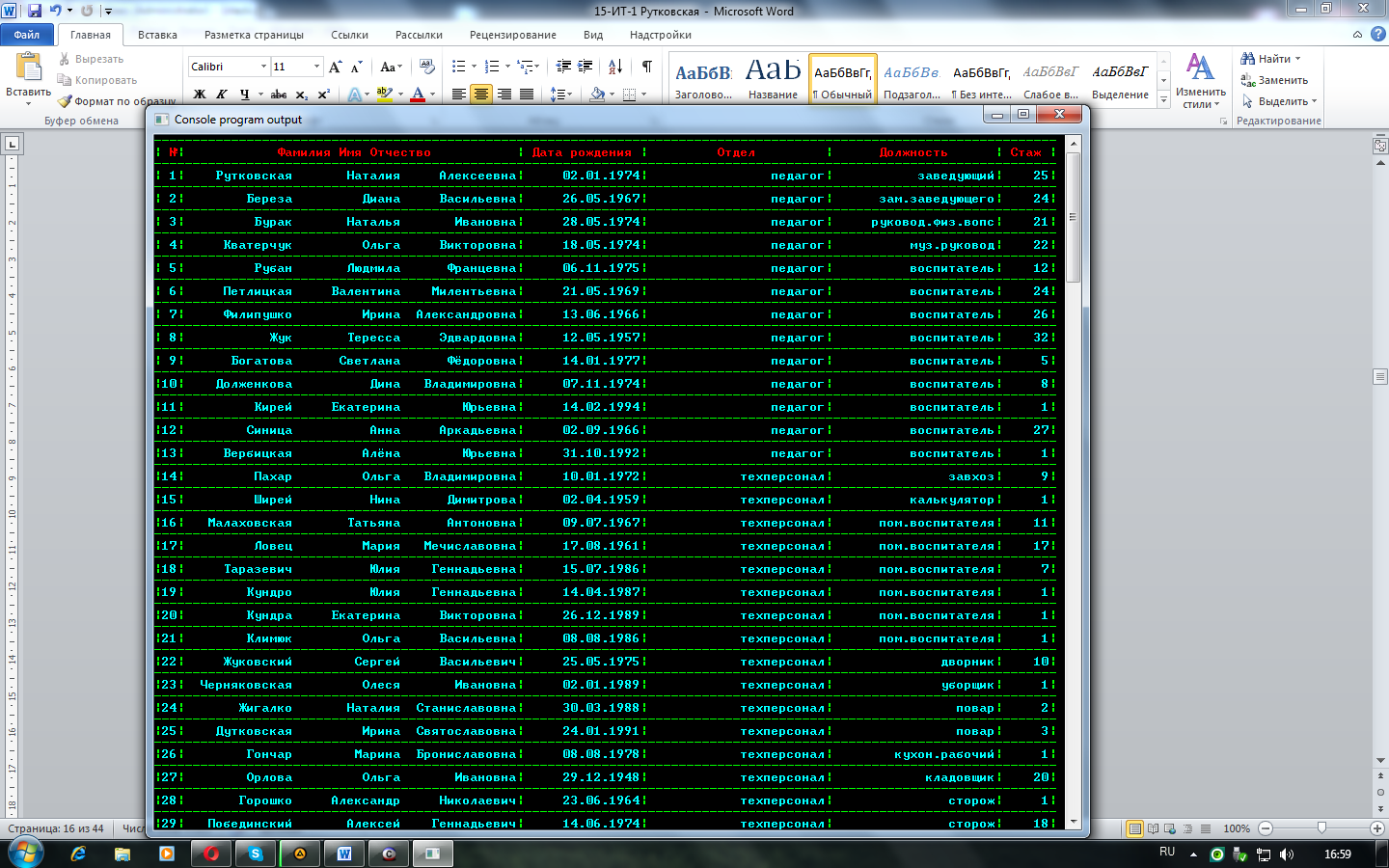


Рисунок 2 – вывод информации

Вторым пунктом в программе идет добавление элемента в список, т.е. сотрудника. При нажатии 2 пользователь видит дополнительное меню с выбором места добавления. Как оно выглядит представлено на рисунке 5.3.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

18

Курсовая работа

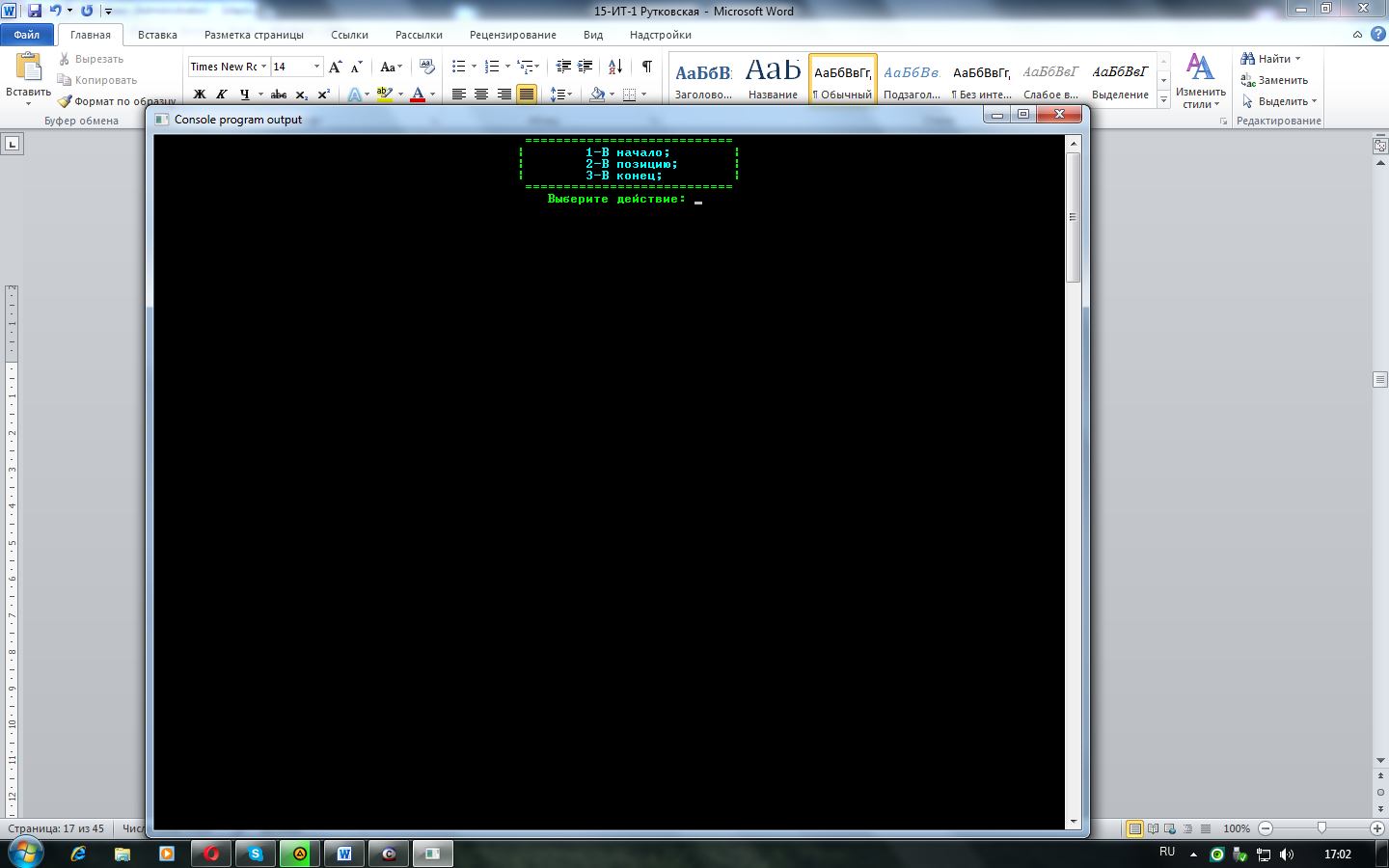


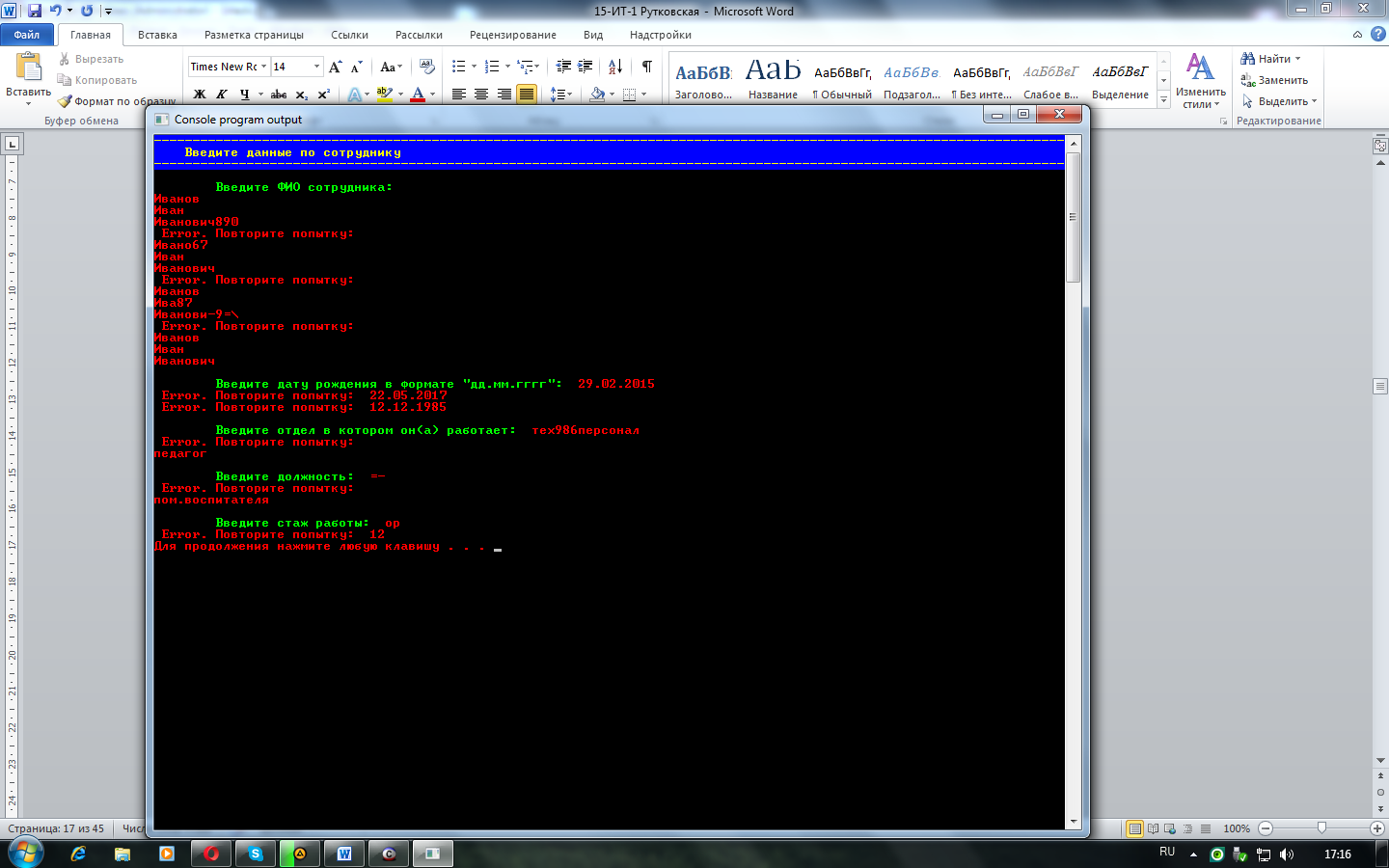
Рисунок 3 – дополнительное меню раздела «Добавить элемент»

При выборе добавления в начало, новый элемент будет добавлен в начало, соответственно также и в конец. Рассмотрим подробнее добавление в позицию. При выборе добавления в позицию пользователю открывается весь список, для удобства, и предоставляется возможность выбрать позицию для ввода. Чтобы выбрать позицию пользователь должен ввести число. Данное действие представлено на рисунке 5.4. Для примера выберем позицию 6.



Рисунок 4 – выбор добавление в позицию 6

Далее пользователь производит ввод данных по сотруднику. Рассмотрим заодно и проверки на корректность ввода, которые представлены на рисунке 5.5.



Изм.

Лист

№ докум.

Подпись

Дата

Лист

19

Курсовая работа

Рисунок 5 – проверки на корректность ввода

Мы вводили нового сотруднику в позицию 6. При отображении списка мы видим, что он в 6-ой позиции на рисунке 5.6.

