

**Методические указания к лабораторной работе № 10 (КЛР/ДЗ)
по курсу Основы программирования (СУЦ)**

**" Структуры, указатели, файлы, массивы, алгоритмы и
списки "**

СОДЕРЖАНИЕ

| | |
|---|----|
| 1. Цель ЛР №10 | 3 |
| 2. Требования к заданию на ЛР №10 | 3 |
| 3. Особенности описания разделения заданий по уровням ЛР №10 | 3 |
| 4. Функции и структуры в домашнем задании | 3 |
| 5. Варианты для выполнения ЛР | 4 |
| 6. Порядок выполнения работы (Уровень А и В)..... | 5 |
| 7. Дополнительные требования для сильных студентов (Уровень С): | 9 |
| 8. Требование к отчету по ЛР | 10 |
| 9. Контрольные вопросы к ДЗ/КЛР | 10 |
| 10. Литература..... | 11 |
| 11. Приложения..... | 12 |
| 11.1. Главный модуль проекта LR10_OP.cpp | 12 |
| 11.2. Заголовочный файл проекта LR10_OP.h..... | 24 |
| 11.3. Результаты работы главного модуля примеров | 34 |

1. Цель ЛР №10

Комплексная лабораторная работа/Домашнее задание (КЛР/ДЗ) № 10 выполняется для получения навыков совместного использования знаний по отдельным приемам программирования, изучаемым в курсе основы программирования, после выполнения комплекса отдельных лабораторных работ дисциплины “**Основы программирования**”.

В предыдущих работах студенты изучают следующие темы: операторы, структуры, указатели, строки, файлы, массивы, алгоритмы и списки. Студенты создают файл своих структурных переменных (набор записей) и комплекс специальных функций-операций для работы с этим файлом и этими записями. Предусматриваются важные операции с файлом: запись файла, его распечатки, его сортировки, доступа к записям в файле, изменение записей, поиск записей и перезапись его в массив записей и обратно. Фактически, на уровне файлов, студенты осваивают работу с прообразом однотабличной базой данных (БД), что в дальнейшем поможет более глубоко понять функционирование сложных систем управления базами и банками данных (СУБД).

2. Требования к заданию на ЛР №10

Данное описание ЛР и задание на ЛР может быть выполнено на трех уровнях сложности, в зависимости от выбора студента:

- базовый уровень (уровень А),
- продвинутый уровень (уровень В),
- уровень для сильных студентов (уровень С).

Пункты задания выделены цветом, соответствующим выбранному уровню. В дальнейшем, конкретизирую требования на ЛР, мы будем использовать эти цвета для привязки к одному из уровней задания. Проще - цвет пункта задания определяет его уровень сложности. Студент может и частично использовать пункты задания из разных уровней, однако **базовый уровень** является обязательным.

Примечание 1. Данную лабораторную работу (№ 10) мы будем также именовать комплексной лабораторной работой (КЛР) либо домашним заданием (ДЗ), так как в отдельных учебных планах на семестр иногда появляется такая позиция. В общем возможен и комбинированный вариант КЛР/ДЗ.

3. Особенности описания разделения заданий по уровням ЛР №10

Описание шагов выполнения КЛР и требований к ней намеренно предлагается в двух вариантах: вариант (№1), в котором примеры располагаются отдельно от текста и вариант (№2), в котором примеры и решения вставлены в текст методических указаний. На сайте для работы студентам доступны оба варианта. Если студент стремится и способен получить максимальный уровень знаний (даже более чем по плану) он должен попробовать использовать первый вариант методических указаний и обращаться к подсказкам во втором варианте только в крайнем случае. Если студенту пока трудно самостоятельно выполнить какой-то конкретный шаг задания, то он может использовать подсказку (извините, шпаргалку) и на основе ее сделать собственное задание.

4. Функции и структуры в домашнем задании

5. Техническое задание (ТЗ) для КЛР

Для выполнения ДЗ/КЛР (Комплексной работы) по дисциплине ОП нужно разработать техническое задание (ТЗ). Оно выполняется на основе документов (на сайте в архиве):

- Методические указания ЛР 10(ТЗ).docx

- Образец ТЗ - Техническое задание(ТЗ).docx
- ШаблонТехническое задание(ТЗ).doc
- Общие методические указания по дисциплине ОП - METHOD_OP_2017.docx
- Данный документ - Методические указания ЛР10_2017.docx

Все названные документы должны быть изучены.

Первоначально в ТЗ описывается структура, используемая в задании (см. ЛР№6), часть сведений о структуре включаются в раздел 5.1 ТЗ (технические/функциональные требования) – основной раздел ТЗ.

Разработанное техническое задание (ТЗ) согласуется с преподавателем и утверждается у него для дальнейшей работы на ДЗ. Только после этого можно приступить к выполнению ДЗ. Более подробно о разработке ТЗ будет сказано на лекциях и семинарах.

Как было отмечено выше, студенты в рамках ЛР разрабатывают программный комплекс, основанный на файловой системе, иллюстрирующий основные операции работы с базами данных (БД). Программный комплекс состоит из: набора специальных функций (фактически библиотеки), и фрагментов программ, для демонстрации работы этих функций. Например, должны быть разработаны функции заполнения файла записями, его распечатки, изменения записей и другие. Кроме этого, в главном модуле могут быть реализованы некоторые действия с файлом без оформления функций (в каждом пункте заданий указывается, когда нужно создавать функцию). Работа с файлами основывается на специальной структуре данных, которую студент самостоятельно разрабатывает в предложенной предметной области (см. варианты задания). Структура данных должна иметь осмысленные поля и представлять единое целое. Например, для описания студента можно выделить поля: фамилия студента, курс студента, размер стипендии и т.д. Часть названий полей предложены в варианте (см. ниже), хотя по согласованию с преподавателем их можно заменить. Неизменным должно оставаться только содержательное название структуры данных (студент, книга и т.д.).

6. Варианты для выполнения ЛР

Варианты для выполнения работы студентами представлены в таблице, в дополнение к трем указанным полям в структуре студент должен добавить еще любые два дополнительных поля, которые он придумает сам. Эти поля должны иметь смысл иметь смысл для собственного проекта. Например, для структуры типа студент можно добавить поля: курс обучения, год поступления, процент посещения занятий и т.д.

В таблице расположенной ниже приведены основные названия структурных переменных и обязательных полей по вариантам.

| № Вар. | Структура (имя структуры придумать самому) | Поля структуры и их тип | | | Примечание (интегральная характеристика) |
|--------|--|-------------------------------|----------------------------------|-------------------------------------|--|
| | | Поле-имя-тип | Поле-имя-тип | Поле-имя-тип | |
| 1. | Кафедра | Название (Name) –char[] | Число студентов (CountS) - int | Число преподавателей (CountP) - int | Среднее Число студентов на кафедре |
| 2. | Книга | Название (Name) –char[] | Автор (Avtor) – char[] | Число страниц (nCountS) - int | Среднее Число страниц |
| 3. | Файл | Имя файла (NameFile) – char[] | Дата создания (DateFile)– char[] | Размер файла (SizeFile) - int | Средний Размер файла |

| № Вар. | Структура (имя структуры придумать самому) | Поля структуры и их тип | | | Примечание (интегральн. характеристика) |
|---|--|-----------------------------------|--------------------------------|--|---|
| | | Поле-имя-тип | Поле-имя-тип | Поле-имя-тип | |
| 4. | Автомобиль | Марка автомобиля (Marka) – char[] | Стоимость (Cost) - double | Мощность (Kraft) - float | Средняя Мощность |
| 5. | Компьютер | Владелец(Fam) – char[] | Размер ОП (SizeMem) - int | Объем HDD (SizeHDD) - int | Средний Объем |
| 6. | Группа | Индекс группы (Name) – char[] | Число студентов (CountS) - int | Средняя оценка в группе в % (AvExam) - float | Среднее число студентов в группе |
| 7. | Человек | Фамилия (Fam) – char[] | Пол (Pol) - char | Возраст (Age) - int | Средний Возраст |
| 8. | Стеллаж | Название (Namt) – char[] | Материал (Mat) - char | Число полок (nPol) - int | Среднее число полок |
| 9. | Дом | Улица (Fam) – char[] | Число этажей (nStage) - int | Номер дома (Numb)- char[] | Среднее число этажей |
| Вариант представленный в примере выполнения ЛР № 10 | | | | | |
| | Студент (Student) | Имя (Name) - char[20] | Номер (Num) - int | Стипендия (Oklad)-double | Задание моего примера |

Примечание 2: Номер варианта студента уточняется по журналу группы. При желании студент может предложить свой вариант темы структурной переменной, но при этом он должен согласовать его с преподавателем.

7. Порядок выполнения работы (Уровень А и В)

0.Разработать и утвердить техническое задание на программу ДЗ.

Важное значение имеет структура, используемая в программе, поэтому ее нужно согласовать с преподавателем отдельно. (см. ЛР №6).

1. Создать консольный проект (главная программа, вспомогательный модуль для функций и заголовочный файл для структур/констант)

Для создания консольного проекта необходимо:

- Запустить систему программирования VS 2005/8/10/12;
- В меню “File” выбрать пункт “New” и в подменю выбрать позицию “Project...”;;
- В списке “Project types” выбрать “Visual C++/Win32”, а в списке “Templates” выбрать “Win32 Console Application”;
- В поле “Name” ввести: LAB1_XDD (где X – номер группы, а DD – номер варианта по журналу группы текущего семестра. Например, для студента группы ИУ5-31 с вариантом 5 – введем – LAB1_15). Далее нажать “OK”;
- В новом окне мастера проектов нажать “Next”. Проверить настройки проекта: “Application Type” должно быть – “Console Application”,

“Additional option” -> “Empty Project” должен быть включено. Остальные галочки должны быть выключены.

- Далее необходимо нажать кнопку “**Finish**”. Новый проект будет создан.
- Необходимо убрать из главных моделей проекта (LAB1_XDD.CPP и LAB1_XDD.H) все лишнее. Этого: в файле LAB1_XDD.H (у нас в примере LAB1_15.H) уберем все, а в файле LAB1_XDD.CPP (LAB1_15.CPP) оставим только следующий текст:

```
#include "lab10_OP.h"
#include <process.h>
#include <stdio.h>
...
void main(void)
{ ... }
```

- Для контроля правильности создания пустого проекта, нажмем клавишу “F7” для проверки создания программы (build) и “F5” для проверки ее выполнения (run/debug). Все перечисленные действия должны быть выполнены безошибочно.

Для корректного отображения текстов на русском языке и его ввода в окне командной строки (после первого запуска программы) нужно сделать настройки шрифта этого окна. Переключаем шрифт в тип - **Lucida Console**. Выбираем настройки (после вывода консольного окна на экран, правой кнопкой вызываем системное меню): СВОЙСВА->ШРИФТ -> **Lucida Console**). После переключения шрифта, на запрос в отдельном окошке нужно выбрать режим – “Для всех окон с данным именем!”. Для правильной русификации окна консоли, кроме этого, в самом начале главной программы нужно переключить кодовую страницу для вывода:

```
system(" chcp 1251 > nul");
```

Для приостановки завершения программы в консольном окне в конце ее работы можно вызвать паузу следующим образом (например, в конце текста программы):

```
system(" PAUSE");
```

На экране появиться следующая строка (смотри ниже) и программа будет ожидать нажатия клавиши:

Для продолжения нажмите любую клавишу . . .

