

Методические указания к лабораторной работе № 10 (КЛР/ДЗ)
по курсу Основы программирования (СУЦ)
(Вариант **А** с встроенными примерами)

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

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	2
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	4
5. Техническое задание (ТЗ) для КЛР.....	5
6. Варианты для выполнения ЛР	6
7. Порядок выполнения работы (Уровень А и В).....	7
8. Дополнительные требования для сильных студентов (Уровень С):	19
9. Требование к отчету по ЛР	20
10. Контрольные вопросы к ДЗ/КЛР	20
11. Литература.....	20
12. Приложения.....	22
12.1. Главный модуль проекта LR10_OP.cpp	22
12.2. Заголовочный файл проекта LR10_OP.h.....	26
12.3. Результаты работы главного модуля примеров	36

1. Цель ЛР №10

Комплексная лабораторная работа/Домашнее задание (КЛР/ДЗ) № 10 выполняется для получения навыков совместного использования знаний по отдельным темам и приемам программирования, изучаемым в курсе “**Основы программирования**”, после последовательного выполнения отдельных лабораторных работ этой дисциплины. Лабораторные работы № 6 и № 7 связаны общей темой задания, которая базируется на конкретной структуре данных. Эта же структура данных используется и в ЛР № 10.

В предыдущих работах студенты уже изучили следующие основные темы: операторы, структуры, указатели, строки, файлы, массивы, алгоритмы и списки. В ЛР № 10 студенты совместно используют эти знания, их совместно используют. Они создают файл своих структурных переменных (набор записей) и комплекс специальных функций-операций для работы с этим файлом и этими записями. Предусматриваются следующие основные важные операции с файлом и записями: записи в файл, чтения из файла, распечатки файла, 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). Это помогает, в дальнейшем, более глубоко понять и усвоить принципы функционирования 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 подготовки и выбора отдельного студента:

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

Пункты задания (см. ниже) выделены цветом, соответствующим выбранному уровню задания. В дальнейшем мы будем использовать эти цвета для привязки к одному из уровней задания. Другими словами - цвет текста пункта задания определяет его уровень сложности. Студент может и частично использовать пункты задания из разных уровней, однако **базовый уровень (синий)** 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 случае возможен и комбинированный вариант КЛР/ДЗ.

3. Особенности описания разделения заданий по уровням ЛР №10

На сайте вы найдете несколько документов с методическими указаниями по данной комплексной лабораторной работе (с примерами и без примеров). В данном документе перечислены задания и предлагается студенту их выполнить без примеров, основанных на структуре, описанной студентом в ЛР №6. В методических указаниях с примерами(см. заголовок документа), непосредственно после пункта задания дан пример (почти как “шпаргалка”) для помощи в выполнении пунктов задания. Желательно, если студент на это способен, не подсматривая в примеры, попробовать выполнить пункт задания. Если это сделать сложно, то целесообразно детально разобраться в приведенном примере и на его основе выполнить этот пункт со своей структурой данных.

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.

Кроме того, во всех методических указаниях в приложении предоставлен в виде исходных модулей полный проект выполнения ЛР № 10 на уровне С.

4. Конструкция основного модуля программы ДЗ/ЛР№10

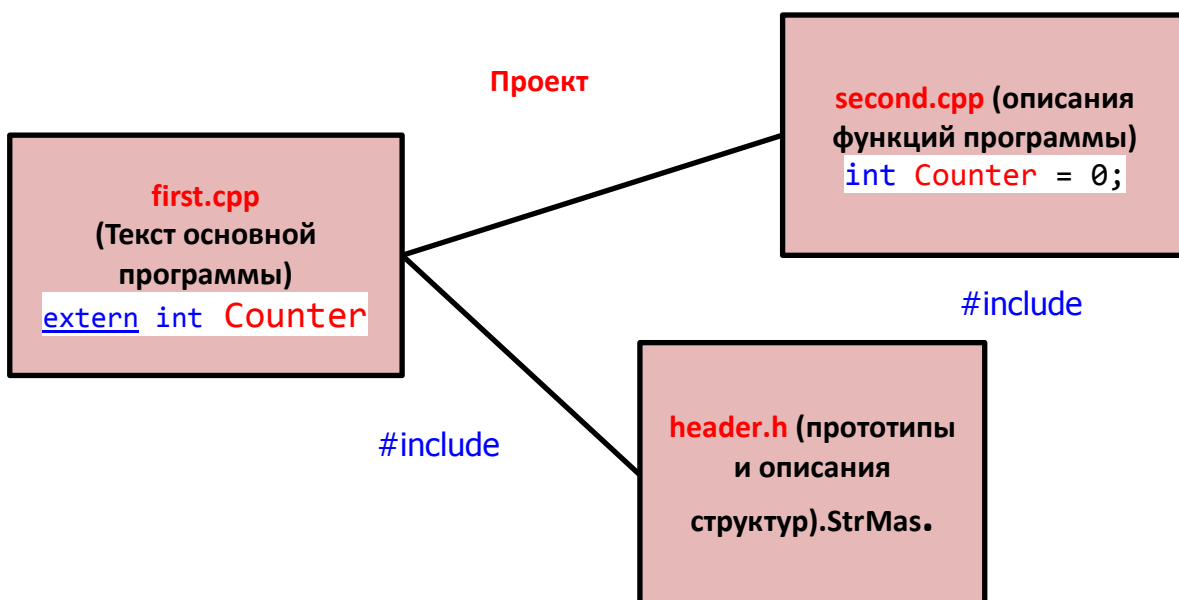
Примечание. Это требование вводится впервые в 2018 году и в примерах текстов программ образцов текста и документов может быть не полностью учтено. Необходимо самостоятельно проверить свои документы и, при необходимости доработать.

Основной модуль (first.cpp) должен быть выполнен на основе заготовки (болванки) из ЛР№1 (раздел 3.18) с меню и переключателем для выполнения пунктов ТЗ. Наименования пунктов ТЗ должны соответствовать пунктам ТЗ из раздела 5.1 ТЗ. Выполнение каждого пункта меню должно выполняться автономно и в произвольном порядке! Модули образцов ДЗ должны быть переработаны.

5. Функции и структуры в домашнем задании

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 функция), 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 назначается или выбирается во время выполнения ЛР №6 и используется студентом в последующих ЛР. Эта структура данных должна иметь осмысленные поля и представлять единое целое с точки зрения предметной области.. Например, для описания студента можно выделить поля: фамилия студента, курс студента, размер стипендии и т.д. Часть названий полей уже предложены в варианте (см. ниже), хотя по согласованию с преподавателем их можно заменить. Неизменным должно оставаться только содержательное англоязычное название структуры данных (Student - студент, Book - книга и т.д.).