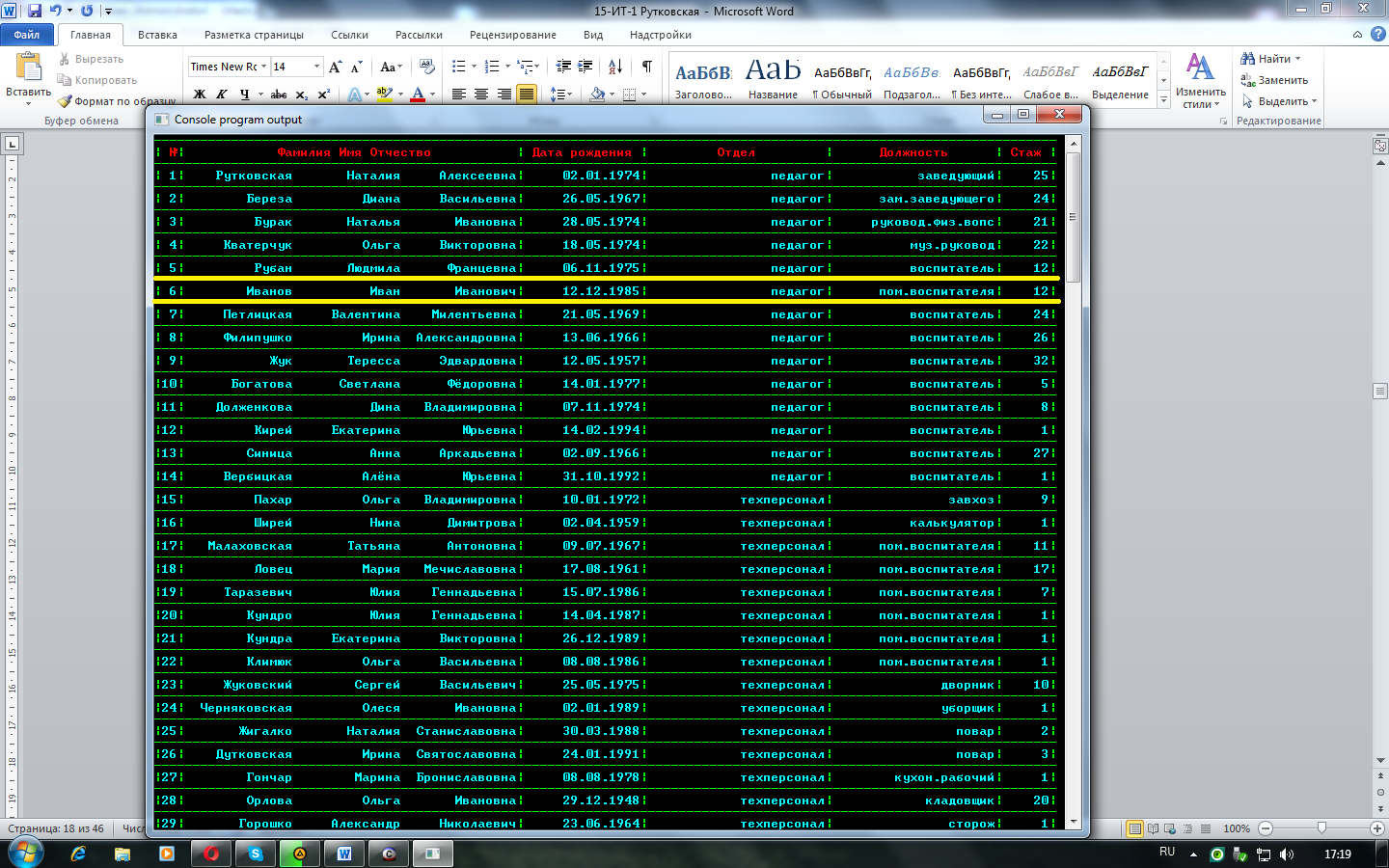


Рисунок 6 – сотрудник «Иванов Иван Иванович» под номером 6

При выборе третьего пункта меню, пользователю, в начале, предлагается выбрать место удаления (как показано на рисунке 5.7). При выборе удаления с начала списка – удаляется первый элемент (сотрудник), при выборе удаления с конца списка – удаляется последний элемент (сотрудник) соответственно. Если же пользователь выбирает удаление в позиции, то здесь уже работа осуществляется, как и с добавлением в позицию. Пользователь может ввести число (номер сотрудника) и элемент под этим номером удаляется. Внешнее отличие добавления от удаления заключается в том, что, если в добавлении список открывался только при добавлении в позицию, то в удалении он имеется всегда, для удобства пользователя.



Рисунок 7 – дополнительное меню раздела «Удалить элемент»

Также в удалении есть ещё функция очистки всего списка, её мы и рассмотрим на рисунке 5.8.

**ПРИМЕЧАНИЕ:** Чтобы не потерять всю уже имеющуюся базу для рассмотрения дальнейших пунктов, я произвела полное удаление списка, а после выйду из программы без предварительного сохранения в файл.

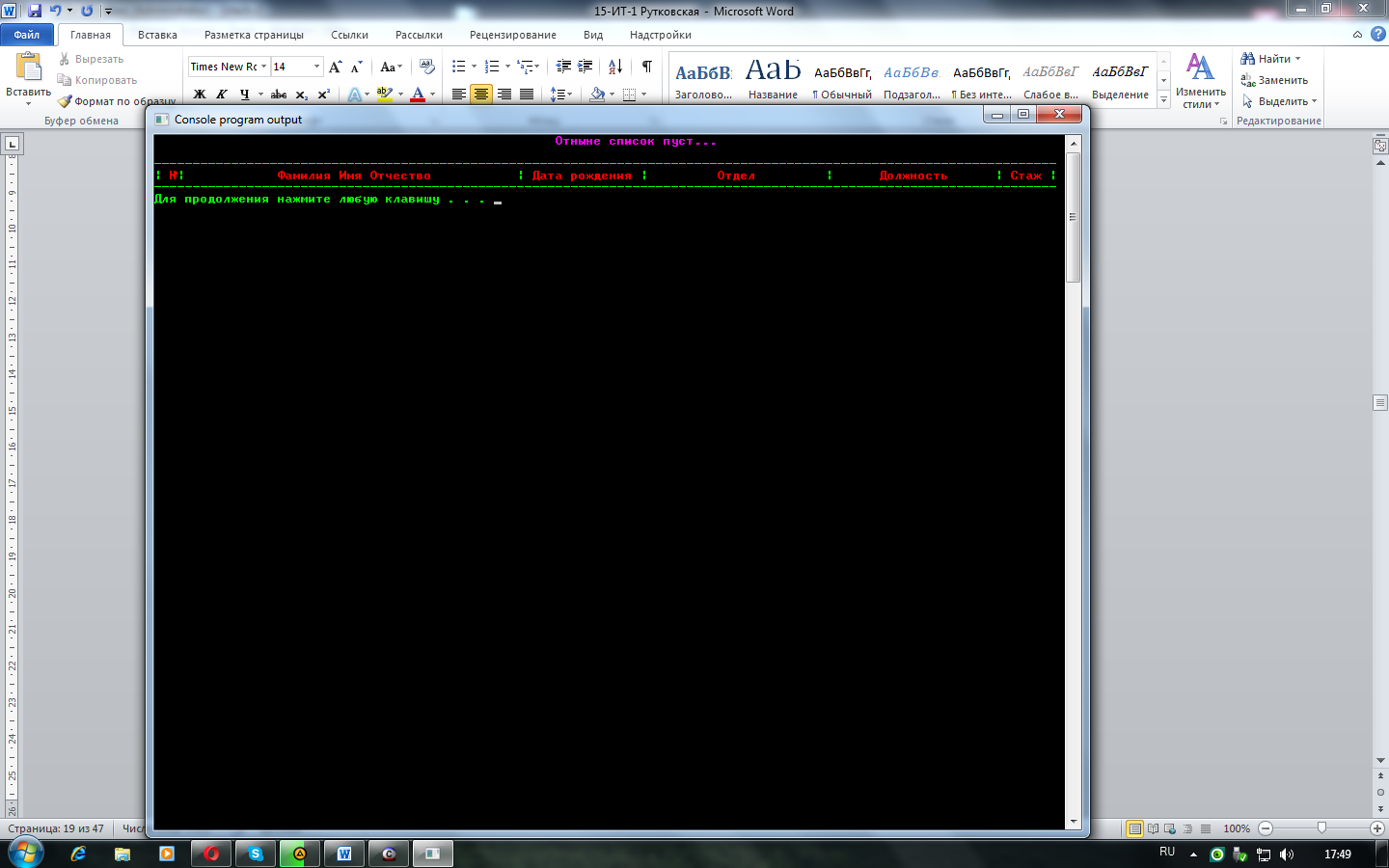


Рисунок 8 – полная очистка списка и сообщение о пустоте списка

Изм.

Лист

№ докум.

Подпись

Дата

Лист

20

Курсовая работа

Выбирая пункт 4, пользователь снова открывает дополнительное меню, но уже раздела «Отображение элемента» (представлено на рисунке 5.9). В данном разделе пользователь может выбрать, какой элемент он хочет отобразить, т.е. по какому сотруднику просмотреть данные. Пользователю на выбор предоставляется отображение первого, отображение последнего и отображение по позиции. Представлено на рисунках 5.10, 5.11, 5.12 соответственно.

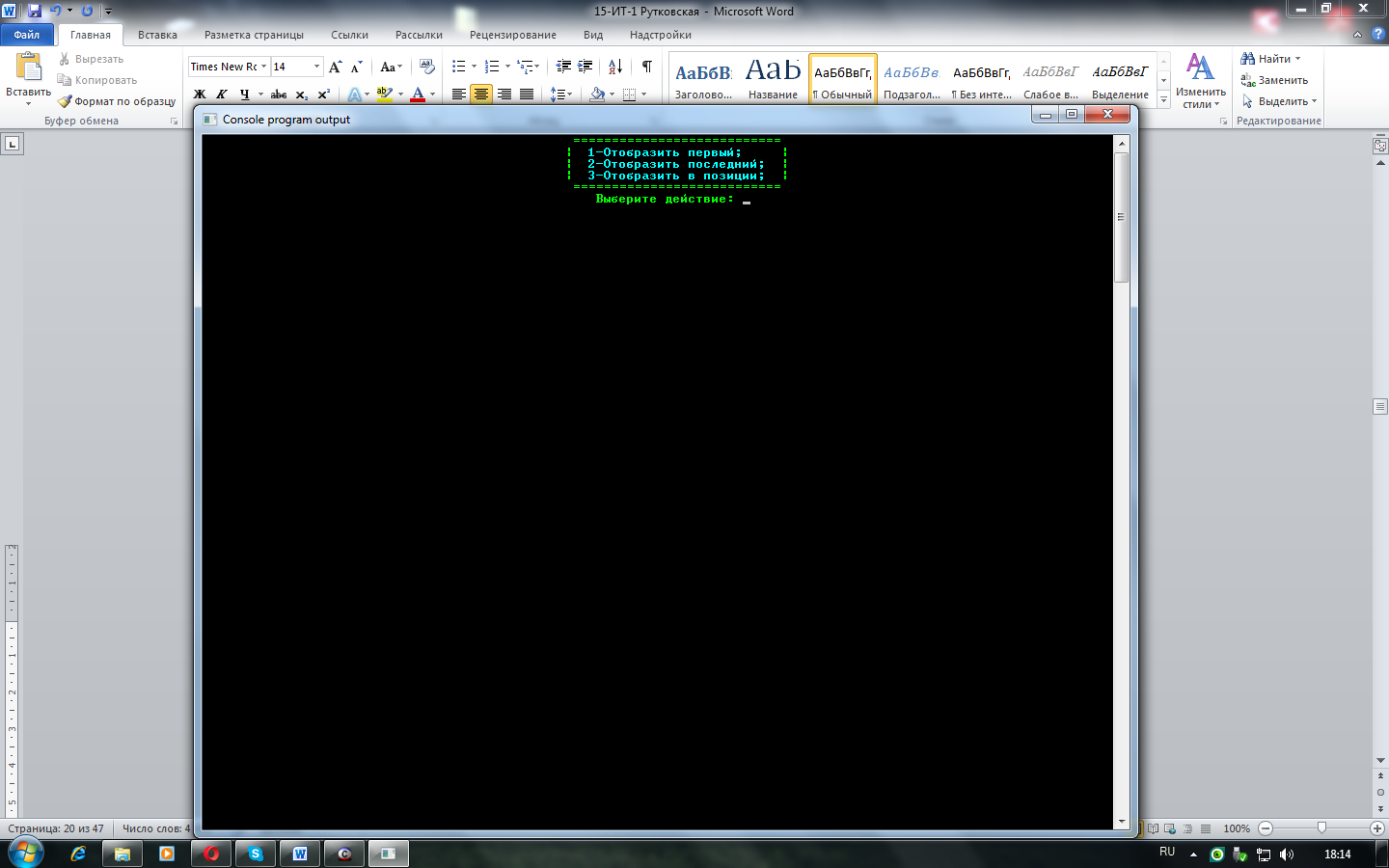


Рисунок 9 – дополнительное меню раздела «Отображение элемента»