Обратите внимание на то, что при другом способе локализации (setlocale(0,"rus");) не все работает правильно. Вывод на консоль и ввод с консоли выполняется правильно, но после этого введенные в консольном окне данные (например, строка) имеют другую кодировку и выводятся неверно! Можете сами это проверить. Поэтому предпочтительно использовать предложенный выше способ с переключением кодовой страницы.

2. **Описать** свою структуру, придумав ее **название**. Типы и названия полей взять из таблицы вариантов (см. ниже). Все данные структуры статические. Дополнить описание структуры двумя полями, придумав их по смыслу. Описание структуры выполнить в заголовочном файле проекта (<имя проекта>.h).

```
(*h).
//////// ЗАГОЛОВОЧНЫЙ ФАЙЛ
// МУ п.2
////////
struct Student { // Описание студента - структура
    char Name[20]; // Фамилия студента
    int Num; // Номер студента
    double Oklad; // Размер стипендии
    int Kurs; // Резерв пока в примере нет для краткости изложения
    char Date[10]; // Резерв пока в примере нет для краткости
};
...
```

3. **Описать** и заполнить одну свою простую структурную переменную с помощью инициализации при ее описании.
4. **Описать** и заполнить еще одну свою структурную переменную вручную (с помощью операторов присваивания – числовые данные и функции копирования – строчные данные). Распечатать значения ее полей двух структурных переменных (Функция - **printf**).
5. Разработать **функцию** распечатки одной данной структурной переменной, передавая в качестве параметра ее адрес, и проверить ее на описанных ранее структурных переменных.
6. Создать динамическую структурную (функция – malloc, библиотека - **<malloc.h>**) заполнить ее динамически и распечатать значение через указатель и с помощью специальной функции печати. Удалить созданную динамическую переменную.
7. **Описать** и проинициализировать массив структурных переменных не менее 4-х. Распечатать этот массив с помощью функции п.5.
8. **Разработать функцию распечатки массива** с записями структурных переменных. Формальные параметры этой функции: указатель на массив структурных переменных и его размер. Использовать свою функцию печати одной структурной переменной. Проверить работу этой функции.
9. Создать цикл для заполнения нового динамического **массива** записей полями со случайными значениями параметров (размер 5-6 элементов). Предусмотреть случайное заполнение одного числового значения (использовать функцию - **rand**) и одного символьного значения (использовать функции: **rand** и **atoi** вместе с функциями копирования/слияния текстов - **strcat, strcpy**). Распечатать созданный массив в этом же цикле.
10. **Придумать** название двоичному файлу для выполнения задания (в нашем примере - "BDStud.bin", например **"BDBook.bin"** – для книг).
11. Создать цикл для заполнения своего файла своими структурными записями (запись в файл) с разными значениями числовых параметров изменяемых в цикле (номер в журнале группы) или основанных на начальной инициализации своего массива структур (в нашем примере на основе массива, который проинициализирован).
11. **Создать** цикл чтения и распечатки сформированного двоичного файла. Данные из файла ввести в статический массив структур, задав его максимальный размер равным 100.
11. Создать цикл для заполнения файла записей полями со случайными значениями параметров. Предусмотреть случайное заполнение одного числового значения (использовать функцию - **rand**) и одного символьного значения (использовать функции: **rand** и **atoi** вместе с функциями копирования/слияния текстов - **strcat, strcpy**). Распечатать созданный файл в цикле заполнения.

12. Разработать функцию заполнения динамического массива структурных переменных на основе двоичного файла с записями структурных переменных. Проверить ее работу. Массив распечатать специальной функцией (см. выше).

13. **Разработать функцию** заполнения двоичного **файла** на основе массива структурных переменных (описанных и проинициализированных выше). В функцию должны передаваться: имя заполняемого файла, адрес исходного массива, размер массива. Изменить некоторые значения записей в массиве структур перед использованием. Для этого в самом массиве, заполненном выше, сделать изменения (например, имя 2-й записи: имя и номер), используя фиксированное значение индекса элемента структурного массива. Проверить работу этой функции. Заполнение файла проверить в файл менеджере (**far** или **total commander**), с помощью операции просмотра файла в шестнадцатеричном формате.

14. **Разработать** отдельную функцию распечатки файла с записями своих структурных переменных. Проверить работу этой функции.

15. Разработать функцию определения числа записей в файле. Проверить ее работу на своем файле.

16. **Разработать функцию** очистки файла. Проверить ее работу совместно с функцией распечатки файла записей – ничего не должно напечататься. Кроме этого проверить размер файла в файл - менеджере (Far, TotalComander и др.), этот размер должен быть равен нулю.

17. Проверить использование функций удаления файла и снятия атрибута защиты файла, взяв ее из проекта примера. Заполним снова и распечатаем.

18. **Разработать функцию Swap** двух структурных переменных (по адресу элемента) структурных переменных. Структурные переменные имеют фиксированный размер. Продемонстрировать и проверить работу этой функции на отдельных переменных и элементах массива.

19. На основе этой функции (**Swap**) выполнить сортировку массива структурных переменных по любому числовому значению своей структуры (**int, long, float или double**), используя метод пузырьковой сортировки. Оптимизацию алгоритма сортировки не проводить. Распечатать массив до и после сортировки.

20. Разработать **функцию** сортировки **массива** структурных переменных по числовому значению. Проверить ее работу. Распечатать массив до и после сортировки.

21. Разработать функцию сортировки файла структурных переменных (записей) по полю структурной переменной (см. вариант). Алгоритм функции включает три шага: (1) чтение

двоичного файла в массив – уже есть готовая функция; (2) пузырьковая сортировка массива структурных переменных – уже есть фрагмент текста и готовая функция и (3) перезапись массива в двоичный файл – уже есть готовая функция. Проверить сортировку файлов со структурными переменными.

22. Прочитать 2-ю запись из сформированного файла структурных переменных и ее распечатать с помощью собственной функции.

22. Создать функцию для чтения из сформированного файла структурных переменных отдельной записи по номеру(параметр функции) и ее распечатать с помощью собственной функции.

22. Разработать фрагмент программы для интегральных вычислений в БД (см. варианты в таблице). Интегральные вычисления для вашего файла структурных переменных заключаются в последовательном чтении всех записей файла. Из каждой из полученных записей для отдельного параметра выполняется суммирование, подсчет, сравнение и т.д. Шаги выполнения этой задачи могут быть такими: чтение файла в массив, вычисления по массиву в цикле (например, числа пятерок), обработка результата (например, получение среднего числа пятерок у студента) и распечатка результата. Можно в цикле по шагам читать последовательно все записи и вычислять интегральные характеристики.

22. Оформить в виде **функции** фрагмент программы для вычисления интегральной характеристики по вашему варианту (см. предыдущее задание).

8. Дополнительные требования для сильных студентов (Уровень С):

Варианты заданий и примеры рассмотрены для случая использования контейнера типа **массив** для хранения записей в программе. Студентам, которые хотят освоить программирование на самом высоком уровне предлагается также самостоятельно решить комплекс заданий, перечисленных ниже для контейнера типа **список**, сделанных на базе ЛР № 7 и 8.

23. Разработать специальные функции для нашей структуры и файла БД, содержащего записи структурных переменных, для следующих операций:

23.1 Добавления записи в файл БД (в начало).

23.2 Добавления записи в файл БД (в конец).

23.3 Добавления записи в файл БД (по номеру записи).

23.4 Удаления записи в файл БД (из начала).

23.5 Удаления записи в файл БД (в конец).

23.6 Удаления записи в файл БД (по номеру записи).

23.7 Поиска и чтения записи из файла БД (по номеру записи).

23.8 Замены записи в файле БД по номеру (вместо одной другая)

23.9 Поиска и выборки по конкретному значению поля имени.. – Возврат номера!!!

23.10 Взаимного обмена (swap!) записей по двум номерам в файле БД.

23.11 Удаления файла БД.

Для всех функций и операций выполнить качественную проверку ее работы: отладку и тестирование во всех режимах. Для самых продвинутых студентов предлагается вариант замены массива на двунаправленные списки во всех пунктах заданий ЛР.

9. Требование к отчету по ЛР

Требования к отчету по 10 ЛР соответствуют требованиям к другим ЛР по дисциплине ОП. См. Документ на сайте.

10. Контрольные вопросы к ДЗ/КЛР

1. Для чего нужны проекты и в чем их преимущество (три)?
2. Дайте определение понятия программа.
3. Что такое сборка программы и как ее сделать?
4. Какие составляющие циклов вы знаете?
5. В чем преимущество возможности ветвления в программах?
6. Какие элементы блок схем вы знаете?
7. Что такое динамические массивы?
8. Какие новые операции для указателей вы знаете?
9. Как вычислить размер массива при выполнении программы?
10. Как определить длину строки?
11. Как сравнить две строки?
12. Что такое **Null Terminated String**?
13. Как преобразовать число в строку и наоборот?
14. Что такое вызов функции, фактические параметры?
15. Что такое прототип функции, его составляющие и для чего он используется?
16. Какие параметры можно задать для функции **main**?
17. Как лучше задавать название функции?
18. Что такое определение функции, его составляющие?
19. В чем массивы и структуры отличаются?
20. Как выполнить инициализацию структуры при ее описании?
21. Как работать с полями структуры через указатель?
22. Для каких случаев можно использовать динамические структуры?
23. Как можно передать структурную переменную в функцию? Массив структур?
24. Могут ли в структуре использованные другие структурные переменные?
25. Дайте определение понятия файл.
26. Какие основные типы файлов вы знаете?
27. Зачем используются операции открытия и закрытия файлов?
28. Какие уровни работы с файлами вы знаете?
29. Какие функции используются для чтения файлов?
30. Какие библиотеки нужно подключить для низкоуровневого ввода и вывода?
31. Что такое список как структура данных? Дайте определение.
32. Что такое элемент списка? Какие структуры элемента списка вы знаете?

33. Какие особенности списков вы знаете по сравнению с массивами?
34. Что такое однонаправленные и двунаправленные списки? Их преимущества?
35. Как используется структура типа список? Что такое хвост и голова списка

11. Литература.

Основная литература

1. Список литературы, доступные книги и необходимые пособия для ЛР ОП размещены на сайте www.sergebolshakov.ru на страничке “2-й к СУЦ”. Пароль для доступа можно взять у преподавателя или старосты группы.
2. Керниган Б., Ритчи Д. К36 Язык программирования Си.\Пер. с англ., 3-е изд., испр. - СПб.: "Невский Диалект", 2001. - 352 с.: ил.
3. Касюк, С.Т. Курс программирования на языке Си: конспект лекций/С.Т. Касюк. — Челябинск: Издательский центр ЮУрГУ, 2010. — 175 с.
4. MSDN Library for Visual Studio 2005 (Microsoft Document Explorer – входит в состав дистрибутива VS. Нужно обязательно развернуть при установке VS VS или настроить доступ через Интернет.)