6. Структура проекта программы для работы с файлами для ДЗ/КЛР

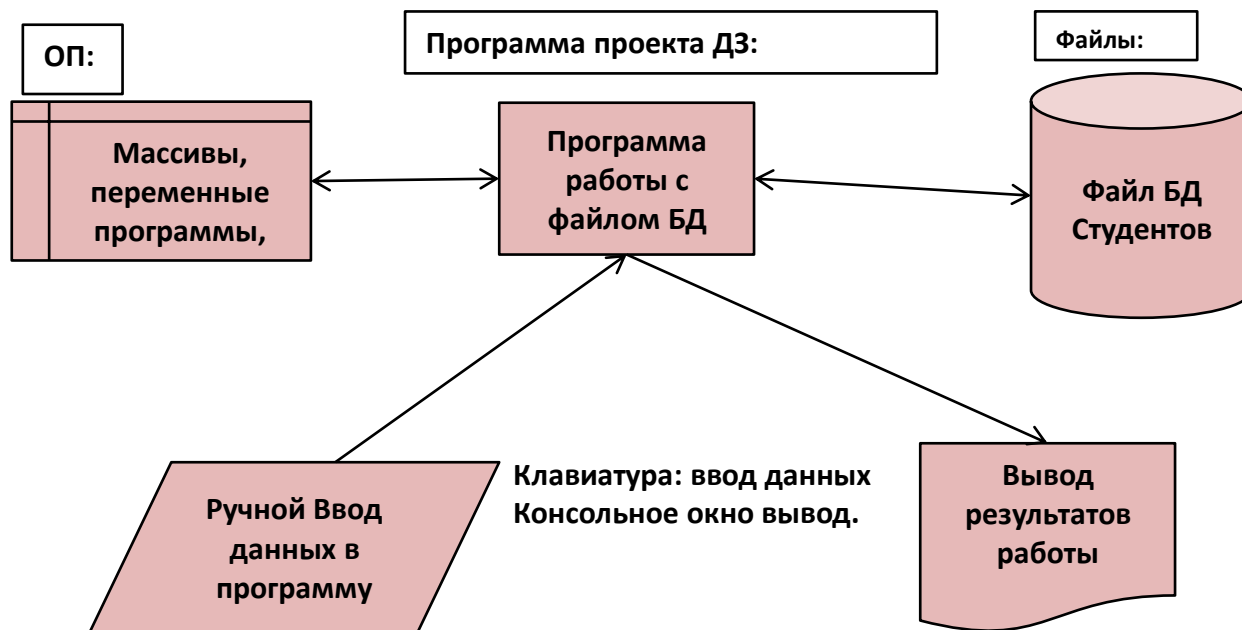


В проект нужно включить три модуля:

- **first.cpp** (для главной программы проекта),
- **second.cpp** (для размещения функций проекта) и
- **header.h** (для описаний структур и прототипов функций проекта). При подключении своего модуля из текущего каталога нужно использовать кавычки:
`#include "header.h"`

Примечание: Если поместить описание (`int Counter = 0;`) в `second.cpp`, то в `first.cpp` нужно указывать `:extern int Counter;` для доступа к этой внешней переменной.

7. Функциональная схема работы программы с БД Студентов для ДЗ/КЛР



В данной схеме выделены следующие составляющие:

- Программа для работы с файлом БД (исполнимый модуль VS)
- Файл, содержащий записи структур студентов.
- Клавиатура: ввод данных
- Консольное окно для вывода данных.
- ОП – оперативная память для хранения переменных и массивов программы.

8. 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
- Шаблон Техническое 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. ЛРН№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 приступать к выполнению ДЗ. Более подробно о разработке ТЗ будет сказано на лекциях и семинарах.

9. 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.

Примечание: Поля в таблице для разных вариантов приведены в качестве примера. Каждый студент вправе использовать собственные типы данных, названия полей и их типы.

№ 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
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				

№ Вар.	Структура (имя структуры придумать самому)	Поля структуры и их тип			Примечание (интегральн. характеристики)
		Поле-имя-тип	Поле-имя-тип	Поле-имя-тип	
	Студент (Student)	Имя (Name) - char[20]	Номер (Num) - int	Стипендия (Oklad)-double	Сумма стипендии студентов в файле

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 до 11-й недели семестра.

10. Порядок выполнения работы (Уровень А и В)

10.1.1. Порядок выполнения работы (Уровень А и В)

1. **Создать** и русифицировать свой консольный проект (главная программа, при необходимости вспомогательный модуль для функций и заголовочный файл для определений/прототипов функций, структур/констант и т.д.)

Для создания консольного проекта необходимо (общий вариант для всех ЛР и ДЗ):

- Запустить систему программирования VS 2005/8/10/12 и т.д.;
- 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 – LAB10_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 LAB10_XDD.H) все лишнее. Этого: в файле LAB10_XDD.H (у нас в примере LAB10_15.H) уберем все, а в файле LAB10_XDD.CPP (LAB10_15.CPP) оставим только следующий текст:

```
#include "lab10_15.h" // Поправить индекс группы и вариант
#include <string.h>
#include <stdlib.h>
#include <process.h>
#include <stdio.h> // Подключение библиотеки ввода вывода
#include <io.h> // Подключение библиотеки
void main(void)
{
// ...
}
```
- Далее нужно добавить заголовочный модуль в проект: LAB10_XDD.H (у нас в примере LAB10_15.H) - в раздел **“Header Files”**. Нажмем правую кнопку на этом тексте, а затем: **“Add”** -> **“New Item ...”** -> **“Code”** -> **“Header File”** -> Ввод поля **“Name”**;

- 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.

В проекте ДЗ должны быть включены минимально три модуля: **first.cpp**, **second.cpp**, **header.h**. См. выше (раздел 5).

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 . . .

Примечание 3: Обратите внимание на то, что при другом способе локализации (`setlocale(0,"rus");`) не все работает правильно. Вывод на консоль и ввод на/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.

10.2.2.Порядок выполнения работы (Уровень А и В)

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).

```
//////// ЗАГОЛОВОЧНЫЙ ФАЙЛ – header.h
// МУ п.2 (СИНИЙ)
////////
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
    // char Date[10]; // Резерв пока в примере нет для краткости
} Stud;
...
```

10.3.3.Порядок выполнения работы (Уровень А и В)

3. **Описать** и заполнить одну свою простую структурную переменную с помощью инициализации при ее описании. Распечатать структуру (функция **printf**).

```
//////// ГЛАВНАЯ ПРОГРАММА first.cpp
...
#include <math.h>
#include <process.h>
```



```
#include <stdio.h>
#include <string.h>
#include "header.h"
...

// МУ п.3 (СИНИЙ)
/////////
Student Stud1 = { "Лаптева", 1, 1000.0 }; // Инициализация трех полей
printf( "Stud1: Имя = %-15s Номер = %2d Стипендия = %8.2lf \n\n",
        Stud1.Name, Stud1.Num, Stud1.Oklad );
```

...