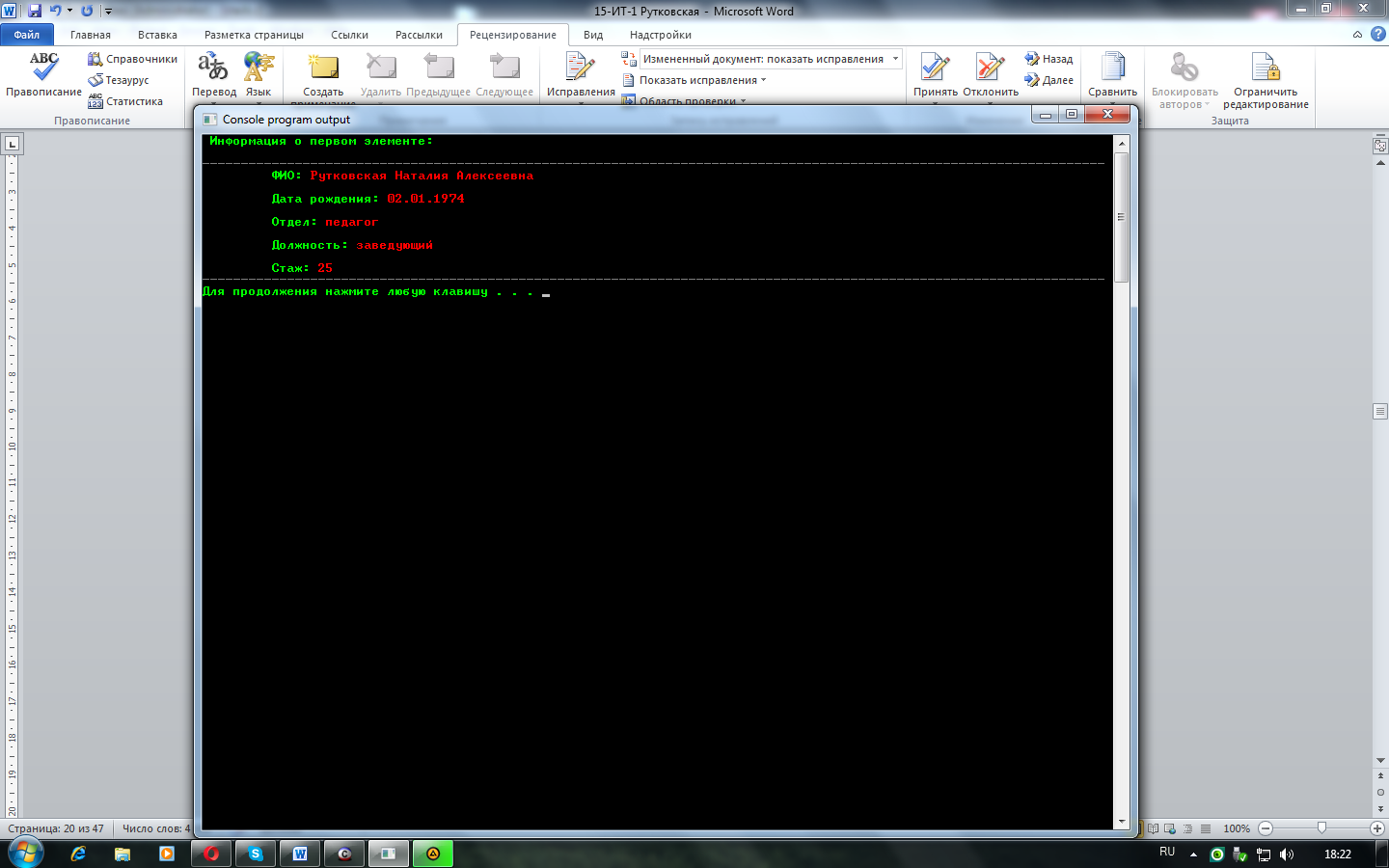


Рисунок 10 – отображение первого элемента списка

Изм.

Лист

№ докум.

Подпись

Дата

Лист

21

Курсовая работа

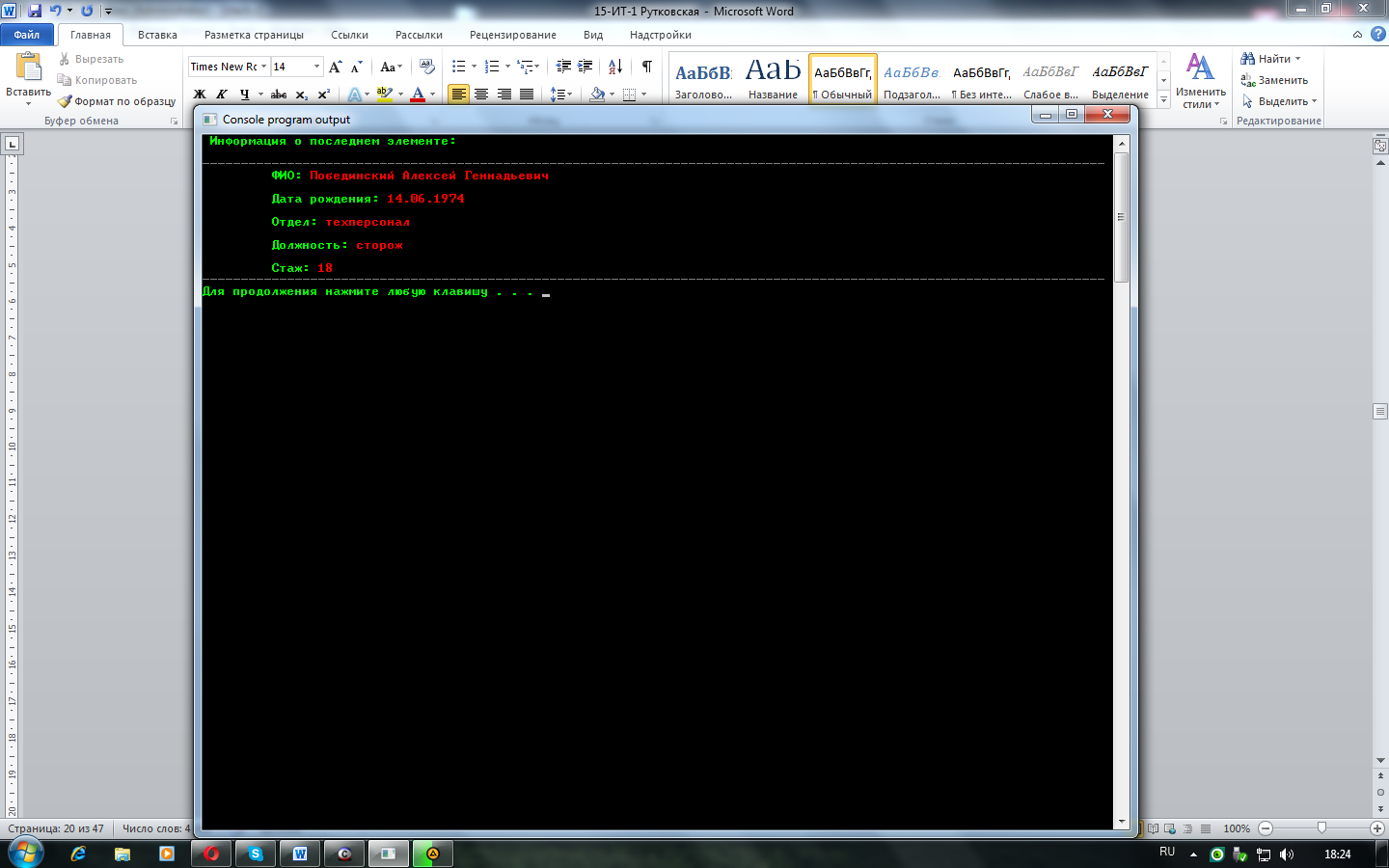


Рисунок 11 – отображение последнего элемента списка

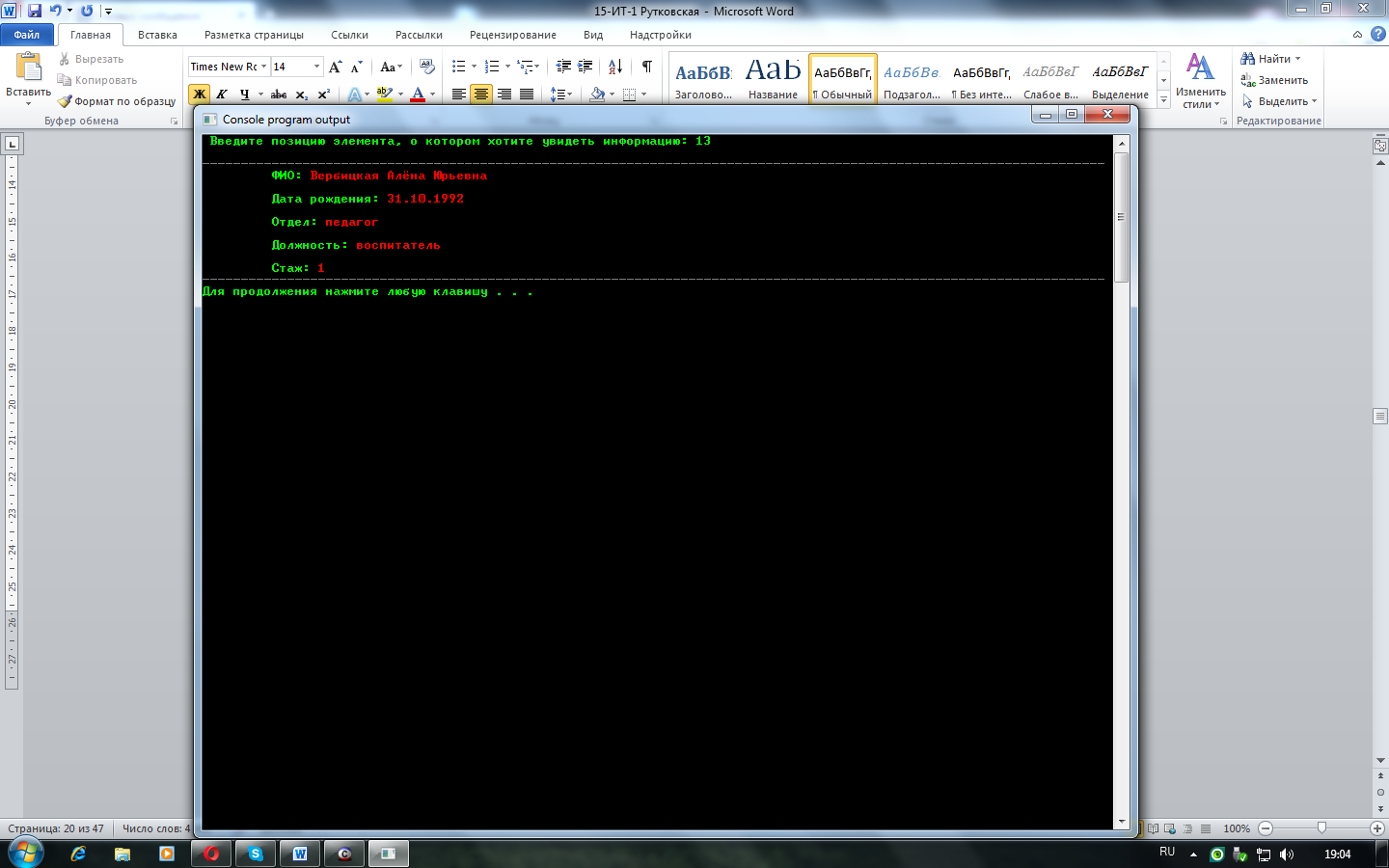


Рисунок 12 – выбор элемента для отображения по позиции

При выборе пункта 5, пользователь может произвести перемещение элемента в списке. Для этого программа запрашивает у пользователя исходную позицию элемента (представлено на рисунке 5.13), который мы хотим переместить, для удобства в программе выводится список в консоль. Далее программа запрашивает новую позицию для элемента (сотрудника) (что представлено на рисунке 5.14), в которую он хочет поместить элемент. Все остальные элементы сдвигаются. Итог представлен на рисунке 5.15.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