Дополнительная литература

5. Общее методическое пособие по курсу для выполнения ЛР и ДЗ (см. на сайте 1-й курс www.sergebolshakov.ru) – см. кнопку в конце каждого раздела сайта!!!
6. Другие методические материалы по дисциплине с сайта www.sergebolshakov.ru.
7. Конспекты лекций по дисциплине “Основы программирования”.
8. Подбельский В.В. Язык Си++: Учебное пособие. – М.: Финансы и статистика, 2003.
9. 5. Подбельский В.В. Стандартный СИ++: Учебное пособие. – М.: Финансы и статистика, 2008.
10. Г. Шилдт “С++ Базовый курс”: Пер. с англ.- М., Издательский дом “Вильямс”, 2011 г. – 672с
11. Фридланд А.Я. Информатика и компьютерные технологии. Основные термины: толковый слов. : 3-е изд. Испр. и доп./ А.Я. Фридланд, Л.С. Хааамирова, И.А. Фридланд. – М.:ООО «Издательство Астрель»: ООО «Издательство АСТ». 2003 – 272с.
12. Г. Шилдт “С++ Руководство для начинающих” : Пер. с англ. - М., Издательский дом “Вильямс”, 2005 г. – 672с
13. Г. Шилдт “Полный справочник по С++”: Пер. с англ.- М., Издательский дом “Вильямс”, 2006 г. – 800с
14. Бьерн Страуструп "Язык программирования С++"- М., Бином, 2010 г.

12. Приложения

12.1. Главный модуль проекта LR10_OP.cpp

LR10_OP.cpp – ГЛАВНЫЙ МОДУЛЬ

```
// Комплексная лабораторная работа
#include "LR10_OP.h"
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <io.h>
#include <time.h>
#include <stdio.h>
#include <conio.h>
#include <string>
#include <string.h>
#include <memory.h>
#include <process.h>
//
#include <errno.h>
////////////////////////////////////
// Задание для работы с файлами (уровень А)
////////////////////////////////////
// Прототипы функций работы с файлами
// МУ п.5
void PrintStudent(Student * pS);
// МУ п.8
void StudPrintMas( Student * pMas , int Razm);
// МУ п.12
void StudFileToMas( const char * FileName , Student * pMas , int * pRazm);
void StudFileToMasWR( const char * FileName , Student ** pMas , int * pRazm);
// МУ п.13
void StudMasToFile( const char * FileName , Student * pMas , int Razm);
void StudMasToFileWR( const char * FileName , Student * pMas , int Razm);
// МУ п.14
void StudPrintFile( const char * FileName );
void StudPrintFileWR( const char * FileName);
// МУ п.15
int StudFileCount( const char * FileName , int * pRazm);
int StudFileCountWR( const char * FileName , int * pRazm);
// МУ п.16
void StudClearFile(const char * FileName);
void StudClearFileWR(const char * FileName);
// МУ п.17
void ClearReadonlyFile(const char * FileName);
// МУ п.18
void SwapStudent ( Student * pA , Student * pB );
// МУ п.20
void StudSortMasNum( Student * pMas , int Razm );
// МУ п.21
void StudSortOklad(const char * FileName );
// МУ п.22
double StudSumm( const char * FileName , double * Sum);
// МУ п.22
void FindStudNum( const char * FileName , Student * pS , int Numb);
// МУ п.22
void FindStudNumWR( const char * FileName , Student * pS , int Numb);
////////////////////////////////////
```

```

// Задание для работы с файлами (уровень В и С)
//////////////////////////////////////////////////
// МУ п.21
void StudSortName(const char * FileName );
// МУ п.21
void StudSortNum(const char * FileName );
// МУ п.23.1/2
void StudAdd(const char * FileName , Student S , TypeAddDel TAD );
// МУ п.23.4/5
void StudDel(const char * FileName , TypeAddDel TAD );
// МУ п.23.3
void StudAddNum(const char * FileName , Student S , int Numb );
// МУ п.23.6
void StudDelNum(const char * FileName , int Numb );
// МУ п.23.8
void ChangeStudNum( const char * FileName , Student pS , int Numb);
// МУ п.23.9
int FindStudName( const char * FileName , Student * pS , const char * FindName );
// МУ п.23.10
void SwapStudFile( const char * FileName , int NumA ,int NumB);
// МУ п.17/23ю11
void StudRemoveFile(const char * FileName);

//////////////////////////////////////////////////
void main(void)
{
system(" chcp 1251 > nul");
////////////////////////////////////////////////// СИ начало
    int pFBin = 0;
//////////////////////////////////////////////////
///////
// МУ п.3 (МУ СИНИЙ)
///////
    Student Stud1 ={ "Лаптева" , 1 , 1000.0};
///////
// МУ п.4 (МУ СИНИЙ)
///////
    Student Stud2;
    Stud2.Num = 2;
    Stud2.Oklad = 2000.00;
    strcpy(Stud2.Name , "Аксенова");
    printf ("Пункт МУ 4:\n");

    printf( "Stud2: Имя = %-15s Номер = %2d Стипендия = %8.2lf \n\n",
        Stud2.Name , Stud2.Num, Stud2.Oklad );
///////
// МУ п.5 (МУ СИНИЙ)
///////
    printf ("Пункт МУ 5:\n");
    PrintStudent(&Stud1) ;
    PrintStudent(&Stud2) ;
///////
// МУ п.6 (МУ ЗЕЛЕНый)
///////
// Student * pStud = new Student;
    Student * pStud = (Student *) malloc (sizeof(Student));
    pStud->Num = 3;
    pStud->Oklad = 3000.00;
    strcpy(pStud->Name , "Большаков");
    printf ("Пункт МУ 6:\n");
    PrintStudent( pStud ) ;

```

```

// delete pStud;
    free( pStud );

pStud= NULL;
////////
// МУ п.7 (МУ СИНИЙ)
////////
Student SMas[] ={{"Первый" , 1 , 1000.0}, {"Второй" , 2 , 2000.0}, {"Третий" , 3 , 3000.0},
    {"Четвертый" , 4 , 4000.0}};
int RazmS = sizeof(SMas)/ sizeof(Student); // Вычисление размерности массива
printf ("Пункт МУ 7:\n");
for (int i = 0 ; i < RazmS ; i++ )
    PrintStudent( &SMas[i] ) ;

////////
// МУ п.8 (МУ СИНИЙ)
////////
printf ("Пункт МУ 8:\n");
StudPrintMas( SMas , RazmS);

////////
// МУ п.9 (МУ ЗЕЛЕНый)
////////
printf ("Пункт МУ 9:\n");
const int Rzm = 6;
Student * pPotok = new Student[Rzm];
// Заполнение
srand( (unsigned)time( NULL ) ); // Настройка датчика случайных чисел
rand(); // Получим первое число для запуска датчика
for (int i=0 ; i < Rzm ; i++ )
{
    char Buf[20];
    char Num[10];

    strcpy(Buf , "Stud № - ");
    pPotok[i].Num = (rand()*9)/ RAND_MAX ; // диапазон 0 - 9
    pPotok[i].Oklad = 1000.0 * 10.0 * rand() / RAND_MAX ;// диапазон 0 - 10000.0
    int n = (rand()*30)/ RAND_MAX ;// диапазон 0 - 30
    strcat(Buf , itoa( n + 1 ,Num, 10 ));
    strcpy( pPotok[i].Name , Buf);
    // PrintStudent( &pPotok[i] );
};
StudPrintMas( pPotok , 6);

////////
// МУ п.10 BDStud.bin
////////
////////
// МУ п.11 (МУ СИНИЙ)
////////
///// Вариант функций ( _open, _close , _write, _read и _eof)
Student ZMas[] ={{"Запись 1" , 0 , 1000.0}, {"Запись 2" , 0 , 2000.0}, {"Запись 3" , 0 , 3000.0},
    {"Запись 4" , 0 , 4000.0}};
int RazmZ = sizeof(ZMas)/ sizeof(Student); // Вычисление размерности массива
printf ("Пункт МУ 11(синий):\n");
system ( "attrib -R BDStud.bin " ); // Снятие атрибута защиты записи, нужно после первого создания
pFBin = _open( "BDStud.bin" , _O_RDWR | _O_BINARY | _O_CREAT | _O_TRUNC);
system("attrib -R BDStud.bin ");
if ( pFBin != -1 )
{
    for ( int i = 0 ; i < RazmZ ; i++ )
    {
        ZMas[i].Num = i + 1;
        _write (pFBin , &ZMas[i] , sizeof(Student));
    }
}

```

```

    };
    _close( pFBin );
};
StudPrintMas( ZMas , RazmZ);

    system("attrib -R BDStud.bin ");
////////
// МУ п.11 Распечатка массива
////// Вариант функций ( _open, _close, _write, _read и _eof)
Student MASStud[100];
Student SRBuf;
printf ("Пункт МУ 11(Печать файла):\n");

pFBin = _open( "BDStud.bin", _O_RDWR | _O_BINARY );
if ( pFBin != -1 )
{
    for ( int i = 0 ; _eof(pFBin) == NULL ; i++ )
    {
        int nByte = _read( pFBin , &MASStud[i] , sizeof(Student));
        PrintStudent( &MASStud[i] );
        RazmZ = i; // для запоминания размера считываемого массива
    }
    _close( pFBin );
};
printf ("Пункт МУ 11(Печать всего массива):\n");
StudPrintMas( MASStud , RazmZ + 1 );
//////////
// ЗАПИСЬ ФАЙЛА
////// Вариант функций ( fopen, fclose, fwrite, fread и feof)
Student ZNMas[] ={{"Запись 1N" , 10 , 1000.0}, {"Запись 2N" , 20 , 2000.0}, {"Запись 3N" , 30 , 3000.0}, {"Запись 4N" , 40 , 4000.0}};
RazmZ = sizeof(ZNMas)/ sizeof(Student); // Вычисление размерности массива
printf ("Пункт МУ 11(WR исходный массив):\n");
StudPrintMas( ZNMas , RazmZ);
FILE * pFN;
pFN = fopen( "BDStudN.bin", "w+b"); // Открытие файла для записи
for ( int i = 0 ; i < RazmZ; i++)
    fwrite( &ZNMas[i], sizeof(Student) , 1, pFN);
fclose(pFN);
//////////
// ЧТЕНИЕ ФАЙЛА
////// Вариант функций ( fopen, fclose, fwrite, fread и feof)
Student MASStudWR[100];
pFN = fopen( "BDStudN.bin" , "r+b"); // Открытие файла для чтения
Student SBuf;
int i = 0;
printf ("Пункт МУ 11(WR чтение и распечатка файла):\n");
while (!feof(pFN) ) // Проверка конца файла
{
    fread( &SBuf, sizeof(Student) , 1, pFN); // чтение одной записи
    if ( !feof(pFN) )
    {
        memcpy( &MASStudWR[i] , &SBuf , sizeof(Student));
        PrintStudent(&SBuf);
    };
    RazmZ = i;
    i++;
};
// Заккрытие файла
fclose( pFN );