Результат работы программы:

Запись: Имя = Лаптева Номер = 1 Стипендия = 1000.00

10.4.4. Порядок выполнения работы (Уровень А и В)

4. **Описать** и заполнить еще одну свою структурную переменную вручную (с помощью операторов присваивания – числовые данные и функции копирования строк – строчные данные). Распечатать значения всех ее полей двух структурных переменных (Функция - **printf**).

///////// ГЛАВНАЯ ПРОГРАММА - first.cpp

```
...
#include <string.h>
...
// МУ п.4 (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 );
```

Результат работы программы:

Пункт МУ 4:

Запись: Имя = Аксенова Номер = 2 Стипендия = 2000.00

10.5.5. Порядок выполнения работы (Уровень А и В)

5. **Разработать** две **функции** распечатки одной структурной переменной вашего варианта, передавая в качестве параметра саму структуру и ее адрес. Проверить эти функции на описанных ранее структурных переменных (п. 3 и 4).

```
///////// ФАЙЛ second.cpp
///////// РЕАЛИЗАЦИЯ ФУНКЦИИ
#include <math.h>
#include <process.h>
#include <stdio.h>
#include <string.h>
#include "header.h"
// МУ п.5 (СИНИЙ)
// Распечатка отдельной структурной переменной студента

///////// Структура
void PrintOneStudent(Student S)
{
    printf( "Запись: Имя = %-15s Номер = %2d Стипендия = %8.2lf \n",
            S.Name, S.Num, S.Oklad );
```

```

};
////////
///////// Указатель
void PrintStudent(Student * pS)
{
    printf( "Запись: Имя = %-15s Номер = %2d Стипендия = %8.2lf \n",
        pS->Name , pS->Num, pS->Oklad );
};
...
///////// ПРОТОТИПЫ header.h
// МУ п.5 (СИНИЙ)
void PrintStudent(Student * pS);
void PrintOneStudent(Student S);
...
///////// ГЛАВНАЯ ПРОГРАММА - first.cpp
// МУ п.5 (СИНИЙ)
/////////
    printf ("Пункт МУ 5:\n");
    PrintStudent(&Stud1) ;
    PrintStudent(&Stud2) ;
    PrintOneStudent(Stud1) ;
    PrintOneStudent(Stud2) ;

```

...
Результат работы программы:

Пункт МУ 5:

Запись: Имя = Лаптева	Номер = 1	Стипендия = 1000.00
Запись: Имя = Аксенова	Номер = 2	Стипендия = 2000.00
Запись: Имя = Лаптева	Номер = 1	Стипендия = 1000.00
Запись: Имя = Аксенова	Номер = 2	Стипендия = 2000.00

10.7.7.Порядок выполнения работы (Уровень А и В)

7. **Описать** и проинициализировать при описании массив структурных переменных (размерность массива не менее 4-х). Распечатать этот массив в цикле с помощью функций п.5 (через указатель и структуру).

```

///////// ГЛАВНАЯ ПРОГРАММА first.cpp
// МУ п.7 (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] ) ; // Печать через указатель
    PrintOneStudent(SMas[i]) ; // Печать через структуру
}
...

```

Результат работы программы:

Пункт МУ 7:

Запись: Имя = Первый	Номер = 1	Стипендия = 1000.00
Запись: Имя = Первый	Номер = 1	Стипендия = 1000.00
Запись: Имя = Второй	Номер = 2	Стипендия = 2000.00
Запись: Имя = Второй	Номер = 2	Стипендия = 2000.00
Запись: Имя = Третий	Номер = 3	Стипендия = 3000.00

Запись: Имя = Третий	Номер = 3	Стипендия = 3000.00
Запись: Имя = Четвертый	Номер = 4	Стипендия = 4000.00
Запись: Имя = Четвертый	Номер = 4	Стипендия = 4000.00

10.8.8.Порядок выполнения работы (Уровень А и В)

8. **Разработать** функцию распечатки массива с записями своих структурных переменных. Формальные параметры этой функции (их два): указатель на массив структурных переменных и его размер массива. Использовать свою функцию печати одной структурной переменной через указатель. Проверить работу этой функции на массиве созданном в п.п. 7.

```

////////// РЕАЛИЗАЦИЯ ФУНКЦИИ
////////// ФАЙЛ second.cpp
// МУ п.8 (СИНИЙ)
//////////
void StudPrintMas( Student * pMas , int Razm)
{
    for (int i = 0 ; i < Razm ; i++ )
    {
        PrintStudent(pMas + i); // Используется готовая функция
    };
};
...
////////// ПРОТОТИП header.h
// МУ п.8 (СИНИЙ)
void StudPrintMas( Student * pMas , int Razm);
...
////////// ГЛАВНАЯ ПРОГРАММА first.cpp
// МУ п.8 (СИНИЙ)
//////////
printf ("Пункт МУ 8:\n");
StudPrintMas( SMas , RazmS);
...

```

Результат работы программы:

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

10.10.10.Порядок выполнения работы (Уровень А и В)

10. **Придумать** название двоичному файлу для выполнения задания (в нашем примере - "BDStud.bin"— для базы данных студентов).

```

////////// ГЛАВНАЯ ПРОГРАММА
// МУ п.10 (СИНИЙ)
// "BDStud.bin" – имя моего файла для работы с записями структурных переменных (студенты)
//////////

```

10.11.11.Порядок выполнения работы (Уровень А и В)

11. **Создать** цикл для заполнения своего файла своими структурными записями (запись в файл). Запись выполняется на основе массива 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). Использовать функции для работы с файлами среднего уровня (**fopen**, **fclose**, **fwrite**, **fread** и **feof** – см. ЛР№7). Проверить результат создания и заполнения файла с помощью **файл менеджера** или **Notepad.exe** (в символьном и шестнадцатеричном представлении). При желании можно снять снимок с экрана (скриншот) и поместить в отчет.

```

////////
//////// ГЛАВНАЯ ПРОГРАММА first.cpp
// МУ п.11 (СИНИЙ)
////////
FILE * pFN;
Student ZNMas[] ={{"Запись 1N" , 10 , 11000.0}, {"Запись 2N" , 20 , 12000.0},
{"Запись 3N" , 30 , 13000.0},
{"Запись 4N" , 40 , 14000.0}};
int RazmZ = sizeof(ZNMas)/ sizeof(Student); // Вычисление размерности массива
printf ("Пункт МУ 11(WR исходный массив):\n");
StudPrintMas( ZNMas , RazmZ); // Распечатка массива
FILE * pFN;
pFN = fopen( "BDStud.bin", "w+b"); // Открытие файла для записи
for ( int i = 0 ; i < RazmZ; i++)
    fwrite( &ZNMas[i], sizeof(Student) , 1, pFN); // Запись в файл
fclose(pFN); // Закрытие файла
//////////
...