22

Курсовая работа

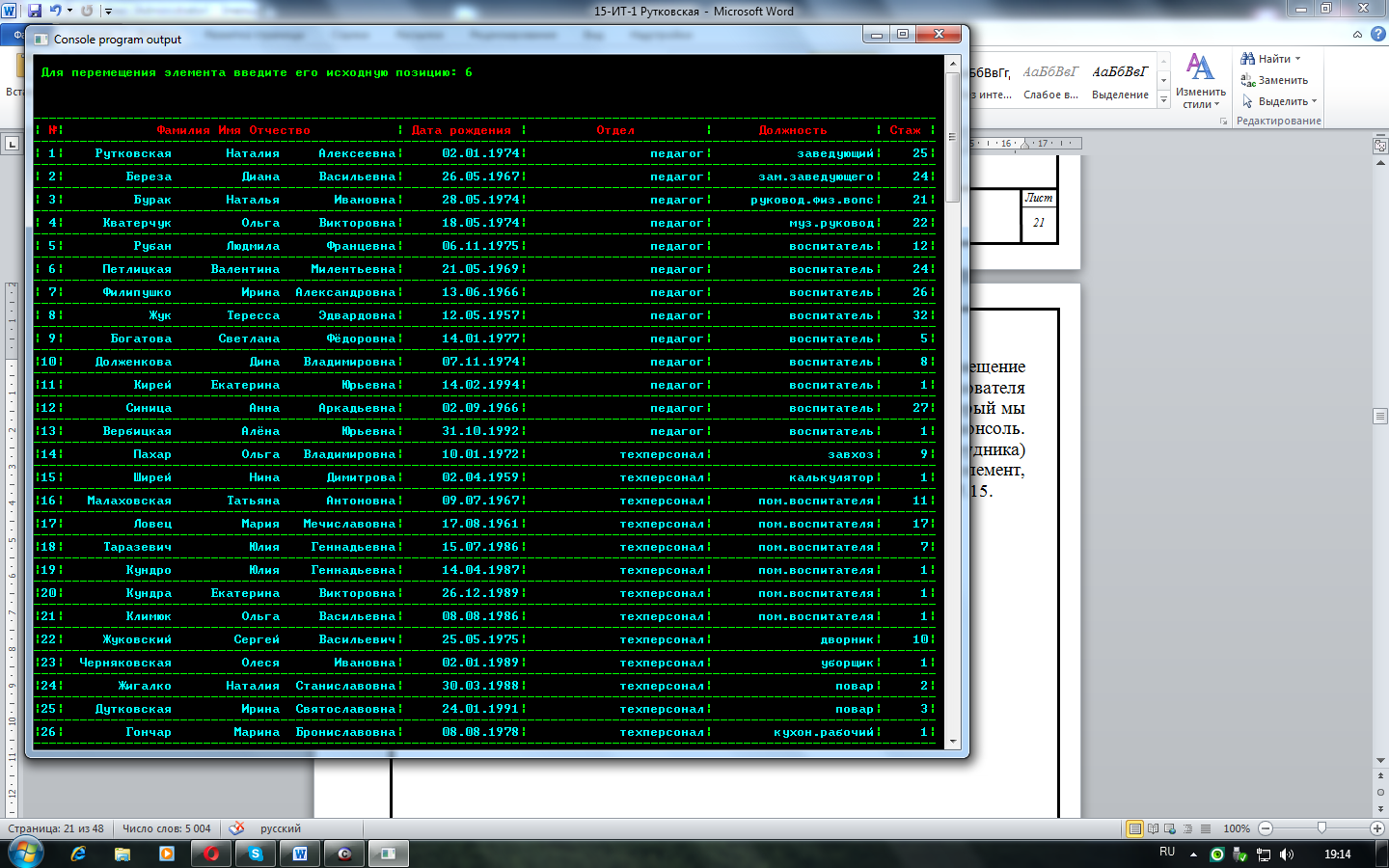


Рисунок 13 – ввод исходной позиции элемента

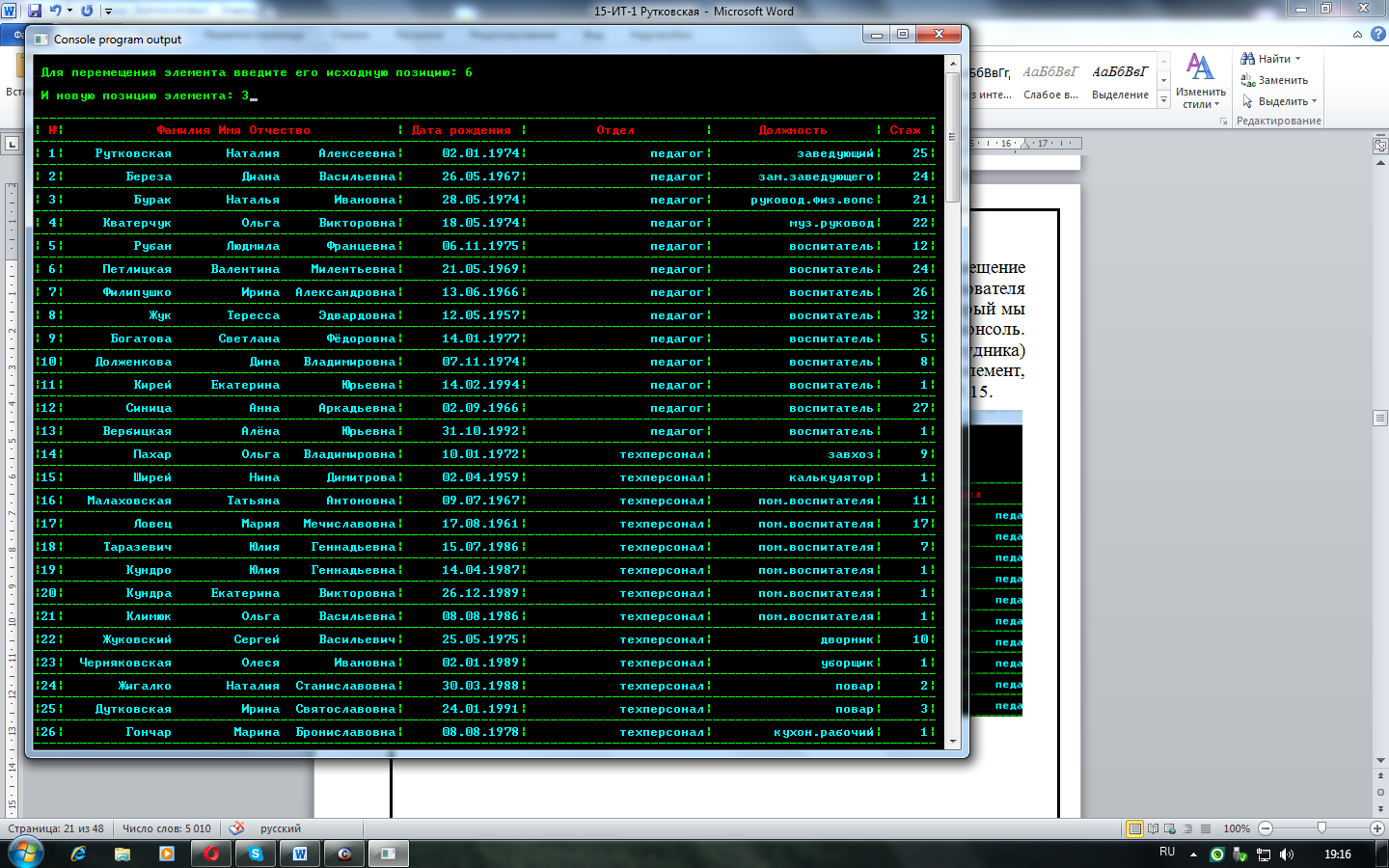


Рисунок 14 – ввод новой позиции элемента

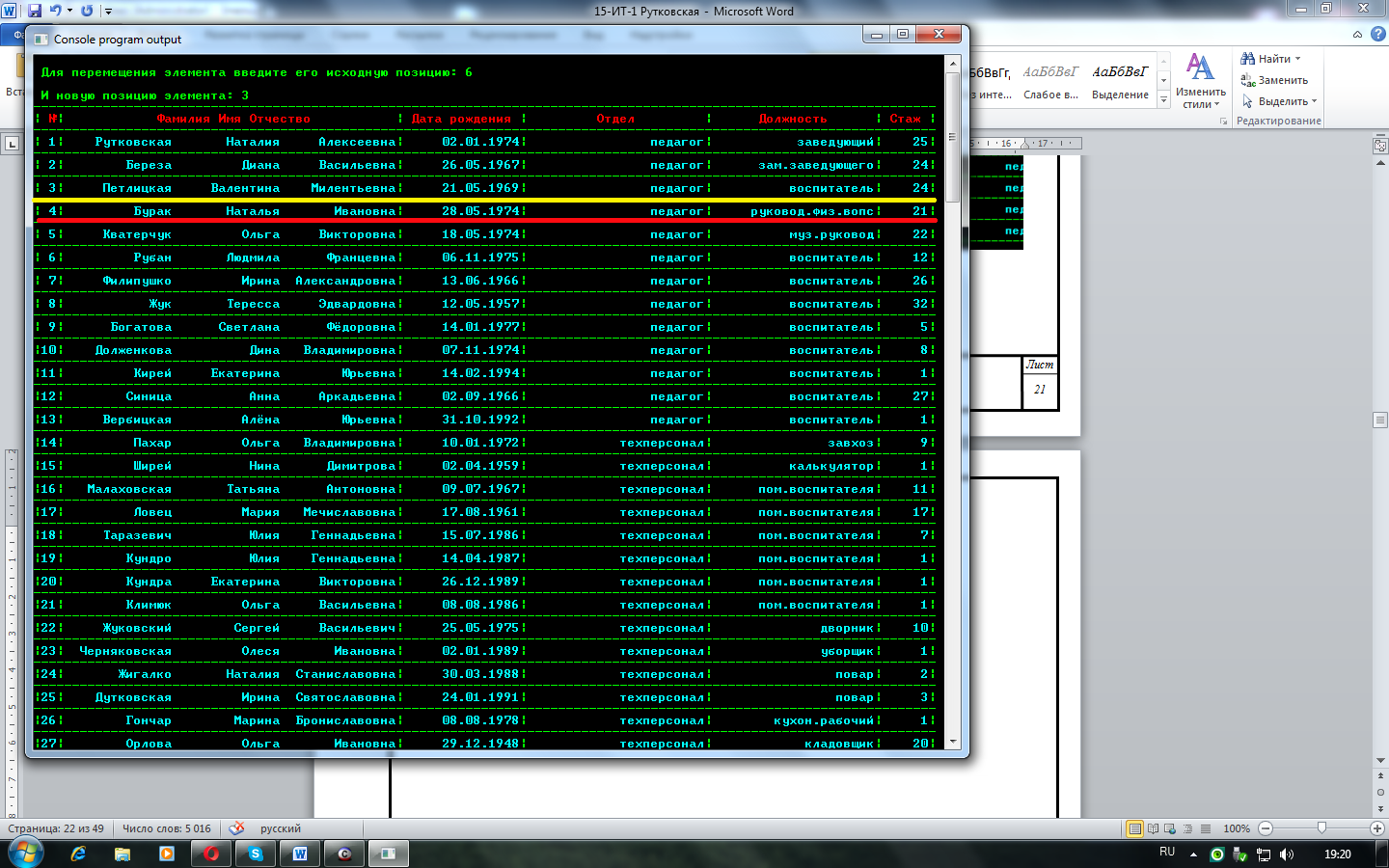


Рисунок 15 – итог перемещения: жёлтым подчёркнут элемент, который мы перенесли из 6 позиции в 3, красным показан элемент который раньше был в позиции 3.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

23

Курсовая работа

Выбирая 6-ой пункт меню, пользователь открывает дополнительное меню раздела сортировки (представлено на рисунке 5.16)и весь список, а далее пользователь может отсортировать данные по любому из 5 полей: Ф.И.О., Дата рождения, Отдел, Должность, Стаж.

После того как сортировка прошла, появляется сообщение о том по какому полю произошла сортировка выводится уже отсортированный список. Итоги сортировки представлены на рисунках 5.17, 5.18, 5.19, 5.20, 5.21 соответственно.

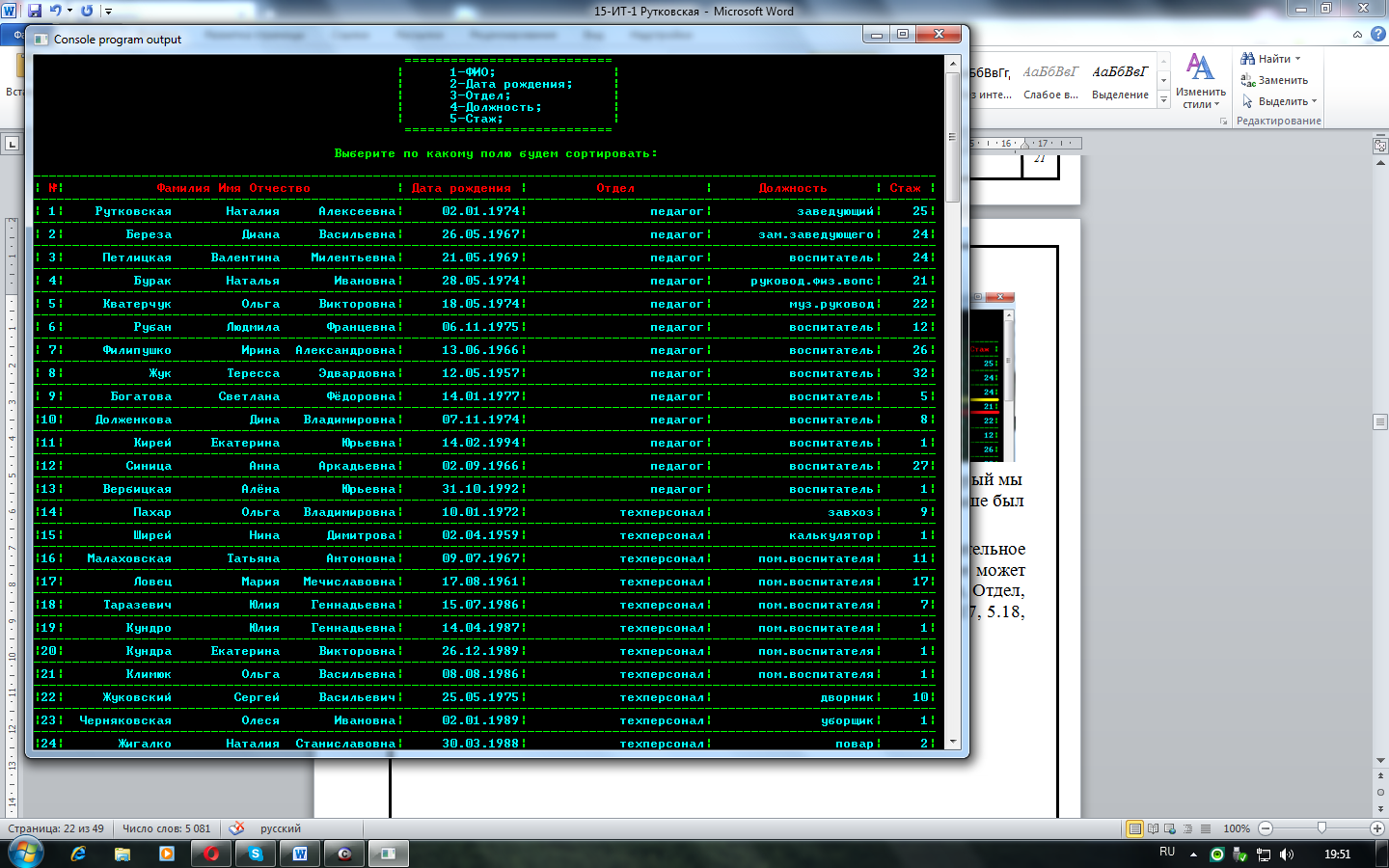
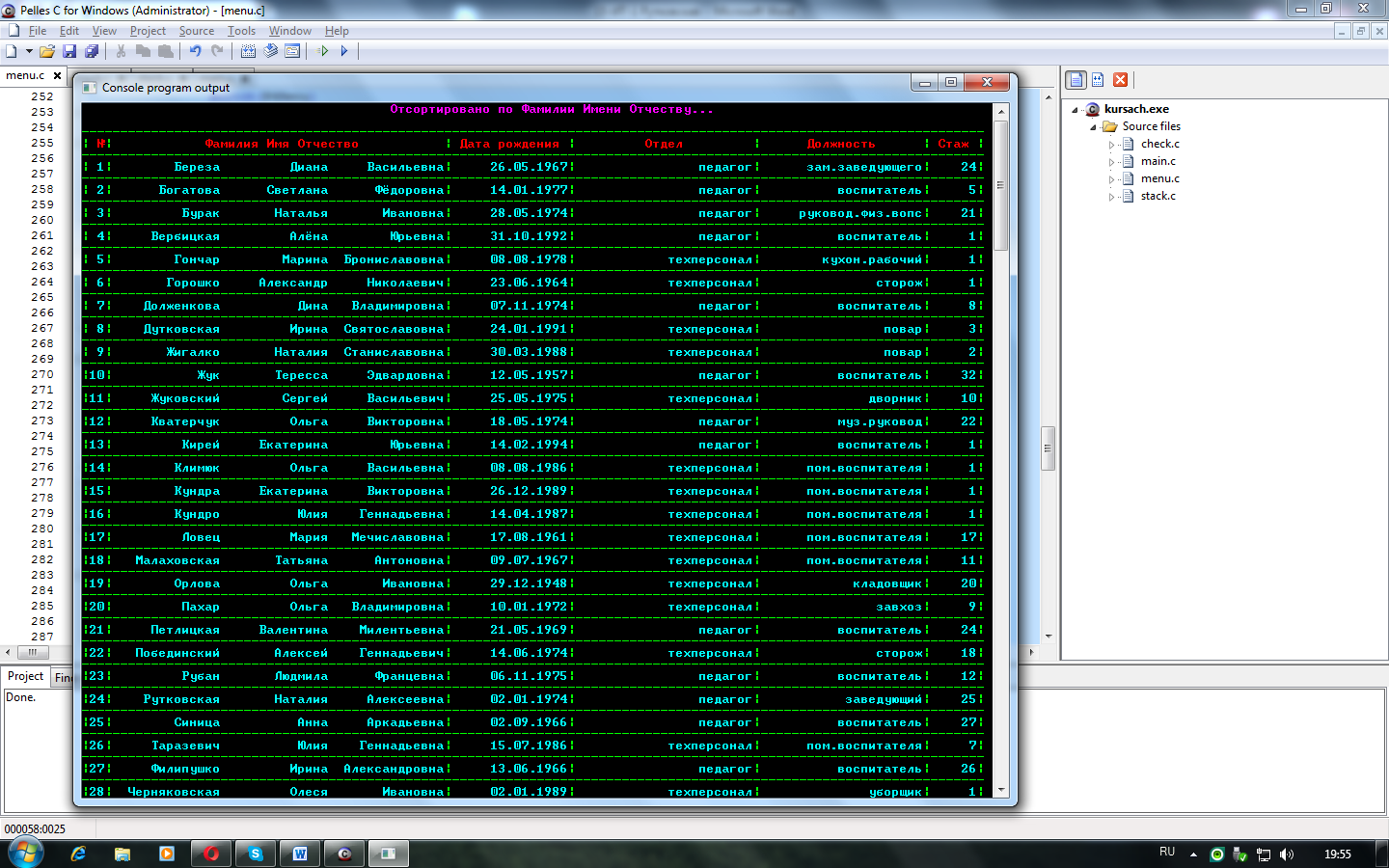