```

```

printf ("Пункт МУ 11(WR чтение из массива - функция):\n");
StudPrintMas( MASSStudWR , RazmZ);

//////////
////////
// МУ п.11 Зеленый (цикл для заполнения )
////////
///// Вариант функций ( _open, _close, _write, _read и _eof)
printf ("Пункт МУ 11: _write, _read \n");
system ( "attrib -R BDStud.bin "); // Снятие атрибута защиты записи, нужно после первого создания
// Заполнение
srand( (unsigned)time( NULL ) ); // Настройка датчика случайных чисел
rand(); // Получим первое число для запуска датчика
pFBin = _open( "BDStud.bin", _O_RDWR | _O_BINARY | _O_CREAT | _O_TRUNC);
system("attrib -R BDStud.bin ");

if ( pFBin != -1 )
{
    for ( int i = 0 ; i < 4 ; i++ )
    {
        Student StudFile;
        char Buf[20];
        char Num[10];
        strcpy(Buf , "Запись № - ");
        StudFile.Num = (rand()*9)/ RAND_MAX ;
        StudFile.Oklad = 1000.0 * 10.0 * rand() / RAND_MAX ;
        strcat(Buf , itoa (i + 1 ,Num, 10 ));
        strcpy( StudFile.Name , Buf);

//
        _write (pFBin , &StudFile , sizeof(Student));
        PrintStudent( &StudFile );
    };

//
    _close( pFBin );
    system("attrib -R BDStud.bin ");
};

///// Вариант функций ( fopen, fclose, fwrite, fread и feof)
printf ("Пункт МУ 11 WR: fwrite, fread \n");
// Заполнение
rand(); // Получим первое число для запуска датчика
pFN = fopen( "BDStudN.bin", "w+b"); // Открытие файла для записи
for ( int i = 0 ; i < 5 ; i++ )
{
    Student StudFile;
    char Buf[20];
    char Num[10];
    strcpy(Buf , "Запись № - ");
    StudFile.Num = (rand()*9)/ RAND_MAX ;
    StudFile.Oklad = 1000.0 * 10.0 * rand() / RAND_MAX ;
    strcat(Buf , itoa (i + 1 ,Num, 10 ));
    strcpy( StudFile.Name , Buf);
    fwrite( &StudFile, sizeof(Student) , 1, pFN);
    PrintStudent( &StudFile );
};
fclose(pFN);
//
printf ("Пункт МУ 11:WR - распечатка функцией\n");

StudPrintFileWR( "BDStudN.bin" );
//

```



```

// Специальные функции:
// - из файла в массив ( имя файла , указатель на массив структурных переменных)
///////////////////////////////////////////////////////////////////
//
// МУ п.12 (МУ ЗЕЛЕНый)
//
Student * pStMas; //
Student * pStudMas; //
int StudCount;
//
// Файл в массив
//
// Вариант функций ( fopen, fclose, fwrite, fread и feof)
printf ("Пункт МУ 12:WR (зеленый)\n");
StudFileToMasWR( "BDStudN.bin", &pStudMas, &RazmS);
StudPrintMas( pStudMas, RazmS);
printf ("Пункт МУ 12:\n");
//printf ("Работа с функциями:\n");
// Вариант функций ( _open, _close, _write, _read и _eof)
StudFileToMas( "BDStud.bin", &pStMas, &StudCount );
StudPrintMas( pStMas, StudCount);

strcpy( (char *)((pStMas + 1)->Name), "Новая");
///////////////////////////////////////////////////////////////////
// - из массива в файл ( имя файла , указатель на массив структурных переменных)
//
// МУ п.13
//
Student NZMas[] ={{"Запись 1", 1, 1000.0}, {"Запись 2", 2, 5000.0}, {"Запись 3", 3, 3000.0},
{"Запись 4", 4, 4000.0}};
RazmZ = sizeof(NZMas)/ sizeof(Student); // Вычисление размерности массива
// Изменение второй (индекс = 1) записи
NZMas[1].Num =5;
strcpy( NZMas[1].Name, "Изменение 2-й");
printf ("Пункт МУ 13:WR\n");
printf ("После редактирования массива и запоминания в файле WR\n");
StudMasToFileWR( "BDStudN.bin", NZMas, RazmZ); // Массив записываем в файл!!!!
StudPrintFileWR( "BDStudN.bin");// Файл рапечатываем!!!!
printf ("Пункт МУ 13:\n");
StudMasToFile( "BDStud.bin", NZMas, RazmZ );
printf ("После редактирования массива и запоминания в файле\n");
StudPrintFile( "BDStud.bin" );

//
// МУ п.14
//
// Распечатка файла записей
printf ("Пункт МУ 14:WR\n");
StudPrintFileWR( "BDStudN.bin" );
printf ("Пункт МУ 14:\n");
StudPrintFile( "BDStud.bin" );
// Число записей в БД
//
// МУ п.15 (ЗЕЛЕНый)
//
printf ("Пункт МУ 15:\n");
//
printf("Число записей в БД = %d \n", StudFileCountWR( "BDStudN.bin", &StudCount ));
printf("Число записей в БД = %d \n", StudFileCount( "BDStud.bin", &StudCount ));
// Заполнение файла на основе массива
//
// МУ п.13 (второй вариант на основе инициализированного массива)

```

```

/////////
printf ("Пункт МУ 13:\n");
Student StudMas[] ={{"Иванов" , 3 , 7000.0}, {"Петров" , 2 , 2000.0}, {"Сидоров" , 1 , 3000.0}, {"Печкин" , 4 , 4000.0}};
StudMasToFile( "BDStud.bin" , StudMas , StudCount );

//
/////////
// МУ п.16 (ЗЕЛЕНый)
/////////
printf ("Пункт МУ 16:\n");
// Распечатка файла записей
StudPrintFileWR( "BDStudN.bin" );
StudClearFileWR("BDStudN.bin");
StudPrintFileWR( "BDStudN.bin" );
// Распечатка файла записей
StudPrintFile( "BDStud.bin" );//
StudClearFile("BDStud.bin");
StudPrintFile( "BDStud.bin" );

/////////
// МУ п.17 (ЗЕЛЕНый)
/////////
//
printf ("Пункт МУ 17:\n");
//StudRemoveFile("BDStudD.bin"); // Временно за комментировано
ClearReadOnlyFile("BDStud.bin");
ClearReadOnlyFile("BDStudN.bin");
StudClearFile("BDStud.bin");
StudPrintFile( "BDStud.bin" );
//
printf ("Пункт МУ 17 снова добавим записи из массива (WR):\n");
StudMasToFileWR( "BDStudN.bin" , NZMas, sizeof(StudMas)/sizeof(Student) );
StudPrintFileWR( "BDStudN.bin" );
printf ("Пункт МУ 17 снова добавим записи из массива:\n");
StudMasToFile( "BDStud.bin" , StudMas , sizeof(StudMas)/sizeof(Student) );
StudPrintFile( "BDStud.bin" );

/////////
// МУ п.18 (СИНИЙ)
/////////
printf ("Пункт МУ 18 (Sawp - до):\n");
Student StudA ={"Лаптева" , 1 , 1000.0};
Student StudB ={"Иванова" , 2 , 2000.0};
PrintStudent(&StudA) ;
PrintStudent(&StudB) ;
printf ("Пункт МУ 18 (Sawp - после):\n");
SwapStudent( &StudA , &StudB); // Обмен двух отдельных записей
PrintStudent(&StudA) ;
PrintStudent(&StudB) ;
// В массиве
printf ("Пункт МУ 18 (Sawp - pStMas - до):\n");
StudPrintMas( pStMas , StudCount);
printf ("Пункт МУ 18 (Sawp - pStMas - после):\n");
SwapStudent( &pStMas[0] , &pStMas[2]); // Обмен двух элементов массива
StudPrintMas( pStMas , StudCount);
// Сортировка массива по одному полю записи
/////////
// МУ п.19 (СИНИЙ)
/////////
printf ("Пункт МУ 19 (SORT - pStMas - до):\n");

```

```

StudPrintMas( pStMas , StudCount);
// Сортировка по окладу
for( int k =0 ; k< StudCount - 1 ; k++)
    for ( int i =0 ; i< StudCount - 1 ; i++)
        if ( (pStMas + i)->Oklad > (pStMas + i + 1)->Oklad ) // Возрастание
            SwapStudent( pStMas + i , pStMas + i + 1 );
printf ("Пункт МУ 19 (SORT - pStMas - после Oklad):\n");
StudPrintMas( pStMas , StudCount);
// Сортировка по имени
for( int k =0 ; k< StudCount - 1 ; k++)
    for ( int i =0 ; i< StudCount - 1 ; i++)
        if ( strcmp((pStMas + i)->Name , (pStMas + i + 1)->Name ) > 0 ) //
        Возрастание
            SwapStudent( pStMas + i , pStMas + i + 1 );
printf ("Пункт МУ 19 (SORT - pStMas - после Name):\n");
StudPrintMas( pStMas , StudCount);
// Сортировка по номеру
for( int k =0 ; k< StudCount - 1 ; k++)
    for ( int i =0 ; i< StudCount - 1 ; i++)
        if ( (pStMas + i)->Num > (pStMas + i + 1)->Num ) // Возрастание
            SwapStudent( pStMas + i , pStMas + i + 1 );
printf ("Пункт МУ 19 (SORT - pStMas - после Num):\n");
StudPrintMas( pStMas , StudCount);
/////////
// МУ п.20 (ЗЕЛЕНЬЙ)
/////////
printf ("Пункт МУ 20 (SORT - функция - до):\n");
StudPrintMas( pStMas , StudCount);
StudSortMasNum( pStMas , StudCount );
printf ("Пункт МУ 20 (SORT - функция - после):\n");
StudPrintMas( pStMas , StudCount);
//
/////////
// МУ п.21 (ЗЕЛЕНЬЙ)
/////////
// Сортировка файла (!!!) по одному полю записи
//////////
printf ("Пункт МУ 21 (SORT - с разными функциями):\n");
//////////
printf ("СОТИРОВКА: Работа с функциями Oklad(возрастание):\n");
StudSortOklad("BDStud.bin" );
StudPrintFile( "BDStud.bin" );
//////////
printf ("СОТИРОВКА: Работа с функциями Name:\n");
StudSortName("BDStud.bin" );
StudPrintFile( "BDStud.bin" );
//////////
printf ("СОТИРОВКА: Работа с функциями Num (убывание):\n");
StudSortNum("BDStud.bin");
StudPrintFile( "BDStud.bin" );
//////////
/////////
// МУ п.22 чтение одной записи 2/3-й (0 - 4 ) (СИНИЙ)
/////////
//
printf ("Пункт МУ 22:WR - ЧТЕНИЕ - 2-я\n");
StudPrintFile( "BDStudN.bin" );
int nFind = 2 ; // Номер записи для чтения
int Pos1;
pFN = fopen( "BDStudN.bin" , "rb"); // Открытие файла для чтения
// Перемещение указателя для чтения