```

Результат работы программы (WR – fwrite, fread) (печать массива):

Пункт МУ 11(WR исходный массив):

Запись: Имя = Запись 1N	Номер = 10	Стипендия = 11000.00
Запись: Имя = Запись 2N	Номер = 20	Стипендия = 12000.00
Запись: Имя = Запись 3N	Номер = 30	Стипендия = 13000.00
Запись: Имя = Запись 4N	Номер = 40	Стипендия = 14000.00

11а. **Создать** цикл чтения и распечатки ранее сформированного двоичного файла из записей. Данные из файла запомнить в статическом массиве структур, задав его максимальный размер массива равным 100. При чтении записей каждую распечатать своей функцией (п.5). Число введенных записей подсчитать в цикле чтения. Ограничить число вводимых записей в массив – не более 100. После ввода всех записей в массив распечатать массив своей функцией (п. 8).

```

////////////////////////////////////
// МУ п.11а (СИНИЙ) first.cpp
// Чтение и распечатка файла
FILE * pFN;
Student MASStudWR[100]; // Массив для записи с запасом
pFN = fopen( "BDStud.bin" , "r+b"); // Открытие файла для чтения
Student SBuf1;
int i = 0;
printf ("Пункт МУ 11(WR чтение и распечатка файла):\n");
while (!feof(pFN) && i < 100) // Проверка конца файла или размера массива
{
    fread( &SBuf1, sizeof(Student) , 1, pFN); // чтение одной записи
    if ( !feof(pFN) )
    {
        memcpy( &MASStudWR[i] , &SBuf1, sizeof(Student));
        PrintStudent(&SBuf1); // Распечатка одной записи
    };
    RazmZ = i; // запомним размер файла в записях = размер массива
    i++;
};
// Закрытие файла
fclose( pFN );

```

```
printf ("Пункт МУ 11(WR Печать из массива - функция):\n");
StudPrintMas( MASStudWR , RazmZ);
```

...

Результат работы программы (WR – fwrite, fread):

Пункт МУ 11(WR чтение и распечатка файла):

Запись: Имя = Запись 1N	Номер = 10	Стипендия = 11000.00
Запись: Имя = Запись 2N	Номер = 20	Стипендия = 12000.00
Запись: Имя = Запись 3N	Номер = 30	Стипендия = 13000.00
Запись: Имя = Запись 4N	Номер = 40	Стипендия = 14000.00

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

Запись: Имя = Запись 1N	Номер = 10	Стипендия = 11000.00
Запись: Имя = Запись 2N	Номер = 20	Стипендия = 12000.00
Запись: Имя = Запись 3N	Номер = 30	Стипендия = 13000.00
Запись: Имя = Запись 4N	Номер = 40	Стипендия = 14000.00

10.13.13.Порядок выполнения работы (Уровень А и В)

13. Разработать функцию заполнения двоичного файла на основе массива структурных переменных (описанных и проинициализированных ранее). В функцию должны передаваться: имя заполняемого файла, адрес исходного массива, размер массива структур. Изменить некоторые значения записей в массиве структур перед использованием. Для этого в самом массиве, заполненном выше, сделать изменения записи по индексу (например, имя 2-й записи: имя и номер), используя фиксированное значение индекса элемента структурного массива (индекс = 1). Проверить работу этой функции. Заполнение файла и изменения в нем проверить в файл-менеджере (**far** или **total commander**) или Notepad.exe, с помощью операции просмотра файла в шестнадцатеричном формате.

```
//////// ФАЙЛ second.cpp header.h
//////// РЕАЛИЗАЦИЯ ФУНКЦИИ
//Перезапись массива структур в файл (Файл очищается!)
// МУ п.13 (СИНИЙ)
////////
//Перезапись массива структур в файл (Файл предварительно очищается!) WR
///// Вариант функций ( fopen, fclose, fwrite, fread и feof)
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);
};
/// ПОТОТИП ФУНКЦИИ header.h
// МУ п.13 (СИНИЙ)
void StudMasToFileWR( const char * FileName , Student * pMas , int Razm);
...
//////// ГЛАВНАЯ ПРОГРАММА first.cpp second.cpp header.h
// МУ п.13 (СИНИЙ)
////////
...
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 записей в массиве второй записи(индекс = 1): номер и имя
NZMas[1].Num =5;
strcpy( NZMas[1].Name , "Изменение 2-й");
///// Вариант функций ( fopen, fclose, fwrite, fread и feof)
```

```
printf ("Пункт МУ 13:WR\n");
StudMasToFileWR( "BDStud.bin" , NZMas, RazmZ); // Массив записываем в файл!!!!
StudPrintFileWR( "BDStud.bin" );
```

...

Результат работы программы (WR):

Пункт МУ 13:WR

Запись: Имя = Запись 1	Номер = 11	Стипендия = 51000.00
Запись: Имя = Изменение 2-й	Номер = 5	Стипендия = 55000.00
Запись: Имя = Запись 3	Номер = 13	Стипендия = 53000.00
Запись: Имя = Запись 4	Номер = 14	Стипендия = 54000.00

Результат работы программы файла - **BDStud.bin** (или при повторном выполнении - **BDStudN.bin**) нужно посмотреть также в любом файл менеджере.

10.14.14.Порядок выполнения работы (Уровень А и В)

14. **Разработать** отдельную функцию распечатки файла с записями своих структурных переменных своего типа. В функцию передается только имя файла. Конец цикла чтения определяется функцией **feof**. Использовать функции: **fopen**, **fclose**, **fwrite**, **fread**. Проверить работу функции распечатки файла.

```
//////// РЕАЛИЗАЦИЯ ФУНКЦИИ second.cpp
// МУ п.14 (СИНИЙ)
//////////
// Печать файла для структуры Student
///// Вариант функций ( fopen, fclose , fwrite, fread и feof)
void StudPrintFileWR( const char * FileName)
{
    FILE * pF;
    pF = fopen( FileName , "r+b"); // Открытие файла для чтения
    Student SBuf;
    while (!feof(pF) ) // Проверка конца файла
    {
        fread( &SBuf, sizeof(Student) , 1, pF); // чтение одной записи
        if ( !feof(pF) )
            PrintStudent(&SBuf);
    };
    // Закрытие файла
    fclose( pF );
};
...
////////// ПРОТОТИП header.h
// МУ п.14 (СИНИЙ)
void StudPrintFileWR( const char * FileName);
```

...

```
////////// ГЛАВНАЯ ПРОГРАММА first.cpp
// МУ п.14 (СИНИЙ)
//////////
// Распечатка файла записей
///// Вариант функций ( fopen, fclose , fwrite, fread и feof)
printf ("Пункт МУ 14:WR\n");
StudPrintFileWR( "BDStudN.bin" );
```

...