Рисунок 16 – дополнительное меню раздела «Сортировка»



Изм.

Лист

№ докум.

Подпись

Дата

Лист

24

Курсовая работа

Рисунок 17 – итог сортировки по Ф.И.О.

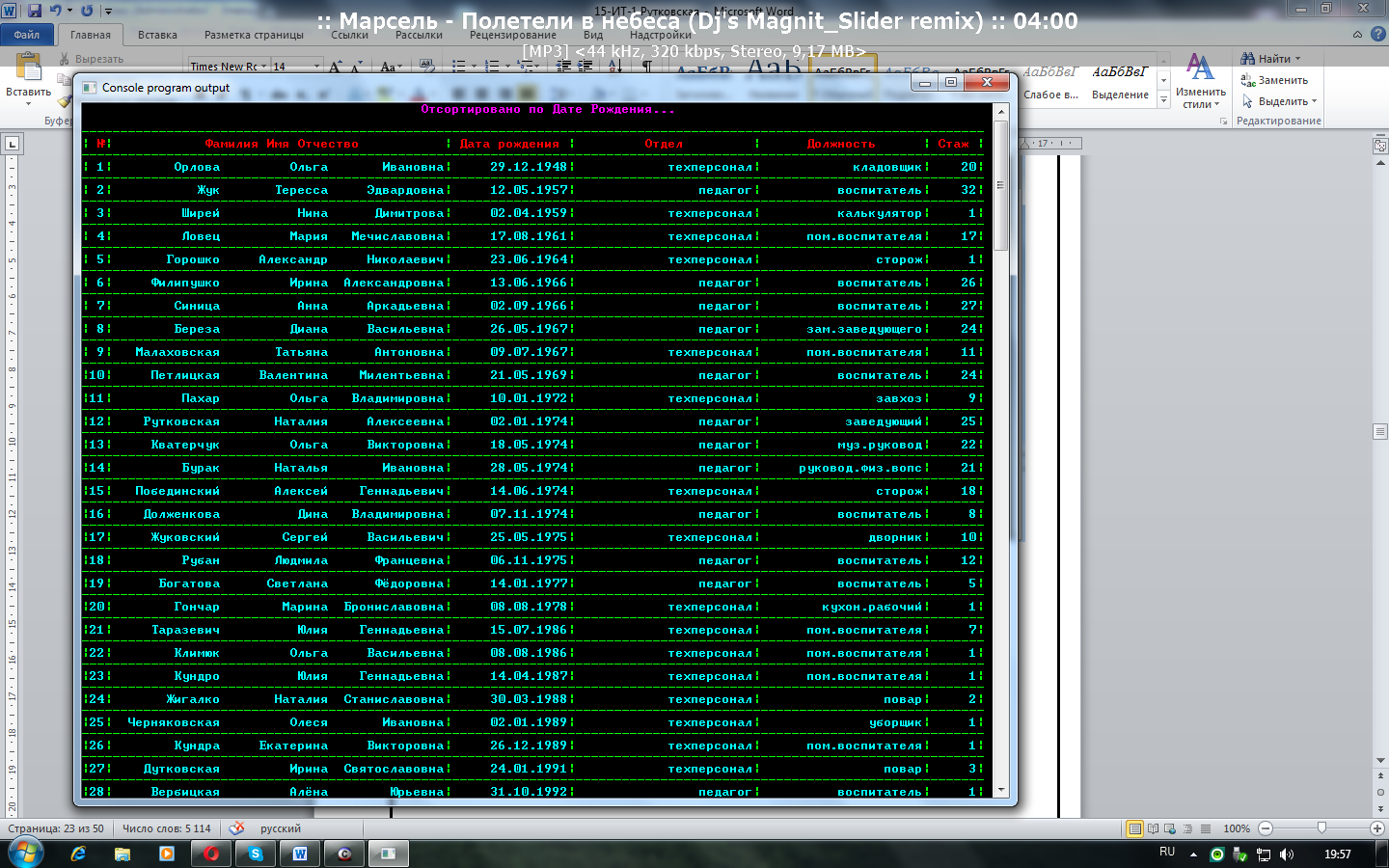


Рисунок 18 – итог сортировки по дате рождения



Рисунок 19 – итог сортировки по отделу

Изм.

Лист

№ докум.

Подпись

Дата

Лист

25

Курсовая работа

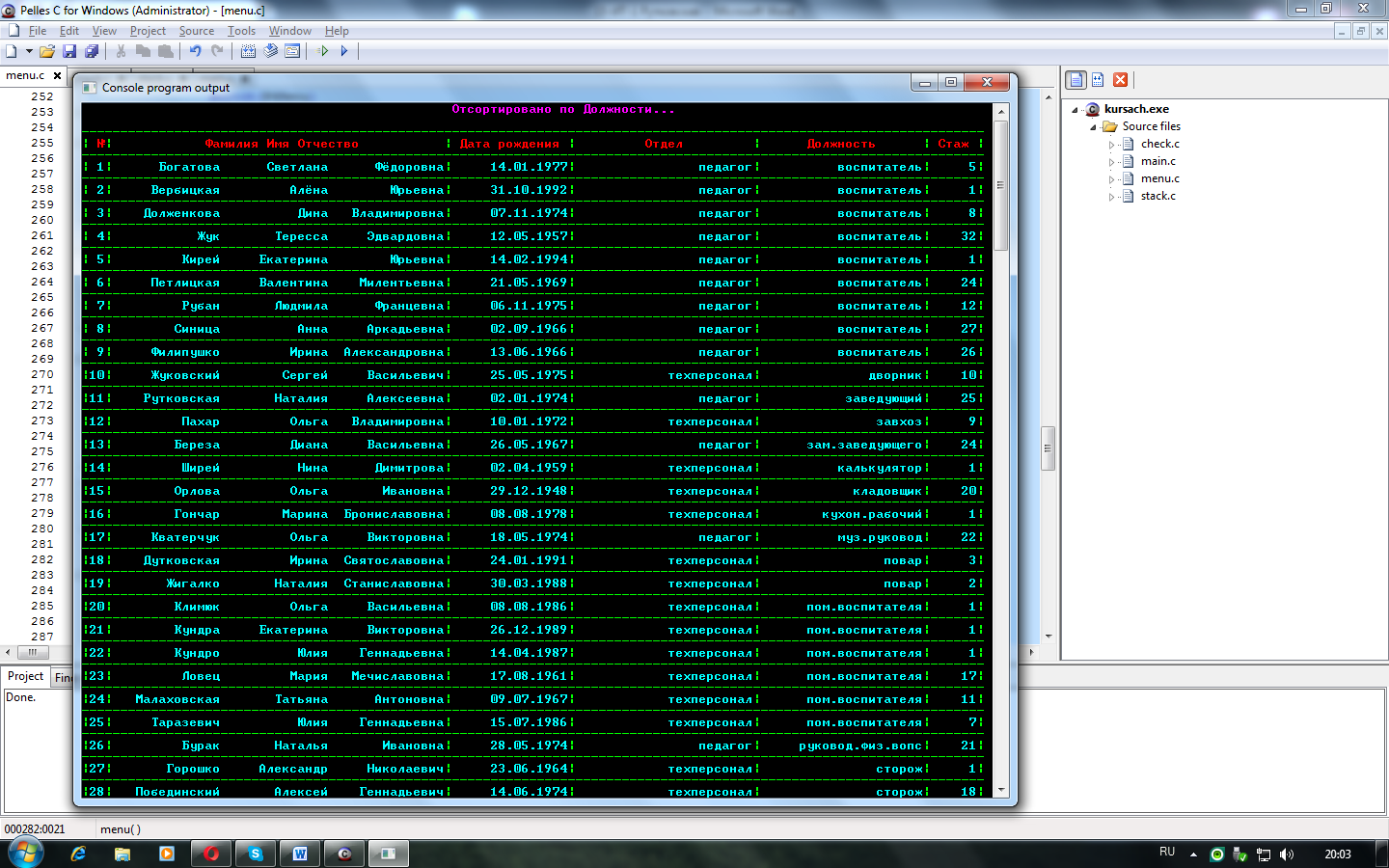


Рисунок 20 – итог сортировка по должности



Рисунок 21 – итог сортировки по стажу

И в конце, если пользователь нажимает 7, ему открывается последнее дополнительное меню, представленное на рисунке 5.22, содержащее пункты: сохранение в файл и выход.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

26

Курсовая работа

При выборе 1, пользователь сохраняет в файл все данные в том виде, в котором они есть на данный момент после проделанных операций.

При выборе 2, пользователь осуществляет выход из консоли.

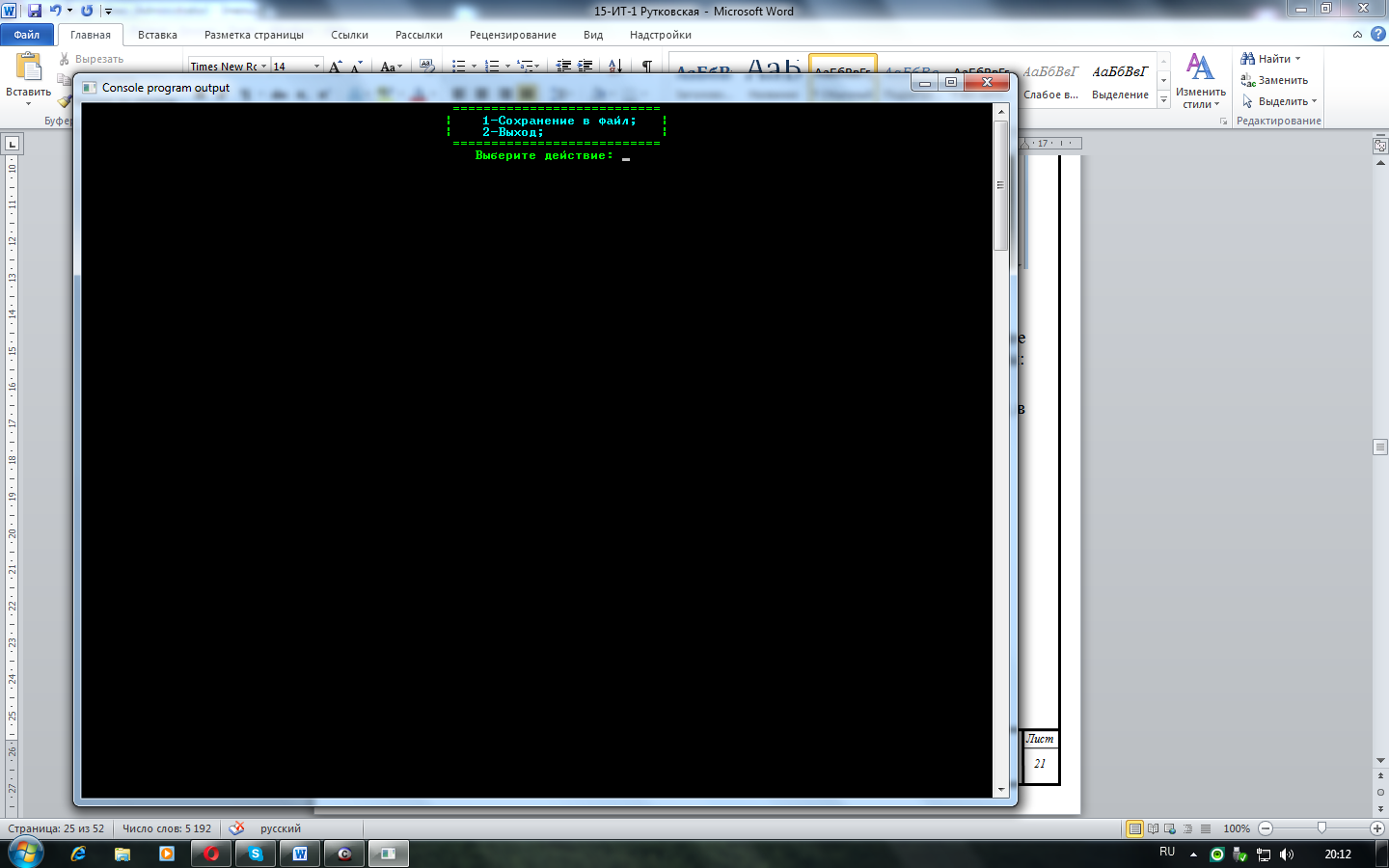


Рисунок 22 – дополнительное меню раздела «Сохранение в файл и выход»

# 6 ЗАКЛЮЧЕНИЕ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

27

Курсовая работа

В данной курсовой работе была поставлена цель, использовав полученные знания языка программирования Cи, реализовать работоспособное ПО, которое должно хранить в себе информацию о сотрудниках дошкольного учреждения образования, а также корректно обрабатывать данную информацию и вносить новые записи.