```

```

if (filelength(pFN->_file)/sizeof(Student) >= nFind ) // контроль номера для чтения
{
    Pos1 = fseek( pFN, (nFind - 1)*sizeof(Student), SEEK_SET); // Перемещение указателя на
нужную запись для чтения
    fread( &SBuf, sizeof(Student) , 1, pFN); // чтение одной записи
    PrintStudent(&SBuf);
}
else
    printf( "Ошибка номера записи- %d !\n" , nFind ); // Ошибка номера
// Закрытие файла
    fclose( pFN );
//
printf ( "Пункт МУ 22:\n");
// Чтение 3-й
nFind = 3;
StudPrintFile( "BDStud.bin" );
Student SFind ;
pFBin = _open( "BDStud.bin" , _S_IREAD |_O_BINARY );
if ( pFBin != -1 )
{
    long posF = _lseek( pFBin, sizeof(Student) * (nFind - 1), SEEK_SET ); // Установить новую текущую
позицию
    int nByte = _read( pFBin , &SFind , sizeof(Student));
};
printf ("Чтение из программы (номер = 3):\n");
PrintStudent( &SFind );
// Закрытие файла
    _close( pFBin );
/////////
/////////
// МУ п.22.1 чтение из функции по номеру (ЗЕЛЕНЫЙ)
/////////
///// - Прочитать запись по номеру
printf ("Чтение из функции (номер = 2):\n");
FindStudNumWR( "BDStudN.bin" , &SFind , 2);
PrintStudent( &SFind );
FindStudNumWR( "BDStudN.bin" , &SFind , 20); // Ошибка номера
printf ("Чтение из функции (номер = 1):\n");
FindStudNum( "BDStud.bin" , &SFind , 1);
PrintStudent( &SFind );
/////////
// МУ п.22 интегрированные расчеты в программе (СИНИЙ)
/////////
///// Вариант функций ( fopen, fclose , fwrite, fread и feof)
FILE * pFS;
double SumOk = 0;
Student SFromFile;
pFS = fopen( "BDStudN.bin" , "r+b"); // Открытие файла для чтения
if ( pFS != NULL )
{
    while (!feof(pFS) ) // Проверка конца файла
    {
        fread( &SFromFile, sizeof(Student) , 1, pFS); // чтение одной записи
        if ( !feof(pFS) ) {
            SumOk = SumOk + SFromFile.Oklad;
        };
    };
};
printf ( "Фонд зарплаты в вычисленный цикл по файлу WR %8.2lf \n" , SumOk );

// Закрытие файла

```

```

        fclose( pFS );
//////////

double SumStip = 0;
Student SIntg;
printf ( "Пункт МУ 22 (Интегральные расчеты):\n");
pFBin = _open( "BDStud.bin" , _S_IREAD | _O_BINARY );
if ( pFBin != -1 )
{
    for ( int i = 0 ; _eof(pFBin) == NULL ; i++ )
    {
        int nByte = _read( pFBin , &SIntg , sizeof(Student));
        SumStip = SumStip + SIntg.Oklad;
    }
    _close( pFBin );
};
printf ( "Фонд зарплаты в вычисленный цикле = по файлу  %8.2lf \n" , SumStip );

//////////
// МУ п.23
//////////
double SumOklad;
printf ( "Фонд зарплаты из функции =  %8.2lf \n" , StudSumm( "BDStud.bin" , &SumOklad));
//////////
//////////
// Очистка файла и его удаление
ClearReadOnlyFile("BDStud.bin");
// StudClearFile("BDStud.bin");
// StudRemoveFile("BDStud.bin");
///// УРОБЕНЬ С
//////////
// МУ п.23
//////////
// МУ п.23.1
//////////
// Добавить запись ( начало )
printf ( "Добавление (начало):\n");
Student SFirst = {"Первый" , 22 , 3400.00};
StudAdd("BDStud.bin", SFirst , First );
StudPrintFile( "BDStud.bin" );
//////////
// МУ п.23.2
//////////
// - Добавить запись ( конец)
printf ( "Добавление (конец):\n");
Student SLast = {"Последний" , 22 , 3400.00};
StudAdd("BDStud.bin", SLast );
StudPrintFile( "BDStud.bin" );
//////////
//////////
// МУ п.23.3
//////////
// Добавить запись (номер)
//////////
printf ( "Добавление (номер):\n");
Student SNumb = {"По номеру - 2" , 33 , 5500.00};
StudAddNum("BDStud.bin", SNumb, 3);
StudPrintFile( "BDStud.bin" );
///// - Удалить запись по номеру (? другие режимы)
//////////

```

```

////////
// МУ п.23.4
////////printf ("Удаление (Начало):\n");
StudDel("BDStud.bin", First );
StudPrintFile( "BDStud.bin" );
////////
////////
// МУ п.23.5
////////
printf ("Удаление (конец):\n");
StudDel("BDStud.bin", Last );
//StudDel("BDStud.bin" );
StudPrintFile( "BDStud.bin" );
////////
////////
// МУ п.23.6
////////
printf ("Удаление (номер):\n");
StudDelNum("BDStud.bin", 4 );
StudPrintFile( "BDStud.bin" );
////////
////////
// МУ п.23.7
////////
///// - Прочитать запись по номеру
printf ("Чтение (номер):\n");
Student * pFind = new Student;
FindStudNum( "BDStud.bin" , pFind , 2);
PrintStudent( pFind );
////////
// - Изменить запись по номеру
////////
// МУ п.23.8
////////
printf ("Изменение записи (номер):\n");
Student SChang ={ "Замена" , 11, 111.00};
ChangeStudNum( "BDStud.bin" , SChang , 2);
StudPrintFile( "BDStud.bin" );
////////
// - Найти запись по признаку равенства поля (имени - первую в файле) по любому полю
////////
// МУ п.23.9
////////
int n;
printf ("Поиск по имени:\n");
if ( ( n =FindStudName( "BDStud.bin" , pFind , "Замена" )) != -1)
{ printf ("Найдена по номеру = %d \n", n );
printf ("Найдена запись: ");
PrintStudent( pFind );
}
else
printf ("Не найдена запись! ");
////////
////////
// МУ п.23.10
////////
// Swap замена записей в файле БД
printf ("Замена SWAP в файле по номеру: 1<->3\n");
SwapStudFile( "BDStud.bin" , 1 ,3 );
StudPrintFile( "BDStud.bin" );
////////

```

```
delete [] pStMas;  
////////////////////////////////////  
system("PAUSE");  
////////////////////////////////////  
  
}
```

12.2. Заголовочный файл проекта LR10_OP.h

LR10_OP.h– Заголовочный МОДУЛЬ

```
// Заголовочный модуль ЛР 10
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <io.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <memory.h>
#include <malloc.h>
#include <process.h>
#include <errno.h>
//
//using namespace std;
//////////
//////////
// МУ п.2
//////////
struct Student {
public:
    char Name[20];
    int Num;
    double Oklad;
};
// Константы для управления функциями
enum TypeAddDel{ First = 1 , Last = 2 , Default = Last};
// Прототипы
//////////
//////////
// МУ п.5
// Распечатка отдельной структурной переменной студента
//////////
void PrintStudent(Student * pS)
{
    printf( "Запись: Имя = %-15s Номер = %2d Стипендия = %8.2lf \n",
        pS->Name , pS->Num, pS->Oklad );
};
//////////
// МУ п.8
void StudPrintMas( Student * pMas , int Razm)
{
    for (int i = 0 ; i < Razm ; i++ )
    {
        PrintStudent(pMas + i); // Используется готовая функция
    };
};
//////////
// Выборка из файла в динамический массив
// МУ п.12
//////////
void StudFileToMas( const char * FileName , Student ** pMas , int * pRazm)
{
    long posF;
    int pFBin = _open( FileName, _S_IREAD | _O_BINARY );
    if ( pFBin == -1 ) { *pRazm = NULL; return;};
    posF = _lseek( pFBin, 0, SEEK_END ); // Установить новую текущую позицию
    *pRazm = posF / sizeof(Student);
    //*pMas = new Student[ *pRazm];
```



```

*pMas = (Student *) calloc( *pRazm , sizeof(Student) );
posF = _lseek( pFBin, 0, SEEK_SET ); // На начало
// Буфер для чтения
Student SBuf;
    for (int i = 0 ; i < *pRazm ; i++ )
    {
        int nByte = _read( pFBin , &SBuf , sizeof(Student));
        if ( nByte == NULL) break;
        memcpy( *pMas + i , &SBuf , sizeof(Student));
    };
// Заккрытие файла
    _close( pFBin );
};
//
//
//
//Перезапись массива структур в файл (Файл очищается!)
// МУ п.13
//
//
void StudMasToFile( const char * FileName , Student * pMas , int Razm)
{
    char Comand[40];
    // Снятие защиты с файла - отключение Readonly
    strcpy (Comand , "attrib -R ");
    strcat (Comand , FileName);
    system( Comand );
    int pFBin = _open( FileName , _O_RDWR | _O_BINARY | _O_CREAT | _O_TRUNC);
    if ( pFBin == -1 ) return;
    // цикл записи
        for ( int i = 0 ; i < Razm ; i++ )
        {
            _write (pFBin , pMas + i , sizeof(Student));
        };

    _close( pFBin );
//
};
//
//
// Печать файла БД Студентов
// МУ п.14
//
//
void StudPrintFile( const char * FileName )
{
    long posF;
    int pFBin = _open( FileName, _S_IREAD | _O_BINARY );
    if ( pFBin != -1 )
    {
        posF = _lseek( pFBin, 0, SEEK_END ); // Установить новую текущую позицию
        int Razm = posF / sizeof(Student);
        if ( Razm == 0 ) { _close( pFBin ); printf( "Записей в файле нет! \n" ) ; return;};
        posF = _lseek( pFBin, 0, SEEK_SET ); // На начало
        // Буфер для чтения
        Student SBuf;
            for (int i = 0 ; i < Razm ; i++ )
            {
                int nByte = _read( pFBin , &SBuf , sizeof(Student));
                if ( nByte == NULL) break;
                // вывод записи
                printf( "%d - " , i + 1);
                PrintStudent(&SBuf);
            };
    };
};