Результат работы программы:

Пункт МУ 14:WR

Запись: Имя = Запись 1	Номер = 11	Стипендия = 51000.00
Запись: Имя = Изменение 2-й	Номер = 5	Стипендия = 55000.00

Запись: Имя = Запись 3	Номер = 13	Стипендия = 53000.00
Запись: Имя = Запись 4	Номер = 14	Стипендия = 54000.00

10.18.18.Порядок выполнения работы (Уровень А и В)

18. **Разработать** функцию **Swap** для обмена двух структурных переменных (по адресу структурных переменных). Структурные переменные имеют фиксированный размер. Продемонстрировать и проверить работу этой функции на отдельных переменных и отдельных элементах массива структур. Распечатать переменные (элементы массива) до и после вызова функции **Swap**.

```

//////// РЕАЛИЗАЦИЯ ФУНКЦИИ
// МУ п.18 (СИНИЙ) second.cpp
#include <memory.h> // Библиотека для memcpy

//////////
// Обмен на основе копирования структурных переменных (так можно только без динамики!!)
void SwapStudent ( Student * pA , Student * pB )
{
    Student Temp;
    memcpy( &Temp , pA , sizeof(Student) ); // Обмен копированием в памяти
    memcpy( pA , pB , sizeof(Student) );
    memcpy( pB , &Temp , sizeof(Student) ); }

...
////////// ПРОТОТИП
// МУ п.18 (СИНИЙ) header.h
void SwapStudent ( Student * pA , Student * pB );

...
////////// ГЛАВНАЯ ПРОГРАММА first.cpp
// МУ п.18 (СИНИЙ)
//////////
int StudCount;
printf ("Пункт МУ 17 снова добавим записи из массива:\n");
StudMasToFileWR( "BDStud.bin" , NZMas, RazmZ);
StudPrintFileWR( "BDStud.bin" );
    printf ("Пункт МУ 18 (Swap - до):\n");
Student StudA ={ "Лаптева" , 1 , 1000.0};
Student StudB ={ "Иванова" , 2 , 2000.0};
    PrintStudent(&StudA) ;
    PrintStudent(&StudB) ;
printf ("Пункт МУ 18 (Swap - после):\n");
SwapStudent( &StudA , &StudB); // Обмен двух отдельных записей
    PrintStudent(&StudA) ;
    PrintStudent(&StudB);

// В массиве
Student NZMasSW[] ={{"Запись 1" , 1 , 1000.0}, {"Запись 2" , 2 , 5000.0}, {"Запись 3"
, 3 , 3000.0},
    {"Запись 4" , 4 , 4000.0}};
RazmZ = sizeof(NZMasSW)/ sizeof(Student); // Вычисление размерности массива
//
printf ("Пункт МУ 18 (Swap - NZMasSW - до):\n");
StudPrintMas( NZMasSW, RazmZ);
printf ("Пункт МУ 18 (Swap - NZMasSW - после):\n");
SwapStudent( &NZMasSW[0] , &NZMasSW[2]); // Обмен двух элементов массива
StudPrintMas( NZMasSW, RazmZ);

...

```

Результат работы программы:

Пункт МУ 18 (Swap - до):

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

Пункт МУ 18 (Swap - после):

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

Запись: Имя = Лаптева	Номер = 1	Стипендия = 1000.00
Пункт МУ 18 (Swap - NZMasSW - до):		
Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00
Пункт МУ 18 (Swap - NZMasSW - после):		
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00
Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00
Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00

10.19.19.Порядок выполнения работы (Уровень А и В)

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. Проверить сортировку по символьному полю (strcmp).

```

////////// ГЛАВНАЯ ПРОГРАММА first.cpp
// Сортировка массива по одному полю записи
// В примере рассмотрены 3 варианта: числовое поле Oklad, числовое поле Num, и строка
Name
//////////
// МУ п.19 (СИНИЙ)
//////////
Student NZMasSORT[] ={{"Запись 1" , 1 , 1000.0}, {"Запись 2" , 2 , 5000.0}, {"Запись
3" , 3 , 3000.0},
                {"Запись 4" , 4 , 4000.0}};
StudCount= sizeof(NZMasSORT)/ sizeof(Student);
printf ("Пункт МУ 19 (SORT - NZMasSORT - до):\n");
StudPrintMas( NZMasSORT, StudCount);
// Сортировка по окладу
for( int k =0 ; k< StudCount - 1 ; k++)
    for ( int i =0 ; i< StudCount - 1 ; i++ )
        if ( (NZMasSORT + i)->Oklad > (NZMasSORT + i + 1)->Oklad ) //
            SwapStudent(NZMasSORT + i , NZMasSORT + i + 1 );
// Возрастание
printf ("Пункт МУ 19 (SORT - NZMasSORT - после Oklad):\n");
StudPrintMas( NZMasSORT, StudCount);
//
// Сортировка по номеру
for( int k =0 ; k< StudCount - 1 ; k++)
    for ( int i =0 ; i< StudCount - 1 ; i++ )
        if ( (NZMasSORT + i)->Num < (NZMasSORT + i + 1)->Num ) //
            SwapStudent(NZMasSORT + i , NZMasSORT + i + 1 );
// Убывание
printf ("Пункт МУ 19 (SORT - NZMasSORT - после Num):\n");
StudPrintMas( NZMasSORT, StudCount);
//
// Сортировка по имени
for( int k =0 ; k< StudCount - 1 ; k++)
    for ( int i =0 ; i< StudCount - 1 ; i++ )
        if ( strcmp((NZMasSORT + i)->Name , (NZMasSORT + i + 1)->Name ) > 0 ) //
            SwapStudent(NZMasSORT + i , NZMasSORT + i + 1 );
// Возрастание
printf ("Пункт МУ 19 (SORT - NZMasSORT - после Name):\n");
StudPrintMas( NZMasSORT, StudCount);
//////////
Результат работы программы:
Пункт МУ 19 (SORT - NZMas - до):

```


Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00

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

Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00

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

Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00
Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00

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

Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00

10.22.22.Порядок выполнения работы (Уровень А и В)

22.1 Прочитать 2-ю запись из сформированного файла структурных переменных и ее распечатать с помощью собственной функции. Проверить число записей в файле (функция filelength и sizeof) и, если номер ошибочный, выдать сообщение: "Ошибка номера записи!". Использовать функции: fopen, fclose, fread. Предварительно проверить чтение записи своей функцией распечатки файла (п. 14).

```

//////// first.cpp
// МУ п.22.1 чтение одной записи 2-й (0 - 4 )
//////// (СИНИЙ)
int nFind = 2 ; // Читаем 2-ю запись считая ч 1 - й
int Pos1;
Student SBuf; // Буферная структура для чтения
//
StudPrintFileWR( "BDStud.bin" ); // Предварительная распечатка файла
printf ("Пункт МУ 22:WR - ЧТЕНИЕ - 2-я\n");
pFN = fopen( "BDStud.bin" , "rb"); // Открытие файла для чтения
// Перемещение указателя
if (filelength(pFN->_file)/sizeof(Student) >= nFind )
{
    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 (2-й)
    PrintStudent(&SBuf);
}
else
    printf( "Ошибка номера записи!- %d !\n" , nFind ); // Ошибка номера

// Заккрытие файла
fclose( pFN );
//////////
...