Удалось реализовать программное обеспечение, которое способно управлять базой данных. Оно добавляет, удаляет, сортирует данные, а также осуществляет перемещение и отображение элементов, реагирует на ошибки и сохраняет данные в текстовый файл в виде псевдографической таблицы.

Программа имеет дружественный интерфейс и проводит проверку на правильность ввода данных. Программа так же осуществляет:

* Запись и загрузку файла базы данных (текстовый файл).
* Добавление новых записей, удаление старых.
* Сортировать записи по любому из полей базы данных.
* Перемещать запись из старой позиции в новую.
* Отображать весь список записей в консоль.
* Отображать информацию по одной записи.

Имеет следующие поля:

1. Фамилия, имя, отчество сотрудника (например: Иванов Иван Иванович).
2. Дата рождения (пишется в формате дд.мм.гггг).
3. Отдел (имеется в виду отдел, к которому относится сотрудник. В дошкольном учреждении образования есть два отдела: педагог и технический персонал).
4. Должность (вводится должность сотрудника).
5. Стаж работы (количество полных лет, которые отработал человек в данной сфере за всю жизнь до сего момента).

**СПИСОК ЛИТЕРАТУРЫ**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

28

Курсовая работа

1. С.П. Чеботарёв УМК «Конструирование программ и языки программирования». Новополоцк: ПГУ, 2007 часть 1.
2. Конспект лекций по дисциплине «Основы алгоритмизации и программирование».
3. Конспект лекций по дисциплине «Языки программирования».

**ПРИЛОЖЕНИЕ А**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

30

Курсовая работа

main

file\_pop

add

add\_l

out\_file

add\_p

del\_f

sort\_fn

del\_last

menu

sort\_date

del\_p

sort\_office

del\_full

sort\_job

print\_f

sort\_experience

print\_l

replace

print\_p

print\_all

Рисунок 23 –функциональная схема программы

**ПРИЛОЖЕНИЕ Б**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

31

Курсовая работа

Листинг **main.c**:

#include <stdio.h>

#include <locale.h>

#include <string.h>

#include <stdlib.h>

#include <conio.h>

#include <time.h>

#include <ctype.h>

int menu();

typedef struct HUMAN

{

struct FullName{

char surname[15];

char name[15];

char patronymic[15];

} FN;

struct Data{

int day;

int month;

int year;

} DATA;

char office[50];

char job[50];

int experience;

struct HUMAN \*next;

} HUMAN;

int main(int argc, char \*argv[])

{

setlocale(LC\_ALL,"RU");

menu();

}

Листинг 7 – модуль main.c

Листинг модуля **menu.c** :

Изм.

Лист

№ докум.

Подпись

Дата

Лист

32

Курсовая работа

#include <stdio.h>

#include <locale.h>

#include <string.h>

#include <stdlib.h>

#include <conio.h>

#include <time.h>

#include <ctype.h>

int CheckINT(void);

typedef struct HUMAN

{

struct FullName

{

char surname[15];

char name[15];

char patronymic[15];

} FN;

struct Data

{

int day;

int month;

int year;

} DATA;

char office[50];

char job[50];

int experience;

struct HUMAN \*next;

} HUMAN;

int CheckINT(void);

int add(void);//Создание

int add\_l(void);//Ввод в конец

int add\_p(int);//Ввод в позицию

int del\_last(void);//Удаление

int del\_last\_buf(void);//Удаление из буфера

int in\_buf\_stack(void);//Ввод в буфер

int out\_buf\_stack(void);//Вывод из буфера

int print\_all(void);//Вывод всех

int print\_f(void);//Вывод первого

int print\_l(void);//вывод последнего

int print\_p(int);//Вывод по позиции

int del\_f(void);//Удаление последнего

int del\_p(int);//Удаление из позиции

Изм.

Лист

№ докум.

Подпись

Дата

Лист

33

Курсовая работа

int del\_full(void);//Полная очистка

int replace(int, int);//Перемещение

int sort\_fn(void);//Сортировка по ФИО

int sort\_date(void);//Сортировка по дате

int sort\_office(void);//Сортировка по отделу

int sort\_job(void);//Сортировка по должности

int sort\_experience(void);//Сортировка по стажу

int copy(HUMAN\*, HUMAN\*);//Копирование

int out\_file(void);//Вывод в файл с псевдографикой

int out\_file2(void);//Вывод в файл

int file\_pop(void);//Чтение из файла

int in\_buf\_buf(HUMAN\*, int);

int out\_buf\_buf(HUMAN\*);

int menu(int argc, char \*\*argv)

{

file\_pop();

del\_last();

while(1)

{

system("cls");

\_textcolor(10);

\_textbackground(9);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("|");

printf(" ");

printf("|");

printf("|\t\t\t\t\t ");

\_textcolor(44);

\_textbackground(1);

printf("База данных сотрудников детского сада");

\_textcolor(10);

\_textbackground(9);

printf("\t\t\t\t |");

printf("| |");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

\_textbackground(0);

\_textcolor(10);

printf("\t\t\t\t\t\t===========================");

\_textbackground(16);

\_textcolor(10); printf("\n | "); \_textcolor(11); printf("1-Отобразить список; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("2-Добавить элемент; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("3-Удалить элемент; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("4-Отобразить элемент; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("5-Перемещение; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("6-Сортировать; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("7-Выход и запись в файл; ");

\_textcolor(10);

printf("|\n\t\t\t\t\t\t===========================");

printf("\n Выберите действие: ");

int NMenu;

NMenu=CheckINT();

switch(NMenu)

{

//Отображение списка

case 1:

{

system("cls");

print\_all();

system("PAUSE");

break;

Изм.

Лист

№ докум.

Подпись

Дата

Лист

34

Курсовая работа

}

//Добавление

case 2:

{

system("cls");

int N2Menu;

\_textcolor(10);

printf("\t\t\t\t\t\t===========================");

\_textbackground(16);

\_textcolor(10); printf("\n | "); \_textcolor(11); printf("1-В начало; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("2-В позицию; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("3-В конец; ");

\_textcolor(10);

printf("|\n\t\t\t\t\t\t===========================");

printf("\n Выберите действие: ");

N2Menu=CheckINT();

switch(N2Menu)

{

case 1:

{

system("cls");

add();

break;

}

case 2:

{

system("cls");

int pos;

\_gotoxy(1,4);

\_textcolor(10);

print\_all();

\_gotoxy(8,2);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

35

Курсовая работа

printf("Введите позицию для добавления нового элемента: ");

pos=CheckINT();

system("cls");

add\_p(pos);

break;

}

case 3:

{

system("cls");

add\_l();

break;

}

}

system("PAUSE");

break;

}

//Удаление

case 3:

{

system("cls");

int N3Menu;

\_textcolor(10);

printf("\t\t\t\t\t\t===========================");

\_textbackground(16);

\_textcolor(10); printf("\n | "); \_textcolor(11); printf("1-Удалить с начала; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("2-Удалить с конца; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("3-Удалить с позиции; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("4-Очистить список; ");

\_textcolor(10);

printf("|\n\t\t\t\t\t\t===========================");

\_gotoxy(1,10);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

36

Курсовая работа

\_textcolor(10);

print\_all();

\_gotoxy(8,7);

printf("\n Выберите действие: ");

N3Menu=CheckINT();

switch(N3Menu)

{

case 1:

{

system("cls");

\_textcolor(45);

printf("\t\t\t\t\t\t Удалён первый...\n\n");

del\_last();

\_textcolor(10);

break;

}

case 2:

{

system("cls");

\_textcolor(45);

printf("\t\t\t\t\t\t Удалён последний...\n\n");

del\_f();

\_textcolor(10);

break;

}

case 3:

{

system("cls");

int pos;

printf("\n\t\t\tВведите позицию элемента для удаления: ");

\_gotoxy(1,4);

\_textcolor(10);

print\_all();

\_gotoxy(64,2);

pos=CheckINT();

system("cls");

\_textcolor(45);

printf("\t\t\t\t\t\t Удалено в позиции...\n\n");

Изм.

Лист

№ докум.

Подпись

Дата

Лист

37

Курсовая работа

del\_p(pos);

\_textcolor(10);

break;

}

case 4:

{

system("cls");

\_textcolor(45);

printf("\t\t\t\t\t\t Отныне список пуст...\n\n");

del\_full();

\_textcolor(10);

break;

}

}

\_textcolor(10);

print\_all();

system("PAUSE");

break;

}

//Отобразить элемент

case 4:

{

system("cls");

int N4Menu;

\_textcolor(10);

printf("\t\t\t\t\t\t===========================");

\_textbackground(16);

\_textcolor(10); printf("\n | "); \_textcolor(11); printf("1-Отобразить первый; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("2-Отобразить последний; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("3-Отобразить в позиции; ");

\_textcolor(10);

printf("|\n\t\t\t\t\t\t===========================");

printf("\n Выберите действие: ");

Изм.

Лист

№ докум.

Подпись

Дата

Лист

38

Курсовая работа

N4Menu=CheckINT();

switch(N4Menu)

{

case 1:

{

system("cls");

printf(" Информация о первом элементе:\n");

print\_f();

break;

}

case 2:

{

system("cls");

printf(" Информация о последнем элементе:\n");

print\_l();

break;

}

case 3:

{

system("cls");

int pos;

printf(" Введите позицию элемента, о котором хотите увидеть информацию: ");

pos=CheckINT();

print\_p(pos);

break;

}

}

system("PAUSE");

break;

}

//Перемещение

case 5:

{

system("cls");

int pos1, pos2;

\_gotoxy(1,6);

\_textcolor(10);

print\_all();

\_gotoxy(1,1);

printf("\n Для перемещения элемента введите его исходную позицию: ");

Изм.

Лист

№ докум.

Подпись

Дата

Лист

39

Курсовая работа

pos1=CheckINT();

printf("\n И новую позицию элемента: ");

pos2=CheckINT();

replace(pos1, pos2);

print\_all();

system("PAUSE");

break;

}

//Сортировка

case 6:

{

system("cls");

int N6Menu;

\_textcolor(10);

printf("\t\t\t\t\t\t===========================");

\_textbackground(16);

\_textcolor(10); printf("\n | "); \_textcolor(11); printf("1-ФИО; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("2-Дата рождения; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("3-Отдел; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("4-Должность; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("5-Стаж; ");

\_textcolor(10);

printf("|\n\t\t\t\t\t\t===========================");

\_gotoxy(1,11);

\_textcolor(10);

print\_all();

\_gotoxy(8,8);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

40

Курсовая работа

printf("\n Выберите по какому полю будем сортировать: ");

N6Menu=CheckINT();

switch(N6Menu)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

41

Курсовая работа

{

case 1:

{

system("cls");

\_textcolor(45);

printf("\t\t\t\t\tОтсортировано по Фамилии Имени Отчеству...\n\n");

sort\_fn();

break;

}

case 2:

{

system("cls");

\_textcolor(45);

printf("\t\t\t\t\t Отсортировано по Дате Рождения...\n\n");

sort\_date();

break;

}

case 3:

{

system("cls");

\_textcolor(45);

printf("\t\t\t\t\t\tОтсортировано по Отделу...\n\n");

sort\_office();

break;

}

case 4:

{

system("cls");

\_textcolor(45);

printf("\t\t\t\t\t\tОтсортировано по Должности...\n\n");

sort\_job();

break;

}

case 5:

{

system("cls");

\_textcolor(45);

printf("\t\t\t\t\t\tОтсортировано по Стажу...\n\n");

Изм.

Лист

№ докум.