```

```

// Закрытие файла
    _close( pFBin );
};

/////////////////////////////////
// Получить число записей в БД
// МУ п.15
int StudFileCount( const char * FileName , int * pRazm)
{
    long posF;
    int pFBin = _open( FileName, _S_IREAD | _O_BINARY );
    if ( pFBin == -1 ) { *pRazm = NULL; return NULL;};
    posF = _lseek( pFBin, 0, SEEK_END ); // Установить новую текущую позицию
    *pRazm = posF / sizeof(Student);
    _close( pFBin );
    return *pRazm;
};
// МУ п.15
int StudFileCountWR( const char * FileName , int * pRazm)
{
    FILE * pF;
    pF = fopen( FileName, "r+b");
    *pRazm = ((int)_filelength ( pF->_file))/sizeof(Student);
    // Закрытие файла
    fclose( pF );
    return *pRazm;
};
/////////////////////////////////
// Очистка файла студентов
// МУ п.16
/////////////////////////////////
void StudClearFile(const char * FileName)
{
    char Comand[40];
    // Временное снятие защиты с файла - отключение Readonly
    strcpy (Comand , "attrib -R ");
    strcat (Comand , FileName);
    system( Comand );
    int pFBin = _open( FileName , _O_RDWR | _O_BINARY | _O_CREAT | _O_TRUNC);
    if ( pFBin == -1 ) return;
    _close( pFBin );
};
/////////////////////////////////
void StudClearFileWR(const char * FileName)
{
    FILE * pF;
    pF = fopen( FileName, "w+");
    // Закрытие файла
    fclose( pF );
};
/////////////////////////////////
// Сброс флага файла
// МУ п.17
/////////////////////////////////
void ClearReadonlyFile(const char * FileName)
{
    char Comand[40];
    // Снятие защиты с файла - отключение Readonly
    strcpy (Comand , "attrib -R ");
    strcat (Comand , FileName);

```

```

system( Comand );
};
//////////
// МУ п.18
//////////
// обмен на основе адресов структурных переменных (возможен только без динамики!!)
void SwapStudent ( Student * pA , Student * pB )
{
Student Temp;
/* можно и так
Temp = *pA;
*pA = *pB;
*pB = Temp;
*/

memcpy( &Temp , pA, sizeof(Student) );
memcpy( pA , pB, sizeof(Student) );
memcpy( pB , &Temp, sizeof(Student) );
};
//////////
// МУ п.20
//////////
void StudSortMasNum( Student * pStMas , int StudCount )
{
for( int k =0 ; k< StudCount - 1 ; k++)
for ( int i =0 ; i< StudCount - 1 ; i++)
if ( (pStMas + i)->Num > (pStMas + i + 1)->Num ) // Возрастание
SwapStudent( pStMas + i , pStMas + i + 1 );
};
//////////
// Сортировка массива по окладу по убыванию
// МУ п.21
//////////
void StudSortOklad(const char * FileName )
{
// Чтение из файла в массив
Student * pStudMas;
int Razm;
StudFileToMas( FileName , &pStudMas , &Razm );
// Сортировка массива по окладу по убыванию
for (int i = 0 ; i < Razm - 1 ; i++)
for (int k = 0 ; k < Razm - 1 ; k++)
// Сортировка разные режимы!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
if ( ((Student *) (pStudMas + k))->Oklad > ((Student *) (pStudMas + k + 1))->Oklad ) //
Убывание курс
{ SwapStudent ( pStudMas + k , pStudMas + k + 1 ); };
//
StudMasToFile( FileName , pStudMas , Razm );
};
//////////
// МУ п.22
//////////
double StudSumm( const char * FileName , double * Sum)
{
Student * pStudMas;
int Razm;
*Sum = 0.0;
StudFileToMas( FileName , &pStudMas , &Razm );
for ( int i = 0 ; i < Razm ; i++)
*Sum+= (pStudMas + i)->Oklad;
delete [] pStudMas;

```

```

return *Sum;
};
//////////
// Сортировка по имени (дополнительные примеры) МУ п.21
//////////
void StudSortName(const char * FileName )
{
// Чтение из файла в массив
Student * pStudMas;
int Razm;
StudFileToMas( FileName , &pStudMas , &Razm );
//
//////////
for (int i = 0 ; i < Razm - 1 ; i++)
    for (int k = 0 ; k < Razm - 1 ; k++)
// Сортировка по имени
    if ( strcmp(((Student *) (pStudMas + k))->Name , ((Student *) (pStudMas + k + 1))->Name) > 0 )
// Убывание имя
        { SwapStudent ( pStudMas + k , pStudMas + k + 1 ); };
//
StudMasToFile( FileName , pStudMas , Razm );
};
//////////
// Сортировка по номеру (дополнительные примеры) МУ п.21
//////////
void StudSortNum(const char * FileName )
{
// Чтение из файла в массив
Student * pStudMas;
int Razm;
StudFileToMas( FileName , &pStudMas , &Razm );
StudSortMasNum( pStudMas , Razm ); // Использование сортировки в массиве!!!
StudMasToFile( FileName , pStudMas , Razm );
};
//////////
//////////
//////////
// Удаление из файла начало или конец файла
//////////
// МУ п.23.1/2
//////////
void StudAdd(const char * FileName , Student S , TypeAddDel TAD = Default ){
//
if ( TAD == Last)
{
ClearReadOnlyFile(FileName);
int pFBin = _open( FileName, _O_RDWR | _O_BINARY | _O_APPEND );
long posF = _lseek( pFBin, 0, SEEK_END ); // Установить новую текущую позицию
        _write (pFBin , &S, sizeof(Student));
// Заккрытие файла
        _close( pFBin );
        return;};
if( TAD == First )
{
Student * pStudMas;
int Razm;
StudFileToMas( FileName , &pStudMas , &Razm );
Student * pTemp = (Student *) new Student [ Razm + 1];
        memcpy( pTemp , &S, sizeof(Student) );
for ( int i = 1 ; i <= Razm ; i++)
        memcpy( pTemp + i , pStudMas + (i - 1), sizeof(Student) );
}
}

```

```

//
StudMasToFile( FileName , pTemp , Razm + 1 );
delete [] pTemp;
delete [] pStudMas;
return;};
return;
};
//////////
// Добавления по номеру в файл
// МУ п.23.3
//////////
void StudAddNum(const char * FileName , Student S , int Numb )
{
Student * pStudMas;
int Razm;
if ( Numb < 0) return;
// Формирование массива
StudFileToMas( FileName , &pStudMas , &Razm );
if (Numb >= Razm )
{
StudAdd(FileName, S , Last );
delete [] pStudMas;
return;
};
//
Student * pTemp = (Student *) new Student [ Razm + 1];

for ( int i = 0 , k=0 ; i <= Razm ; i++, k++)
{
if ( ( i - 1) == Numb)
{
memcpy( pTemp + i , &S, sizeof(Student) );
k--;
}
else
{
memcpy( pTemp + i , pStudMas + k, sizeof(Student) );
};
};
// Запоминание в файл нового
StudMasToFile( FileName , pTemp , Razm + 1 );
delete [] pTemp;
delete [] pStudMas;
};
//////////
// Удаление первого или последнего из файла
// МУ п.23.4/5
//////////
void StudDel(const char * FileName , TypeAddDel TAD = Last )
{
Student * pStudMas;
int Razm;
if ( TAD == Last)
{
StudFileToMas( FileName , &pStudMas , &Razm );
StudMasToFile( FileName , pStudMas , Razm - 1 );
delete [] pStudMas;
return;
};
if ( TAD == First )
{

```

```

StudFileToMas( FileName , &pStudMas , &Razm );
StudMasToFile( FileName , pStudMas + 1 , Razm - 1 );
delete [] pStudMas;
}

};
////////////////////////////////////
// Удаление по номеру из файла
// МУ п.23.6
////////////////////////////////////
void StudDelNum(const char * FileName , int Numb )
{
Student * pStudMas;
int Razm;
StudFileToMas( FileName , &pStudMas , &Razm );
if ( Numb < 0 || Numb > Razm ) return;
if ( Numb == 0 ) {
StudDel( FileName , First );
delete [] pStudMas;
return;};
if ( Numb == Razm -1 ) {
StudDel( FileName , Last );
delete [] pStudMas;
return;
};
// Удаление из середины
Student * pTemp = (Student *) new Student [ Razm - 1];

for (int i =0, k =0 ; i < Razm ; i++ , k++)
{
if ( i == Numb )
k--;
else
memcpy( pTemp + k , pStudMas + i , sizeof(Student) );
};
StudMasToFile( FileName , pTemp , Razm - 1 );
delete [] pTemp;
delete [] pStudMas;
};
//
////////////////////////////////////
// Поиск и выборка одной записи по номеру
// МУ п.23.7
////////////////////////////////////
void FindStudNum( const char * FileName , Student * pS , int Numb){
Student * pStudMas;
int Razm;
StudFileToMas( FileName , &pStudMas , &Razm );
if ( Numb < 0 || Numb >= Razm )
{
pS->Name[0] = '\0';
pS->Num = NULL;
pS->Oklad = NULL;
}
else
memcpy( pS , pStudMas + Numb , sizeof(Student) );
delete [] pStudMas;
};
////////////////////////////////////
// МУ п.23.8
////////////////////////////////////

```

```

void ChangeStudNum( const char * FileName , Student pS , int Numb){
Student * pStudMas;
int Razm;
StudFileToMas( FileName , &pStudMas , &Razm );
if ( Numb < 0 || Numb >= Razm )
{
    delete [] pStudMas;
    return;
}
else
    memcpy( pStudMas + Numb , &pS, sizeof(Student) );
StudMasToFile( FileName , pStudMas , Razm );
delete [] pStudMas;
};
//////////
// Поиск по имени содержимого записей в файле в файле (первая запись)
// МУ п.23.9
//////////
int FindStudName( const char * FileName , Student * pS , const char * FindName )
{
Student * pStudMas;
int Razm;
int i ;
StudFileToMas( FileName , &pStudMas , &Razm );
for ( i = 0 ; i < Razm ; i++)
    if ( strcmp ( (pStudMas + i)->Name , FindName ) == 0) break;
    if ( i == Razm )
    {
        // Не найдено
        pS->Name[0] = '\0';
        pS->Num = NULL;
        pS->Oklad = NULL;
        delete [] pStudMas;
        return -1 ;
    }
    // Найдено
    else
        memcpy( pS , pStudMas + i , sizeof(Student) );

        delete [] pStudMas;
        return i ;
};
//////////
// Замена по номерам записей в файле
//////////
// МУ п.23.10
//////////
void SwapStudFile( const char * FileName , int NumA ,int NumB)
{
Student * pStudMas;
int Razm;
int i ;
StudFileToMas( FileName , &pStudMas , &Razm );
if ( NumA < 0 || NumA >= Razm || NumB < 0 || NumB >= Razm )
{ }
else
{SwapStudent ( pStudMas + NumA , pStudMas + NumB );
StudMasToFile( FileName , pStudMas , Razm );};
//
delete [] pStudMas;
};