```

Результат работы программы:

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 22:WR - 2-я

2. Запись: Имя = Изменение 2-й	Номер = 5	Стипендия = 5000.00
--------------------------------	-----------	---------------------

10.23.23_2.Порядок выполнения работы (Уровень А и В)

22.2 Создать функцию для чтения из сформированного файла структурных переменных отдельной записи по номеру (отдельный параметр функции) и ее распечатать с помощью собственной функции. В функции предусмотреть три параметра: имя файла записей, буфер-структура для чтения записи и номер считываемой записи. В функции проконтролировать номер записи для чтения и, при необходимости, выдать диагностическое сообщение: "Ошибка номера записи!". Использовать функции: `fopen`, `fclose`, `fread`. Проверить функцию в правильном и ошибочном режиме. Файл структур предварительно распечатать своей функцией (п. 14).

```

////////////////////
//////// РЕАЛИЗАЦИЯ ФУНКЦИИ
////////////////////
// Поиск и выборка одной записи по номеру
// МУ п.22.2 (СИНИЙ) second.cpp
#include <io.h>
////////////////////
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
pS->Num = 0; // Очистка полей структуры
pS->Oklad = 0.0;
strcpy(pS->Name , "");
};
// Закрытие файла
fclose( pF );
};
...
//////// ПРОТОТИП
// МУ п.22.2 (СИНИЙ) header.h
void FindStudNumWR( const char * FileName , Student * pS , int Numb);
...
//////// ГЛАВНАЯ ПРОГРАММА
// МУ п.22.2 чтение из файла по номеру из функции
//////// (СИНИЙ) first.cpp
///// - Прочитать запись по номеру
Student SFind; // Буфер структура для чтения записи
printf( "Чтение из функции (номер = 2):\n");
FindStudNumWR( "BDStud.bin" , &SFind , 2);
PrintStudent( &SFind );
FindStudNumWR( "BDStud.bin" , &SFind , 20); // Ошибка номера
PrintStudent( &SFind );
...

```

Содержание файлов BDStud.bin и BDStudN.bin:

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

...

Результат работы программы:

Чтение из функции (номер = 2):

Запись: Имя = Изменение 2-й Номер = 5 Стипендия = 5000.00

Ошибка номера записи- 20 !

Запись: Имя = Номер = 0 Стипендия = 0.00

10.24.23_3.Порядок выполнения работы (Уровень А и В)

22.3 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. Для работы с файлом использовать функции: `fopen`, `fclose`, `fread` и `feof`. Шаги выполнения этой задачи могут и 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.

//////// ГЛАВНАЯ ПРОГРАММА

// МУ п.22.3 (СИНИЙ)

//////// Чтение и подсчет непосредственно из файла

FILE * pFS;

double SumOk = 0.0 ; // Переменная для суммирования стипендии всех студентов из файла

Student SFromFile; // Структура для чтения из файла

pFS = fopen("BDStud.bin" , "r+b"); // Открытие файла для чтения

if (pFS != NULL) // Проверка открытия файла

{

 while (!feof(pFS)) // Проверка конца файла

{

 fread(&SFromFile, sizeof(Student) , 1, pFS); // чтение одной записи

 if (!feof(pFS)) // Проверка конца файла для исключения лишнего

суммирования для последней записи

 SumOk = SumOk + SFromFile.OkIad;

 };

 printf ("Фонд зарплаты в вычисленный цикл по файлу = %7.3lf \n", SumOk);

// Закрытие файла

 fclose(pFS);

};

Результат работы программы:

Пункт МУ 22 (Интегральные расчеты):

Фонд зарплаты в вычисленный цикл по файлу = 213000.000

22.4 Оформить в виде **функции** фрагмент программы для вычисления интегральной характеристики по вашему варианту (см. предыдущее задание). (см. вариант методических указаний для С).

Для всех заданий, 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.

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

Дополнительные требования изложены в отдельном файле заданий(А-С).

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.

Работа считается выполненной с дополнительными требованиями, если выполнены любые шесть пунктов из красного раздела.

12. Требование к отчету по ЛР

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

13. 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

14. 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 (Vicrosoft 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.

15. Приложения

15.1. first.cpp Главный модуль проекта

first.cpp – ГЛАВНЫЙ МОДУЛЬ (по порядку)

// Комплексная лабораторная работа по ОП СУЦ

```
#include <math.h>
#include <process.h>
#include <stdio.h>
#include <string.h>
#include <io.h>
#include "header.h"
void main(void)
{
    system(" chcp 1251 > nul");
    printf("ДЗ по ОП №!!!\n");
    ///
    ////////// ГЛАВНАЯ ПРОГРАММА
    // МУ п 2.3 (СИНИЙ)
    //////////
    Student Stud1 = { "Лаптева" , 1 , 1000.0}; // Инициализация трех полей
    printf( "Stud1: Имя = %-15s Номер = %2d Стипендия = %8.2lf \n\n",
        Stud1.Name , Stud1.Num, Stud1.Oklad );
    ////////// ГЛАВНАЯ ПРОГРАММА
    // МУ п.4 (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 );
    ////////// ГЛАВНАЯ ПРОГРАММА - first.cpp
    // МУ п.5 (СИНИЙ)
    //////////
    printf ("Пункт МУ 5:\n");
    PrintStudent(&Stud1) ;
    PrintStudent(&Stud2) ;
    PrintOneStudent(Stud1) ;
    PrintOneStudent(Stud2) ;
    ////////// ГЛАВНАЯ ПРОГРАММА
    // МУ п.7 (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] ) ; // Печать через указатель
        PrintOneStudent(SMas[i]) ; // Печать через структуру
    }
    ////////// ГЛАВНАЯ ПРОГРАММА first.cpp
    // МУ п.8 (СИНИЙ)
    //////////
    printf ("Пункт МУ 8:\n");
    StudPrintMas( SMas , RazmS);
```

```

////////// ГЛАВНАЯ ПРОГРАММА
// МУ п.10 (СИНИЙ)
// "BDStud.bin" - имя моего файла для работы с записями структурных переменных
(студенты)
//////////
// МУ п.11 (СИНИЙ)
//////////
Student ZNMas[] ={{"Запись 1N" , 10 , 11000.0}, {"Запись 2N" , 20 , 12000.0},
{"Запись 3N" , 30 , 13000.0},
{"Запись 4N" , 40 , 14000.0}};
int RazmZ = sizeof(ZNMas)/ sizeof(Student); // Вычисление размерности массива
printf ("Пункт МУ 11(WR исходный массив):\n");
StudPrintMas( ZNMas , RazmZ); // Распечатка массива
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); // Запись в файл
fclose(pFN); // Закрытие файла
///
//////////
// МУ п.11a (СИНИЙ)
// Чтение и распечатка файла
Student MASStudWR[100]; // Массив для записи с запасом
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) && i < 100) // Проверка конца файла или размера массива
{
    fread( &SBuf, sizeof(Student) , 1, pFN); // чтение одной записи
    if ( !feof(pFN) )
    {
        memcpy( &MASStudWR[i] , &SBuf , sizeof(Student));
        PrintStudent(&SBuf); // Распечатка одной записи
    };
    RazmZ = i; // запомним размер файла в записях = размер массива
    i++;
};
// Закрытие файла
fclose( pFN );
printf ("Пункт МУ 11(WR Печать из массива - функция):\n");
StudPrintMas( MASStudWR , RazmZ);
//////////
////////// ГЛАВНАЯ ПРОГРАММА first.cpp
// МУ п.13 (СИНИЙ)
//////////
//
Student NZMas[] ={{"Запись 1" , 11 , 51000.0}, {"Запись 2" , 12 , 55000.0}, {"Запись
3" , 13 , 53000.0},
{"Запись 4" , 14 , 54000.0}};
RazmZ = sizeof(NZMas)/ sizeof(Student); // Вычисление размерности массива
// Изменение записей в массиве второй записи(индекс = 1): номер и имя
NZMas[1].Num =5;
strcpy( NZMas[1].Name , "Изменение 2-й");
///// Вариант функций ( fopen , fclose , fwrite , fread и feof)
printf ("Пункт МУ 13:WR\n");
StudMasToFileWR( "BDStudN.bin" , NZMas, RazmZ); // Массив записываем в файл!!!!
//StudPrintFileWR( "BDStudN.bin" ); // из п. 14
//
////////// ГЛАВНАЯ ПРОГРАММА first.cpp
// МУ п.14 (СИНИЙ)
//////////