Подпись

Дата

Лист

42

Курсовая работа

sort\_experience();

break;

}

}

\_textcolor(10);

print\_all();

system("PAUSE");

break;

}

//Запись в файл и выход

case 7:

{

system("cls");

int N7Menu;

\_textcolor(10);

printf("\t\t\t\t\t\t===========================");

\_textbackground(16);

\_textcolor(10); printf("\n | "); \_textcolor(11); printf("1-Сохранение в файл; ");

\_textcolor(10); printf("|\n | "); \_textcolor(11); printf("2-Выход; ");

\_textcolor(10);

printf("|\n\t\t\t\t\t\t===========================");

printf("\n Выберите действие: ");

N7Menu=CheckINT();

switch(N7Menu)

{

case 1:

{

system("cls");

out\_file2();

out\_file();

break;

}

case 2:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

43

Курсовая работа

{

system("cls");

return 0;

}

}

system("PAUSE");

break;

}

default: printf("Неверное число:\n");

}

}

}

Листинг 8 – модуль menu.c

Листинг модуля **check.c**:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

44

Курсовая работа

#include <stdio.h>

#include <locale.h>

#include <string.h>

#include <stdlib.h>

#include <conio.h>

#include <time.h>

#include <ctype.h>

typedef struct HUMAN

{

struct FullName

{

char surname[15];

char name[15];

char patronymic[15];

} FN;

struct Data

{

int day;

int month;

int year;

} DATA;

char office[50];

char job[50];

int experience;

struct HUMAN \*next;

} HUMAN;

void CheckFN(char \*f, char \*n, char \*p)

{

while (1)

{

char \*ff, \*nn, \*pp;

fflush(stdin);

gets(f);

fflush(stdin);

gets(n);

fflush(stdin);

gets(p);

ff = strpbrk(f, "0123456789.,?!{}[]()^:;<>'%#@$&+\*-/");

nn = strpbrk(n, "0123456789.,?!{}[]()^:;<>'%#@$&+\*-/");

Изм.

Лист

№ докум.

Подпись

Дата

Лист

45

Курсовая работа

pp = strpbrk(p, "0123456789.,?!{}[]()^:;<>'%#@$&+\*-/");

if (ff == NULL && nn == NULL && pp == NULL)

break;

printf(" Error. Повторите попытку: \n");

}

}

void CheckDATA(int \*d, int \*m, int \*y)

{

int nod=0,error,vis=0;

char month[3],day[3],year[5];

const char r[20];

do

{

nod=0;

error=0;

fflush(stdin);

gets(r);

if(strlen(r)!=10)

{

error=1;

}

if((isdigit(r[0]))&&(isdigit(r[1]))&&(r[2]=='.')&&(isdigit(r[3]))&&(isdigit(r[4]))&&(r[5]=='.')&&(isdigit(r[6]))&&(isdigit(r[7]))&&(isdigit(r[8]))&&(isdigit(r[9])))

{

day[0]=r[0];

day[1]=r[1];

month[0]=r[3];

month[1]=r[4];

year[0]=r[6];

year[1]=r[7];

year[2]=r[8];

year[3]=r[9];

\*d=atoi(day);

\*m=atoi(month);

\*y=atoi(year);

if(\*y>2016)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

46

Курсовая работа

{

error=1;

printf(" Error. Повторите попытку: ");

}

if(\*m>12)

{

error=1;

printf(" Error. Повторите попытку: ");

}

if((\*y%100!=0)&&(\*y%4==0))

{

vis=1;

}

if((\*m==1 || \*m==3 || \*m==5 || \*m==7 || \*m==8 || \*m==10 || \*m==12) && \*d>31)

{

error=1;

printf(" Error. Повторите попытку: ");

}

if((\*m==4 || \*m==6 || \*m==9 || \*m==11) && \*d>30)

{

error=1;

printf(" Error. Повторите попытку: ");

}

if(\*m==2 && vis!=1 && \*d>28)

{

error=1;

printf(" Error. Повторите попытку: ");

}

if(\*m==2 && vis==1 && \*d>29)

{

error=1;

printf(" Error. Повторите попытку: ");

}

}

else

Изм.

Лист

№ докум.

Подпись

Дата

Лист

47

Курсовая работа

{

printf(" Error. Повторите попытку: ");

error=1;

}

} while(error==1);

}

void CheckWORD(char \*str)

{

while ( 1 )

{

char \*ofc;

fflush(stdin);

gets(str);

ofc = strpbrk(str, "0123456789?!{}()^:;<>'%#@$&+\*-/");

if (ofc==NULL)

break;

printf (" Error. Повторите попытку: \n");

}

}

int CheckINT(void)

{

char exper[3];

int experi=0;

char ch;

while(experi<=0 || strlen(exper)>2)

{

exper[0]='\0';

exper[1]='\0';

exper[2]='\0';

fflush(stdin);

int i=0;

while(1)

{

ch=getchar();

if (ch>=48 && ch<=57)

exper[i]=ch;

if (ch=='\n')

{

experi=atoi(exper);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

48

Курсовая работа

break;

}

if (ch<48 || ch>57){ experi=-1; break;}

i++;

}

if(experi<=0 || strlen(exper)>2)

printf(" Error. Повторите попытку: ");

}

return experi;

}

Листинг 9 – модуль check.c

Листинг модуля **stack.c**:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

49

Курсовая работа

#include <stdio.h>

#include <locale.h>

#include <string.h>

#include <stdlib.h>

#include <conio.h>

#include <time.h>

#include <ctype.h>

void CheckFN(char \*, char \*, char \*);

void CheckDATA(int \*, int \*, int \*);

void CheckWORD(char \*);

int CheckINT(void);

typedef struct HUMAN

{

struct FullName

{

char surname[15];

char name[15];

char patronymic[15];

} FN;

struct Data

{

int day;

int month;

int year;

} DATA;

char office[50];

char job[50];

int experience;

struct HUMAN \*next;

} HUMAN;

int size = 0, size\_buf = 0;

HUMAN \*head = NULL, \*head\_buf = NULL;

int add(void); //Создание

int add\_l(void);//Ввод в конец

int add\_p(int);//Ввод в позицию

int del\_last(void);//Удаление

int del\_last\_buf(void);//Удаление из буфера

Изм.

Лист

№ докум.

Подпись

Дата

Лист

50

Курсовая работа

int in\_buf\_stack(void);//Ввод в буфер

int out\_buf\_stack(void);//Вывод из буфера

int print\_all(void);//Вывод всех

int print\_f(void);//Вывод первого

int print\_l(void);//вывод последнего

int print\_p(int);//Вывод по позиции

int del\_f(void);//Удаление последнего

int del\_p(int);//Удаление из позиции

int del\_full(void);//Полная очистка

int replace(int, int);//Перемещение

int sort\_fn(void);//Сортировка по ФИО

int sort\_date(void);//Сортировка по дате

int sort\_office(void);//Сортировка по офису

int sort\_job(void);//Сортировка по работе

int sort\_experience(void);//Сортировка по опыту

int copy(HUMAN\*, HUMAN\*);//Копирование

int out\_file(void);//Вывод в файл с псевдографикой

int out\_file2(void);//Вывод в файл

int file\_pop(void);//Чтение из файла

int in\_buf\_buf(HUMAN\*, int);

int out\_buf\_buf(HUMAN\*);

/\*Возвращение элемента пузырька в стек\*/

int out\_buf\_buf(HUMAN\* buf)

{

HUMAN \*retur\_n = (HUMAN\*)malloc(sizeof(HUMAN));

copy(retur\_n, buf);

retur\_n->next = head;

head = retur\_n;

size++;

return 0;

}

/\*Копирование верхнего элемента пузырька в отдельный буфер\*/

int in\_buf\_buf(HUMAN\* buf, int j)

{

for ( int g = 0; g < j-2; g++)

in\_buf\_stack();

copy(buf,head);

del\_last();

return 0;

}

/\*Перемещение\*/

Изм.

Лист

№ докум.

Подпись

Дата

Лист

51

Курсовая работа

int replace(int pos1, int pos2)

{

for (int i = 0; i < pos1-1; i++ )

in\_buf\_stack();

HUMAN \*start = (HUMAN\*)malloc(sizeof(HUMAN));

copy(start,head);

del\_last();

out\_buf\_stack();

for (int i = 0; i < pos2-1; i++ )

in\_buf\_stack();

HUMAN \*finish = (HUMAN\*)malloc(sizeof(HUMAN));

copy(finish,start);

finish->next = head;

head = finish;

size++;

out\_buf\_stack();

return 0;

}

/\*Сортировка по опыту\*/

int sort\_experience(void)

{

int size\_b = size;

for ( int i = 0; i < size\_b; i++ )

for ( int j = size\_b; j > 0; j-- )

{

HUMAN \*buf = (HUMAN\*)malloc(sizeof(HUMAN));

in\_buf\_buf(buf, j);

if ( head->experience >= buf->experience )

{

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( head->experience < buf->experience)

{

in\_buf\_stack();

out\_buf\_buf(buf);

out\_buf\_stack();

}

}

return 0;

}

/\*Сортировка по работе\*/

Изм.

Лист

№ докум.

Подпись

Дата

Лист

52

Курсовая работа

int sort\_job(void)

{

int size\_b = size;

for ( int i = 0; i < size\_b; i++ )

for ( int j = size\_b; j > 0; j-- )

{

HUMAN \*buf = (HUMAN\*)malloc(sizeof(HUMAN));

in\_buf\_buf(buf, j);

if ( strcmp( head->job, buf->job ) >= 0 )

{

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( strcmp( head->job, buf->job ) == -1 )

{

in\_buf\_stack();

out\_buf\_buf(buf);

out\_buf\_stack();

}

}

return 0;

}

/\*Сортировка по офису\*/

int sort\_office(void)

{

int size\_b = size;

for ( int i = 0; i < size\_b; i++ )

for ( int j = size\_b; j > 0; j-- )

{

HUMAN \*buf = (HUMAN\*)malloc(sizeof(HUMAN));

in\_buf\_buf(buf, j);

if ( strcmp( head->office, buf->office ) >= 0 )

{

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( strcmp( head->office, buf->office ) == -1 )

{

in\_buf\_stack();

out\_buf\_buf(buf);

out\_buf\_stack();

Изм.

Лист

№ докум.

Подпись

Дата

Лист

53

Курсовая работа

}

}

return 0;

}

/\*Сортировка по дате\*/

int sort\_date(void)

{

int size\_b = size;

for ( int i = 0; i < size\_b; i++ )

for ( int j = size\_b; j > 0; j-- )

{

HUMAN \*buf = (HUMAN\*)malloc(sizeof(HUMAN));

in\_buf\_buf(buf, j);

if ( head->DATA.year > buf->DATA.year )

{

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( head->DATA.year < buf->DATA.year )

{

in\_buf\_stack();

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( head->DATA.year == buf->DATA.year )

{

if ( head->DATA.month > buf->DATA.month )

{

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( head->DATA.month < buf->DATA.month )

{

in\_buf\_stack();

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( head->DATA.month == buf->DATA.month )

Изм.

Лист

№ докум.

Подпись

Дата

Лист

54

Курсовая работа

{

if ( head->DATA.day >= buf->DATA.day )

{

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( head->DATA.day < buf->DATA.day )

{

in\_buf\_stack();

out\_buf\_buf(buf);

out\_buf\_stack();

}

}

}

}

return 0;

}

/\*Сортировка по ФИО\*/

int sort\_fn(void)

{

int size\_b = size;

for ( int i = 0; i < size\_b; i++ )

for ( int j = size\_b; j > 0; j-- )

{

HUMAN \*buf = (HUMAN\*)malloc(sizeof(HUMAN));

in\_buf\_buf(buf, j);

if ( strcmp( head->FN.surname, buf->FN.surname ) == 1 )

{

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( strcmp( head->FN.surname, buf->FN.surname ) == -1 )

{

in\_buf\_stack();

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( strcmp( head->FN.surname, buf->FN.surname ) == 0 )

Изм.

Лист

№ докум.

Подпись

Дата

Лист

55

Курсовая работа

{

if ( strcmp( head->FN.name, buf->FN.name ) == 1 )

{

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( strcmp( head->FN.name, buf->FN.name ) == -1 )

{

in\_buf\_stack();

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( strcmp( head->FN.name, buf->FN.name ) == 0 )

{

if ( strcmp( head->FN.patronymic, buf->FN.patronymic ) >= 0 )

{

out\_buf\_buf(buf);

out\_buf\_stack();

}

else if ( strcmp( head->FN.patronymic, buf->FN.patronymic ) == -1 )

{

in\_buf\_stack();

out\_buf\_buf(buf);

out\_buf\_stack();

}

}

}

}

return 0;

}

/\*Полная очистка\*/

int del\_full(void)

{

int size\_b = size;

for ( int i = 0; i < size\_b; i++ )

del\_last();

return 0;

}

/\*Удаление из позиции\*/

Изм.

Лист

№ докум.

Подпись

Дата

Лист

56

Курсовая работа

int del\_p(int pos)

{

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

in\_buf\_stack();

del\_last();

out\_buf\_stack();

return 0;

}

/\*Удаление последнего элемента\*/

int del\_f(void)

{

int size\_b = size-1;

for (int i = 0; i < size\_b; i++ )

in\_buf\_stack();

del\_last();

out\_buf\_stack();

return 0;

}

/\*Добавление в позицию\*/

int add\_p(int pos)

{

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

in\_buf\_stack();

add();

out\_buf\_stack();

return 0;

}

/\*Ввод в последнюю позицию\*/

int add\_l(void)

{

int size\_b=size;

for ( int i=0; i<size\_b; i++ )

in\_buf\_stack();

add();

out\_buf\_stack();

return 0;

}

/\*Вывод по позиции\*/

int print\_p(int pos)

{

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

Изм.

Лист

№ докум.