```

```

//////////
//////////
// МУ п.17/23.11
//////////
void StudRemoveFile(const char * FileName)
{
    char Comand[40];
    // Снятие защиты с файла - отключение Readonly
    strcpy (Comand , "attrib -R ");
    strcat (Comand , FileName);
    system( Comand );
    strcpy (Comand , "attrib -A ");
    strcat (Comand , FileName);
    system( Comand );
    // Удаление файла все дескрипторы файла должны быть закрыты
    strcpy (Comand , "del ");
    strcat (Comand , FileName);
    system( Comand );
};
//////////
//////////
// Выборка из файла в динамический массив WR
void StudFileToMasWR( const char * FileName , Student ** pMas , int * pRazm)
{
    FILE * pF;
    pF = fopen( FileName , "rb"); // Открытие файла для чтения
    *pRazm = _filelength(pF ->_file) /sizeof(Student);
    Student SBuf;
    *pMas = (Student * ) calloc( *pRazm , sizeof(Student));
    int i = 0;
    while (!feof(pF) ) // Проверка конца файла
    {
        fread( &SBuf, sizeof(Student) , 1, pF); // чтение одной записи
        if ( !feof(pF) )
        { memcpy( *pMas + i , &SBuf , sizeof(Student)); i++; };
    };
    // Закрытие файла
    fclose( pF );
};
//////////
// МУ ЛР7 - 4.8
// Печать файла для структуры Student
void StudPrintFileWR( const char * FileName)
{
    FILE * pF;
    pF = fopen( FileName , "r+b"); // Открытие файла для чтения
    Student SBuf;
    int i = 0;
    while (!feof(pF) ) // Проверка конца файла
    {
        fread( &SBuf, sizeof(Student) , 1, pF); // чтение одной записи
        if ( !feof(pF) ) {
            i++;
            printf("%d. " , i);
            PrintStudent(&SBuf);
        };
    };
    // Закрытие файла
    fclose( pF );
};
//////////

```



```

//Перезапись массива структур в файл (Файл очищается!) WR
void StudMasToFileWR( const char * FileName , Student * pMas , int Razm)
{
    FILE * pF;
    pF = fopen( FileName, "w+b"); // Открытие файла для записи
    for ( int i = 0 ; i < Razm; i++)
        fwrite( pMas + i, sizeof(Student) , 1, pF);
    fclose(pF);
};

//////////
//////////
void FindStudNumWR( const char * FileName , Student * pS , int Numb){
    FILE * pF;
    pF = fopen( FileName , "r+b"); // Открытие файла для чтения
    int Razm = _filelength(pF ->_file) /sizeof(Student);
    if ( (Numb > 0 ) && ( Numb <= Razm ) )
    {
        long Pos1 = fseek( pF, (Numb - 1)*sizeof(Student), SEEK_SET); // Перемещение указателя на нужную
        запись для чтения
        fread( pS, sizeof(Student) , 1, pF); // чтение одной записи (2-й)
        // PrintStudent(pS);
    }
    else
    {
        printf( "Ошибка номера записи- %d !\n" , Numb ); // Ошибка номера
    };

    // Закрытие файла
    fclose( pF );
};

```

12.3. Результаты работы главного модуля примеров

Пункт МУ 4:

Stud2: Имя = Аксенова Номер = 2 Стипендия = 2000.00

Пункт МУ 5:

Запись: Имя = Лаптева Номер = 1 Стипендия = 1000.00

Запись: Имя = Аксенова Номер = 2 Стипендия = 2000.00

Пункт МУ 6:

Запись: Имя = Большаков Номер = 3 Стипендия = 3000.00

Пункт МУ 7:

Запись: Имя = Первый Номер = 1 Стипендия = 1000.00

Запись: Имя = Второй Номер = 2 Стипендия = 2000.00

Запись: Имя = Третий Номер = 3 Стипендия = 3000.00

Запись: Имя = Четвертый Номер = 4 Стипендия = 4000.00

Пункт МУ 8:

Запись: Имя = Первый Номер = 1 Стипендия = 1000.00

Запись: Имя = Второй Номер = 2 Стипендия = 2000.00

Запись: Имя = Третий Номер = 3 Стипендия = 3000.00

Запись: Имя = Четвертый Номер = 4 Стипендия = 4000.00

Пункт МУ 9:

Запись: Имя = Stud № - 26 Номер = 8 Стипендия = 5765.86

Запись: Имя = Stud № - 5 Номер = 2 Стипендия = 1937.93

Запись: Имя = Stud № - 11 Номер = 5 Стипендия = 5739.31

Запись: Имя = Stud № - 13 Номер = 5 Стипендия = 5062.72

Запись: Имя = Stud № - 26 Номер = 1 Стипендия = 4487.14

Запись: Имя = Stud № - 21 Номер = 4 Стипендия = 2313.30

Пункт МУ 11(синий) :

Запись: Имя = Запись 1 Номер = 1 Стипендия = 1000.00

Запись: Имя = Запись 2 Номер = 2 Стипендия = 2000.00

Запись: Имя = Запись 3 Номер = 3 Стипендия = 3000.00

Запись: Имя = Запись 4 Номер = 4 Стипендия = 4000.00

Пункт МУ 11(Печать файла) :

Запись: Имя = Запись 1 Номер = 1 Стипендия = 1000.00

Запись: Имя = Запись 2 Номер = 2 Стипендия = 2000.00

Запись: Имя = Запись 3 Номер = 3 Стипендия = 3000.00

Запись: Имя = Запись 4 Номер = 4 Стипендия = 4000.00

Пункт МУ 11(Печать всего массива) :

Запись: Имя = Запись 1 Номер = 1 Стипендия = 1000.00

Запись: Имя = Запись 2 Номер = 2 Стипендия = 2000.00

Запись: Имя = Запись 3 Номер = 3 Стипендия = 3000.00

Запись: Имя = Запись 4 Номер = 4 Стипендия = 4000.00

Пункт МУ 11(WR исходный массив) :

Запись: Имя = Запись 1N Номер = 10 Стипендия = 1000.00

Запись: Имя = Запись 2N Номер = 20 Стипендия = 2000.00

Запись: Имя = Запись 3N Номер = 30 Стипендия = 3000.00

Запись: Имя = Запись 4N Номер = 40 Стипендия = 4000.00

Пункт МУ 11(WR чтение и распечатка файла) :

Запись: Имя = Запись 1N Номер = 10 Стипендия = 1000.00

Запись: Имя = Запись 2N Номер = 20 Стипендия = 2000.00

Запись: Имя = Запись 3N Номер = 30 Стипендия = 3000.00

Запись: Имя = Запись 4N Номер = 40 Стипендия = 4000.00

Пункт МУ 11(WR чтение из массива - функция) :

Запись: Имя = Запись 1N Номер = 10 Стипендия = 1000.00

Запись: Имя = Запись 2N Номер = 20 Стипендия = 2000.00

Запись: Имя = Запись 3N Номер = 30 Стипендия = 3000.00

Запись: Имя = Запись 4N Номер = 40 Стипендия = 4000.00

Пункт МУ 11: _write, _read

Запись: Имя = Запись № - 1 Номер = 8 Стипендия = 5765.86

Запись: Имя = Запись № - 2 Номер = 7 Стипендия = 2980.74

Запись: Имя = Запись № - 3 Номер = 1 Стипендия = 1514.94

Запись: Имя = Запись № - 4 Номер = 5 Стипендия = 5739.31

Пункт МУ 11 WR: fwrite, fread

Запись: Имя = Запись № - 1 Номер = 5 Стипендия = 5062.72

Запись: Имя = Запись № - 2 Номер = 3 Стипендия = 2072.82
 Запись: Имя = Запись № - 3 Номер = 4 Стипендия = 8489.64
 Запись: Имя = Запись № - 4 Номер = 4 Стипендия = 2313.30
 Запись: Имя = Запись № - 5 Номер = 6 Стипендия = 6584.98

Пункт МУ 11:WR - распечатка функцией

1. Запись: Имя = Запись № - 1 Номер = 5 Стипендия = 5062.72
 2. Запись: Имя = Запись № - 2 Номер = 3 Стипендия = 2072.82
 3. Запись: Имя = Запись № - 3 Номер = 4 Стипендия = 8489.64
 4. Запись: Имя = Запись № - 4 Номер = 4 Стипендия = 2313.30
 5. Запись: Имя = Запись № - 5 Номер = 6 Стипендия = 6584.98

Пункт МУ 12:WR (зеленый)

Запись: Имя = Запись № - 1 Номер = 5 Стипендия = 5062.72
 Запись: Имя = Запись № - 2 Номер = 3 Стипендия = 2072.82
 Запись: Имя = Запись № - 3 Номер = 4 Стипендия = 8489.64
 Запись: Имя = Запись № - 4 Номер = 4 Стипендия = 2313.30
 Запись: Имя = Запись № - 5 Номер = 6 Стипендия = 6584.98

Пункт МУ 12:

Запись: Имя = Запись № - 1 Номер = 8 Стипендия = 5765.86
 Запись: Имя = Запись № - 2 Номер = 7 Стипендия = 2980.74
 Запись: Имя = Запись № - 3 Номер = 1 Стипендия = 1514.94
 Запись: Имя = Запись № - 4 Номер = 5 Стипендия = 5739.31

Пункт МУ 13:WR

После редактирования массива и запоминания в файле WR

1. Запись: Имя = Запись 1 Номер = 1 Стипендия = 1000.00
 2. Запись: Имя = Изменение 2-й Номер = 5 Стипендия = 5000.00
 3. Запись: Имя = Запись 3 Номер = 3 Стипендия = 3000.00
 4. Запись: Имя = Запись 4 Номер = 4 Стипендия = 4000.00

Пункт МУ 13:

После редактирования массива и запоминания в файле

1 - Запись: Имя = Запись 1 Номер = 1 Стипендия = 1000.00
 2 - Запись: Имя = Изменение 2-й Номер = 5 Стипендия = 5000.00
 3 - Запись: Имя = Запись 3 Номер = 3 Стипендия = 3000.00
 4 - Запись: Имя = Запись 4 Номер = 4 Стипендия = 4000.00

Пункт МУ 14:WR

1. Запись: Имя = Запись 1 Номер = 1 Стипендия = 1000.00
 2. Запись: Имя = Изменение 2-й Номер = 5 Стипендия = 5000.00
 3. Запись: Имя = Запись 3 Номер = 3 Стипендия = 3000.00
 4. Запись: Имя = Запись 4 Номер = 4 Стипендия = 4000.00

Пункт МУ 14:

1 - Запись: Имя = Запись 1 Номер = 1 Стипендия = 1000.00
 2 - Запись: Имя = Изменение 2-й Номер = 5 Стипендия = 5000.00
 3 - Запись: Имя = Запись 3 Номер = 3 Стипендия = 3000.00
 4 - Запись: Имя = Запись 4 Номер = 4 Стипендия = 4000.00

Пункт МУ 15:

Число записей в БД = 4

Число записей в БД = 4

Пункт МУ 13:

Пункт МУ 16:

1. Запись: Имя = Запись 1 Номер = 1 Стипендия = 1000.00
 2. Запись: Имя = Изменение 2-й Номер = 5 Стипендия = 5000.00
 3. Запись: Имя = Запись 3 Номер = 3 Стипендия = 3000.00
 4. Запись: Имя = Запись 4 Номер = 4 Стипендия = 4000.00
 1 - Запись: Имя = Иванов Номер = 3 Стипендия = 7000.00
 2 - Запись: Имя = Петров Номер = 2 Стипендия = 2000.00
 3 - Запись: Имя = Сидоров Номер = 1 Стипендия = 3000.00
 4 - Запись: Имя = Печкин Номер = 4 Стипендия = 4000.00

Записей в файле нет!