```



```

// Распечатка файла записей
///// Вариант функций ( fopen, fclose , fwrite, fread и feof)
printf ("Пункт МУ 14:WR\n");
StudPrintFileWR( "BDStudN.bin" );
//////////
////////// ГЛАВНАЯ ПРОГРАММА first.cpp
// МУ п.18 (СИНИЙ)
//////////
int StudCount= sizeof(NZMas)/ sizeof(Student); // Вычисление размерности массива;
printf ("Пункт МУ 17 снова добавим записи из массива:\n");
StudMasToFileWR( "BDStud.bin" , NZMas, RazmZ);
StudPrintFileWR( "BDStud.bin" );
    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
Student NZMasSW[] ={{"Запись 1" , 1 , 1000.0}, {"Запись 2" , 2 , 5000.0}, {"Запись 3"
, 3 , 3000.0},
    {"Запись 4" , 4 , 4000.0}};
RazmZ = sizeof(NZMasSW)/ sizeof(Student); // Вычисление размерности массива
//
printf ("Пункт МУ 18 (Sawp - NZMasSW - до):\n");
StudPrintMas( NZMasSW, RazmZ);
printf ("Пункт МУ 18 (Sawp - NZMasSW - после):\n");
SwapStudent( &NZMasSW[0] , &NZMasSW[2]); // Обмен двух элементов массива
StudPrintMas( NZMasSW, RazmZ);
///
////////// ГЛАВНАЯ ПРОГРАММА first.cpp
// Сортировка массива по одному полю записи
// В примере рассмотрены 3 варианта: числовое поле Oklad, числовое поле Num, и строка
Name
//////////
// МУ п.19 (СИНИЙ)
//////////
Student NZMasSORT[] ={{"Запись 1" , 1 , 1000.0}, {"Запись 2" , 2 , 5000.0}, {"Запись
3" , 3 , 3000.0},
    {"Запись 4" , 4 , 4000.0}};
StudCount= sizeof(NZMasSORT)/ sizeof(Student);
printf ("Пункт МУ 19 (SORT - NZMasSORT - до):\n");
StudPrintMas( NZMasSORT, StudCount);
// Сортировка по окладу
    for( int k =0 ; k< StudCount - 1 ; k++)
        for ( int i =0 ; i< StudCount - 1 ; i++ )
            if ( (NZMasSORT + i)->Oklad > (NZMasSORT + i + 1)->Oklad ) //
Возрастание
                SwapStudent(NZMasSORT + i , NZMasSORT + i + 1 );
printf ("Пункт МУ 19 (SORT - NZMasSORT - после Oklad):\n");
StudPrintMas( NZMasSORT, StudCount);
//
// Сортировка по номеру
    for( int k =0 ; k< StudCount - 1 ; k++)
        for ( int i =0 ; i< StudCount - 1 ; i++ )
            if ( (NZMasSORT + i)->Num < (NZMasSORT + i + 1)->Num ) //
Убывание
                SwapStudent(NZMasSORT + i , NZMasSORT + i + 1 );
printf ("Пункт МУ 19 (SORT - NZMasSORT - после Num):\n");
StudPrintMas( NZMasSORT, StudCount);

```



```

//
// Сортировка по имени
for( int k =0 ; k< StudCount - 1 ; k++)
    for ( int i =0 ; i< StudCount - 1 ; i++ )
        if ( strcmp((NZMasSORT + i)->Name , (NZMasSORT + i + 1)->Name ) > 0 ) //
Возрастание
                SwapStudent(NZMasSORT + i , NZMasSORT + i + 1 );
printf ("Пункт МУ 19 (SORT - NZMasSORT - после Name):\n");
StudPrintMas( NZMasSORT, StudCount);
//
//////////
// МУ п.22.1 чтение одной записи 2-й (0 - 4 )
////////// (СИНИЙ)
int nFind = 2 ; // Читаем 2-ю запись считая с 1 - й
int Pos1;
// Student SBuf; // Буферная структура для чтения
//
printf ("Пункт МУ 22: Распечатка файла\n");
StudPrintFileWR( "BDStud.bin" ); // Предварительная распечатка файла
printf ("Пункт МУ 22:WR - ЧТЕНИЕ - 2-я\n");
pFN = fopen( "BDStud.bin" , "rb"); // Открытие файла для чтения
// Перемещение указателя
if (filelength(pFN->_file)/sizeof(Student) >= nFind )
{
    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 (2-й)
    PrintStudent(&SBuf);
}
else
    printf( "Ошибка номера записи!- %d !\n" , nFind ); // Ошибка номера

// Закрытие файла
    fclose( pFN );
//////////
////////// ГЛАВНАЯ ПРОГРАММА
// МУ п.22.2 чтение из файла по номеру из функции
////////// (СИНИЙ) first.cpp
///// - Прочитать запись по номеру
Student SFind; // Буфер структура для чтения записи
printf ("Чтение из функции (номер = 1):\n");
FindStudNumWR( "BDStud.bin" , &SFind , 1);
PrintStudent( &SFind );
FindStudNumWR( "BDStud.bin" , &SFind , 20); // Ошибка номера
PrintStudent( &SFind );
//////////
////////// ГЛАВНАЯ ПРОГРАММА
// МУ п.22.3 (СИНИЙ)
////////// Чтение и подсчет непосредственно из файла
FILE * pFS;
double SumOk = 0.0 ; // Пекременная для суммирования стипендии всех студентов из
файла
Student SFromFile; // Структура для чтения из файла
pFS = fopen( "BDStud.bin" , "r+b"); // Открытие файла для чтения
if ( pFS != NULL ) // Проверка открытия файла
{
    while (!feof(pFS) ) // Проверка конца файла
    {
        fread( &SFromFile, sizeof(Student) , 1, pFS); // чтение одной записи
        if ( !feof(pFS) ) // Проверка конца файла для исключения лишнего
суммирования для последней записи
            SumOk = SumOk + SFromFile.Okklad;
    }
};