Подпись

Дата

Лист

57

Курсовая работа

in\_buf\_stack();

\_textcolor(8);

printf("\n---------------------------------------------------------------------------------------------------------------------\n");

\_textcolor(10);

printf("\t ФИО: ");\_textcolor(12);printf("%s %s %s\n", head->FN.surname, head->FN.name, head->FN.patronymic);

\_textcolor(10);

div\_t d1,m1;

d1=div(head->DATA.day,10);

m1=div(head->DATA.month,10);

if ((d1.quot>=1) && (m1.quot==1)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("%d.%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==0)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("0%d.0%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==1)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("0%d.%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot>=1) && (m1.quot==0)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("%d.0%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

\_textcolor(10);

printf("\n\n\t Отдел: ");\_textcolor(12);printf("%s\n", head->office);

\_textcolor(10);

printf("\n\t Должность: ");\_textcolor(12);printf("%s\n", head->job);

\_textcolor(10);

printf("\n\t Cтаж: ");\_textcolor(12);printf("%d\n", head->experience);

\_textcolor(8);

printf("---------------------------------------------------------------------------------------------------------------------\n");

\_textcolor(10);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

58

Курсовая работа

out\_buf\_stack();

return 0;

}

/\*Вывод последнего\*/

int print\_l(void)

{

int size\_b = size-1;

for ( int i=0; i<size\_b; i++ )

in\_buf\_stack();

\_textcolor(8);

printf("\n---------------------------------------------------------------------------------------------------------------------\n");

\_textcolor(10);

printf("\t ФИО: ");\_textcolor(12);printf("%s %s %s\n", head->FN.surname, head->FN.name, head->FN.patronymic);

\_textcolor(10);

div\_t d1,m1;

d1=div(head->DATA.day,10);

m1=div(head->DATA.month,10);

if ((d1.quot>=1) && (m1.quot==1)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("%d.%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==0)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("0%d.0%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==1)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("0%d.%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot>=1) && (m1.quot==0)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("%d.0%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

\_textcolor(10);

printf("\n\n\t Отдел: ");\_textcolor(12);printf("%s\n", head->office);

\_textcolor(10);

printf("\n\t Должность: ");\_textcolor(12);printf("%s\n", head->job);

\_textcolor(10);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

59

Курсовая работа

printf("\n\t Cтаж: ");\_textcolor(12);printf("%d\n", head->experience);

\_textcolor(8);

printf("---------------------------------------------------------------------------------------------------------------------\n");

\_textcolor(10);

out\_buf\_stack();

return 0;

}

/\*Вывод первого\*/

int print\_f(void)

{

\_textcolor(8);

printf("\n---------------------------------------------------------------------------------------------------------------------\n");

\_textcolor(10);

printf("\t ФИО: ");\_textcolor(12);printf("%s %s %s\n", head->FN.surname, head->FN.name, head->FN.patronymic);

\_textcolor(10);

div\_t d1,m1;

d1=div(head->DATA.day,10);

m1=div(head->DATA.month,10);

if ((d1.quot>=1) && (m1.quot==1)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("%d.%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==0)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("0%d.0%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==1)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("0%d.%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot>=1) && (m1.quot==0)) {

\_textcolor(10);printf("\n\t Дата рождения: ");\_textcolor(12);printf("%d.0%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

\_textcolor(10);

printf("\n\n\t Отдел: ");\_textcolor(12);printf("%s\n", head->office);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

60

Курсовая работа

\_textcolor(10);

printf("\n\t Должность: ");\_textcolor(12);printf("%s\n", head->job);

\_textcolor(10);

printf("\n\t Cтаж: ");\_textcolor(12);printf("%d\n", head->experience);

\_textcolor(8);

printf("---------------------------------------------------------------------------------------------------------------------\n");

\_textcolor(10);

return 0;

}

/\*Вывод всех блоков\*/

int print\_all(void)

{

int size\_b = size;

int x, j;

int number=0;

printf("---------------------------------------------------------------------------------------------------------------------\n");

\_textcolor(10);printf("|");\_textcolor(12);printf(" №");

\_textcolor(10);printf("|");\_textcolor(12);printf(" Фамилия Имя Отчество ");

\_textcolor(10);printf("|");\_textcolor(12);printf(" Дата рождения ");

\_textcolor(10);printf("|");\_textcolor(12);printf(" Отдел ");

\_textcolor(10);printf("|");\_textcolor(12);printf(" Должность ");

\_textcolor(10);printf("|");\_textcolor(12);printf(" Стаж ");

\_textcolor(10);printf("|");

for (int i=0; i<size\_b; i++)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

61

Курсовая работа

{

x=3; j=4;

\_textcolor(10);printf("\n---------------------------------------------------------------------------------------------------------------------\n");

\_textcolor(10);printf("|");\_textcolor(11);printf("%2d",number+1);

\_textcolor(10);printf("|");\_textcolor(11);printf("%14s %13s %14s",head->FN.surname,head->FN.name,head->FN.patronymic);

div\_t d1,m1;

d1=div(head->DATA.day,10);

m1=div(head->DATA.month,10);

if ((d1.quot>=1) && (m1.quot==1)) {

\_textcolor(10);printf("|");\_textcolor(11);printf("%5 %d.%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==0)) {

\_textcolor(10);printf("|");\_textcolor(11);printf("%5 0%d.0%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==1)) {

\_textcolor(10);printf("|");\_textcolor(11);printf("%5 0%d.%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot>=1) && (m1.quot==0)) {

\_textcolor(10);printf("|");\_textcolor(11);printf("%5 %d.0%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

\_textcolor(10);printf("|");\_textcolor(11);printf("%23s",head->office); \_textcolor(10);printf("|");\_textcolor(11);printf("%21s",head->job);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

62

Курсовая работа

\_textcolor(10);printf("|");\_textcolor(11);printf("%6d",head->experience);

\_textcolor(10);printf("|");

x++;

x++;

j++;

j++;

number++;

in\_buf\_stack();

}

printf("\n---------------------------------------------------------------------------------------------------------------------\n");

out\_buf\_stack();

return 0;

}

/\*Вывод в файл с псевдографикой\*/

int out\_file(void)

{

int x, j;

int size\_b = size;

FILE \*out;

out=fopen("detsad.txt", "w");

fprintf(out,"---------------------------------------------------------------------------------------------------------------------\n");

fprintf(out,"|");fprintf(out," Фамилия Имя Отчество ");

fprintf(out,"|");fprintf(out," Дата рождения ");

fprintf(out,"|");fprintf(out," Отдел ");

fprintf(out,"|");fprintf(out," Должность ");

fprintf(out,"|");fprintf(out," Стаж ");

fprintf(out,"|");

for (int i=0; i<size\_b; i++)

{

x=3; j=4;

Изм.

Лист

№ докум.

Подпись

Дата

Лист

63

Курсовая работа

fprintf(out,"\n---------------------------------------------------------------------------------------------------------------------\n");

fprintf(out,"|");fprintf(out,"%15s %14s %15s",head->FN.surname,head->FN.name,head->FN.patronymic);

div\_t d1,m1;

d1=div(head->DATA.day,10);

m1=div(head->DATA.month,10);

if ((d1.quot>=1) && (m1.quot==1)) {

fprintf(out,"|");fprintf(out,"%5 %d.%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==0)) {

fprintf(out,"|");fprintf(out,"%5 0%d.0%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==1)) {

fprintf(out,"|");fprintf(out,"%5 0%d.%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot>=1) && (m1.quot==0)) {

fprintf(out,"|");fprintf(out,"%5 %d.0%d.%d",head->DATA.day,head->DATA.month,head->DATA.year);}

fprintf(out,"|");fprintf(out,"%23s",head->office);

fprintf(out,"|");fprintf(out,"%21s",head->job);

fprintf(out,"|");fprintf(out,"%6d",head->experience);

fprintf(out,"|");

x++;

x++;

j++;

j++;

in\_buf\_stack();

}

fprintf(out,"\n---------------------------------------------------------------------------------------------------------------------\n");

Изм.

Лист

№ докум.

Подпись

Дата

Лист

64

Курсовая работа

out\_buf\_stack();

fclose (out);

return 0;

}

/\*Вывод в файл\*/

int out\_file2(void)

{

int size\_b = size;

FILE \*out;

out=fopen("detsad2.txt", "w");

for (int i=0; i<size\_b; i++)

{

fprintf(out,"%s\n%s\n%s\n",head->FN.surname,head->FN.name,head->FN.patronymic);

div\_t d1,m1;

d1=div(head->DATA.day,10);

m1=div(head->DATA.month,10);

if ((d1.quot>=1) && (m1.quot==1)) {

fprintf(out,"%d.%d.%d\n",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==0)) {

fprintf(out,"0%d.0%d.%d\n",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot==0) && (m1.quot==1)) {

fprintf(out,"0%d.%d.%d\n",head->DATA.day,head->DATA.month,head->DATA.year);}

if ((d1.quot>=1) && (m1.quot==0)) {

fprintf(out,"%d.0%d.%d\n",head->DATA.day,head->DATA.month,head->DATA.year);}

fprintf(out,"%s\n",head->office);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

65

Курсовая работа

fprintf(out,"%s\n",head->job);

fprintf(out,"%d\n",head->experience);

in\_buf\_stack();

}

out\_buf\_stack();

fclose (out);

return 0;

}

/\*Чтение из файла\*/

int file\_pop(void)

{

FILE \*off;

char exp[3];

unsigned char data[10];

char month[3],day[3],year[5];

// char \*pstr;

off = fopen("detsad2.txt", "r");

while (1)

{

HUMAN \*new = (HUMAN \*)malloc(sizeof(HUMAN));

// pstr = fgets(pstr, 1, off);

fscanf(off,"%s",new->FN.surname);

fscanf(off,"%s",new->FN.name);

fscanf(off,"%s",new->FN.patronymic);

fscanf(off,"%s",data);

if((isdigit(data[0]))&&(isdigit(data[1]))&&(data[2]=='.')&&(isdigit(data[3]))&&(isdigit(data[4]))&&(data[5]=='.')&&(isdigit(data[6]))&&(isdigit(data[7]))&&(isdigit(data[8]))&&(isdigit(data[9])))

{

day[0]=data[0];

day[1]=data[1];

month[0]=data[3];

month[1]=data[4];

year[0]=data[6];

year[1]=data[7];

year[2]=data[8];

year[3]=data[9];

new->DATA.day=atoi(day);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

66

Курсовая работа

new->DATA.month=atoi(month);

new->DATA.year=atoi(year);

}

fscanf(off,"%s",new->office);

fscanf(off,"%s",new->job);

fscanf(off,"%s",exp);

new->experience=atoi(exp);

new->next = head;

head = new;

size++;

// if (pstr == NULL)

// {

if (feof(off) != 0) //feof – проверка наличия признака конца файла

{

break;

}

// }

}

fclose(off);

return 0;

}

/\*Возврат из буфера\*/

int out\_buf\_stack(void)

{

int size\_b = size\_buf;

for (int i=0; i<size\_b; i++ )

{

HUMAN \*new = (HUMAN\*)malloc(sizeof(HUMAN));

copy(new,head\_buf);

del\_last\_buf();

new->next = head;

head = new;

size++;

}

return 0;

}

/\*Копирование в буфер\*/

int in\_buf\_stack(void)

{

HUMAN \*new\_buf = (HUMAN\*)malloc(sizeof(HUMAN));

copy(new\_buf,head);

Изм.

Лист

№ докум.

Подпись

Дата

Лист

67

Курсовая работа

del\_last();

new\_buf->next = head\_buf;

head\_buf = new\_buf;

size\_buf++;

return 0;

}

/\*Удаление последнего введённого в буфер\*/

int del\_last\_buf(void)

{

HUMAN \*delete = head\_buf;

head\_buf = head\_buf->next;

free(delete);

size\_buf--;

return 0;

}

/\*Удаление последнего введённого\*/

int del\_last(void)

{

HUMAN \*delete = head;

head = head->next;

free(delete);

size--;

return 0;

}

/\*Копирование\*/

int copy(HUMAN \*finish, HUMAN \*start)

{

strcpy(finish->FN.surname,start->FN.surname);

strcpy(finish->FN.name,start->FN.name);

strcpy(finish->FN.patronymic,start->FN.patronymic);

finish->DATA.day = start->DATA.day;

finish->DATA.month = start->DATA.month;

finish->DATA.year = start->DATA.year;

strcpy(finish->office,start->office);

strcpy(finish->job,start->job);

finish->experience = start->experience;

return 0;

}

/\* Добавление нового блока \*/

int add(void)

{

HUMAN \*new = (HUMAN\*)malloc(sizeof(HUMAN));

\_textcolor(14);

\_textbackground(9);

printf("----------------------------------------------------------------------------------------------------------------------");

Изм.

Лист

№ докум.

Подпись

Дата

Лист

68

Курсовая работа

printf(" Введите данные по сотруднику ");

printf("----------------------------------------------------------------------------------------------------------------------");

\_textcolor(10);

\_textbackground(0);

printf("\n\tВведите ФИО сотрудника: \n");

\_textcolor(44);

CheckFN(new->FN.surname,new->FN.name,new->FN.patronymic);

\_textcolor(10);

\_textbackground(0);

printf("\n\tВведите дату рождения в формате \"дд.мм.гггг\": ");

\_textcolor(44);

int d=0,m=0,y=0;

CheckDATA(&d, &m, &y);

new->DATA.day=d;

new->DATA.month=m;

new->DATA.year=y;

\_textcolor(10);

\_textbackground(0);

printf("\n\tВведите отдел в котором он(а) работает: ");

\_textcolor(44);

CheckWORD(new->office);

\_textcolor(10);

\_textbackground(0);

printf("\n\tВведите должность: ");

\_textcolor(44);

CheckWORD(new->job);

\_textcolor(10);

\_textbackground(0);

printf("\n\tВведите стаж работы: ");

\_textcolor(44);

new->experience=CheckINT();

Изм.

Лист

№ докум.

Подпись

Дата

Лист

69

Курсовая работа

new->next = head;

head = new;

size++;

return 0;

}

Листинг 10 – модуль stack.c