Пункт МУ 17:

Записей в файле нет!

Пункт МУ 17 снова добавим записи из массива (WR) :

1. Запись: Имя = Запись 1 Номер = 1 Стипендия = 1000.00
 2. Запись: Имя = Изменение 2-й Номер = 5 Стипендия = 5000.00
 3. Запись: Имя = Запись 3 Номер = 3 Стипендия = 3000.00
 4. Запись: Имя = Запись 4 Номер = 4 Стипендия = 4000.00

Пункт МУ 17 снова добавим записи из массива:

| | | |
|---------------------------|-----------|---------------------|
| 1 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| 2 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 3 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |
| 4 - Запись: Имя = Печкин | Номер = 4 | Стипендия = 4000.00 |

Пункт МУ 18 (Sawp - до):

| | | |
|-----------------------|-----------|---------------------|
| Запись: Имя = Лаптева | Номер = 1 | Стипендия = 1000.00 |
| Запись: Имя = Иванова | Номер = 2 | Стипендия = 2000.00 |

Пункт МУ 18 (Sawp - после):

| | | |
|-----------------------|-----------|---------------------|
| Запись: Имя = Иванова | Номер = 2 | Стипендия = 2000.00 |
| Запись: Имя = Лаптева | Номер = 1 | Стипендия = 1000.00 |

Пункт МУ 18 (Sawp - pStMas - до):

| | | |
|----------------------------|-----------|---------------------|
| Запись: Имя = Запись № - 1 | Номер = 8 | Стипендия = 5765.86 |
| Запись: Имя = Новая | Номер = 7 | Стипендия = 2980.74 |
| Запись: Имя = Запись № - 3 | Номер = 1 | Стипендия = 1514.94 |
| Запись: Имя = Запись № - 4 | Номер = 5 | Стипендия = 5739.31 |

Пункт МУ 18 (Sawp - pStMas - после):

| | | |
|----------------------------|-----------|---------------------|
| Запись: Имя = Запись № - 3 | Номер = 1 | Стипендия = 1514.94 |
| Запись: Имя = Новая | Номер = 7 | Стипендия = 2980.74 |
| Запись: Имя = Запись № - 1 | Номер = 8 | Стипендия = 5765.86 |
| Запись: Имя = Запись № - 4 | Номер = 5 | Стипендия = 5739.31 |

Пункт МУ 19 (SORT - pStMas - до):

| | | |
|----------------------------|-----------|---------------------|
| Запись: Имя = Запись № - 3 | Номер = 1 | Стипендия = 1514.94 |
| Запись: Имя = Новая | Номер = 7 | Стипендия = 2980.74 |
| Запись: Имя = Запись № - 1 | Номер = 8 | Стипендия = 5765.86 |
| Запись: Имя = Запись № - 4 | Номер = 5 | Стипендия = 5739.31 |

Пункт МУ 19 (SORT - pStMas - после Oklad):

| | | |
|----------------------------|-----------|---------------------|
| Запись: Имя = Запись № - 3 | Номер = 1 | Стипендия = 1514.94 |
| Запись: Имя = Новая | Номер = 7 | Стипендия = 2980.74 |
| Запись: Имя = Запись № - 4 | Номер = 5 | Стипендия = 5739.31 |
| Запись: Имя = Запись № - 1 | Номер = 8 | Стипендия = 5765.86 |

Пункт МУ 19 (SORT - pStMas - после Name):

| | | |
|----------------------------|-----------|---------------------|
| Запись: Имя = Запись № - 1 | Номер = 8 | Стипендия = 5765.86 |
| Запись: Имя = Запись № - 3 | Номер = 1 | Стипендия = 1514.94 |
| Запись: Имя = Запись № - 4 | Номер = 5 | Стипендия = 5739.31 |
| Запись: Имя = Новая | Номер = 7 | Стипендия = 2980.74 |

Пункт МУ 19 (SORT - pStMas - после Num):

| | | |
|----------------------------|-----------|---------------------|
| Запись: Имя = Запись № - 3 | Номер = 1 | Стипендия = 1514.94 |
| Запись: Имя = Запись № - 4 | Номер = 5 | Стипендия = 5739.31 |
| Запись: Имя = Новая | Номер = 7 | Стипендия = 2980.74 |
| Запись: Имя = Запись № - 1 | Номер = 8 | Стипендия = 5765.86 |

Пункт МУ 20 (SORT - функция - до):

| | | |
|----------------------------|-----------|---------------------|
| Запись: Имя = Запись № - 3 | Номер = 1 | Стипендия = 1514.94 |
| Запись: Имя = Запись № - 4 | Номер = 5 | Стипендия = 5739.31 |
| Запись: Имя = Новая | Номер = 7 | Стипендия = 2980.74 |
| Запись: Имя = Запись № - 1 | Номер = 8 | Стипендия = 5765.86 |

Пункт МУ 20 (SORT - функция - после):

| | | |
|----------------------------|-----------|---------------------|
| Запись: Имя = Запись № - 3 | Номер = 1 | Стипендия = 1514.94 |
| Запись: Имя = Запись № - 4 | Номер = 5 | Стипендия = 5739.31 |
| Запись: Имя = Новая | Номер = 7 | Стипендия = 2980.74 |
| Запись: Имя = Запись № - 1 | Номер = 8 | Стипендия = 5765.86 |

Пункт МУ 21 (SORT - с разными функциями):

СОРТИРОВКА: Работа с функциями Oklad(возрастание):

| | | |
|---------------------------|-----------|---------------------|
| 1 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 2 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |
| 3 - Запись: Имя = Печкин | Номер = 4 | Стипендия = 4000.00 |
| 4 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |

СОРТИРОВКА: Работа с функциями Name:

| | | |
|---------------------------|-----------|---------------------|
| 1 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| 2 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 3 - Запись: Имя = Печкин | Номер = 4 | Стипендия = 4000.00 |
| 4 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |

СОРТИРОВКА: Работа с функциями Num (убывание):

| | | |
|---------------------------|-----------|---------------------|
| 1 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |
| 2 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 3 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |

| | | |
|--|------------|---------------------|
| 4 - Запись: Имя = Печкин | Номер = 4 | Стипендия = 4000.00 |
| Пункт МУ 22:WR - ЧТЕНИЕ - 2-я | | |
| 1 - Запись: Имя = Запись 1 | Номер = 1 | Стипендия = 1000.00 |
| 2 - Запись: Имя = Изменение 2-й | Номер = 5 | Стипендия = 5000.00 |
| 3 - Запись: Имя = Запись 3 | Номер = 3 | Стипендия = 3000.00 |
| 4 - Запись: Имя = Запись 4 | Номер = 4 | Стипендия = 4000.00 |
| Запись: Имя = Изменение 2-й | Номер = 5 | Стипендия = 5000.00 |
| Пункт МУ 22: | | |
| 1 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |
| 2 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 3 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| 4 - Запись: Имя = Печкин | Номер = 4 | Стипендия = 4000.00 |
| Чтение из программы (номер = 3): | | |
| Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| Чтение из функции (номер = 2): | | |
| Запись: Имя = Изменение 2-й | Номер = 5 | Стипендия = 5000.00 |
| Ошибка номера записи- 20 ! | | |
| Чтение из функции (номер = 1): | | |
| Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| Фонд зарплаты в вычисленный циклe по файлу WR 13000.00 | | |
| Пункт МУ 22 (Интегральные расчеты): | | |
| Фонд зарплаты в вычисленный циклe = по файлу | | 16000.00 |
| Фонд зарплаты из функции = 16000.00 | | |
| Добавление (начало): | | |
| 1 - Запись: Имя = Первый | Номер = 22 | Стипендия = 3400.00 |
| 2 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |
| 3 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 4 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| 5 - Запись: Имя = Печкин | Номер = 4 | Стипендия = 4000.00 |
| Добавление (конец): | | |
| 1 - Запись: Имя = Первый | Номер = 22 | Стипендия = 3400.00 |
| 2 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |
| 3 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 4 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| 5 - Запись: Имя = Печкин | Номер = 4 | Стипендия = 4000.00 |
| 6 - Запись: Имя = Последний | Номер = 22 | Стипендия = 3400.00 |
| Добавление (номер): | | |
| 1 - Запись: Имя = Первый | Номер = 22 | Стипендия = 3400.00 |
| 2 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |
| 3 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 4 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| 5 - Запись: Имя = По номеру - 2 | Номер = 33 | Стипендия = 5500.00 |
| 6 - Запись: Имя = Печкин | Номер = 4 | Стипендия = 4000.00 |
| 7 - Запись: Имя = Последний | Номер = 22 | Стипендия = 3400.00 |
| 1 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |
| 2 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 3 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| 4 - Запись: Имя = По номеру - 2 | Номер = 33 | Стипендия = 5500.00 |
| 5 - Запись: Имя = Печкин | Номер = 4 | Стипендия = 4000.00 |
| 6 - Запись: Имя = Последний | Номер = 22 | Стипендия = 3400.00 |
| Удаление (конец): | | |
| 1 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |
| 2 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 3 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| 4 - Запись: Имя = По номеру - 2 | Номер = 33 | Стипендия = 5500.00 |
| 5 - Запись: Имя = Печкин | Номер = 4 | Стипендия = 4000.00 |
| Удаление (номер): | | |
| 1 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |
| 2 - Запись: Имя = Петров | Номер = 2 | Стипендия = 2000.00 |
| 3 - Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| 4 - Запись: Имя = По номеру - 2 | Номер = 33 | Стипендия = 5500.00 |
| Чтение (номер): | | |
| Запись: Имя = Иванов | Номер = 3 | Стипендия = 7000.00 |
| Изменение записи (номер): | | |
| 1 - Запись: Имя = Сидоров | Номер = 1 | Стипендия = 3000.00 |

2 - Запись: Имя = Петров Номер = 2 Стипендия = 2000.00
 3 - Запись: Имя = Замена Номер = 11 Стипендия = 111.00
 4 - Запись: Имя = По номеру - 2 Номер = 33 Стипендия = 5500.00

Поиск по имени:

Найдена по номеру = 2

Найдена запись: Запись: Имя = Замена Номер = 11 Стипендия = 111.00

Замена SWAP в файле по номеру: 1<->3

1 - Запись: Имя = Сидоров Номер = 1 Стипендия = 3000.00
 2 - Запись: Имя = По номеру - 2 Номер = 33 Стипендия = 5500.00
 3 - Запись: Имя = Замена Номер = 11 Стипендия = 111.00
 4 - Запись: Имя = Петров Номер = 2 Стипендия = 2000.00

Для продолжения нажмите любую клавишу . . .