```

```

        printf ("Фонд зарплаты в вычисленный цикл по файлу = %7.3lf  \n", SumOk );
// Заккрытие файла
        fclose( pFS );
};

//
system(" PAUSE");
///
};

```

15.2. header.h Заголовочный файл проекта

header.h– Заголовочный МОДУЛЬ

```

// Комплексная лабораторная работа по ОП СУЦ
////////// ЗАГОЛОВОЧНЫЙ ФАЙЛ
// МУ п.2 (СИНИЙ)
//////////
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
//    char Date[10]; // Резерв пока в примере нет для краткости
};
/*
////////// ЗАГОЛОВОЧНЫЙ ФАЙЛ
// МУ п.2 (СИНИЙ)
//////////
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
    char Date[10]; // Резерв пока в примере нет для краткости
};
*/
////////// ПРОТОТИПЫ header.h
// МУ п.5 (СИНИЙ)
void PrintStudent(Student * pS);
void PrintOneStudent(Student S);
// МУ п.8 (СИНИЙ)
void StudPrintMas( Student * pMas , int Razm);
/// ПОТОТИП ФУНКЦИИ header.h
// МУ п.13 (СИНИЙ)
void StudMasToFileWR( const char * FileName , Student * pMas , int Razm);
// МУ п.14 (СИНИЙ)
void StudPrintFileWR( const char * FileName);
// МУ п.18 (СИНИЙ) header.h
void SwapStudent ( Student * pA , Student * pB );
// МУ п.22.2 (СИНИЙ) header.h
void FindStudNumWR( const char * FileName , Student * pS , int Numb);

```

first.cpp – ГЛАВНЫЙ МОДУЛЬ (переключатель!!!)

```

#include "header.h"
void main(void)
{
    // // п.
    system(" chcp 1251 > nul");
    // Здесь расположить тексты примеров из МУ
    //////////////////////////////////////
    // 2-й СПОСОБ для демонстрации и занятий

```

```

char SW;

MENU:;
// Вывод меню
system (" CLS");
MENU2:;
printf("1. П. ТЗ 5.1.4. структурная переменная (инициализация)...\n");
printf("2. П. ТЗ 5.1.5. структурная переменная (присваивание)...\n");
printf("3. П. ТЗ 5.1.6 функция печати структурной переменной...\n");
printf("4. П. ТЗ 5.1.7 динамическая структурная переменная...\n");
printf("5. П. ТЗ 5.1.8 массив (инициализация и печать)...\n");
printf("6. П. ТЗ 5.1.9 функция печати массива...\n");
printf("7. П. ТЗ 5.1.11 заполнение файла из массива...\n");
//
printf("8. П. ТЗ 5.1.12 распечатка файла...\n");
printf("a. П. ТЗ 5.1.13 функция заполнения файла на основе массива\n");
printf("b. П. ТЗ 5.1.14 функция распечатки файла\n");
printf("c. П. ТЗ 5.1.15 Свар студентов...\n");
printf("d. П. ТЗ 5.1.16 Сортировка...\n");
printf("f. П. ТЗ 5.1.17 чтение 2-й записи ... \n");
printf("g. П. ТЗ 5.1.18 Интегральные вычисление сумма...\n");
//
//printf("h. П. ТЗ 5.1.** ... \n");
//printf("i. П. ТЗ 5.1.** ... \n");
/////printf("*. П. ТЗ 5.1.*... \n");
// ....
printf("в/у/е. Выход ... \n");
printf("0. Меню ... \n");

// ...
// выбрать нужный пункт меню
printf("Выберете пункт меню:\n");
SW = getchar(); // Ввод нажатой клавиши
// Переключатель

switch ( SW)
{
    case '1':
    {
        printf("Выбор 1 (ТЗ 5.1.4.)!\n");
        // проверяемая программа - после этого пункта завершение проверки и программы !!
        // ТЗ 5.1.4.
        // МУ п.3 (СИНИЙ)
        //////////
        Student Stud1={ "Лаптева" , 1 , 1000.0}; // Инициализация трех полей
        printf( "Stud1: Имя = %-15s Номер = %2d Стипендия = %8.2lf \n\n",
            Stud1.Name , Stud1.Num, Stud1.Oklad );
        SW = getchar(); // Сброс ENTER
        goto MENU2;

    break;
    }
    case '2':
    {
        printf("Выбор 2 (ТЗ 5.1.5.)!\n");
        //проверяемая программа - после этого пункта выход в меню
        //п.5.1.5. ...
        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 );

// system ("CLS");
SW = getchar(); // Сброс ENTER
goto MENU2;
break;
}

    case '3':
    {
printf("Выбор 3 (ТЗ 5.1.6)!\n");
//проверяемая программа - пункта выход в меню с сохранением результата
//ТЗ 5.1.6. ...
Student Stud1 ={ "Лаптева" , 1 , 1000.0}; // Инициализация трех полей
Student Stud2;
Stud2.Num = 2;
Stud2.Oklad = 2000.00;
strcpy(Stud2.Name , "Аксенова");
printf ("Пункт МУ 4:\n");
// Печать функциями структур и через указатель

        printf ("Структуры:\n");
PrintOneStudent(Stud1) ;
PrintOneStudent(Stud2) ;

        printf ("Через указатель:\n");
PrintStudent(&Stud1) ;
PrintStudent(&Stud2) ;

    ///
SW = getchar(); // Сброс ENTER
goto MENU2;
break;
}

    case '4':
    {
printf("Выбор 4 (ТЗ 5.1.7.)!\n");
// проверяемая программа - после этого пункта завершение проверки и программы !!
// ТЗ 5.1.7. Динамическая структура
// МУ п.3 (СИНИЙ)
/////////

Student * StudD =(Student *)malloc (sizeof(Student));
StudD->Num = 3;
StudD->Oklad = 5000.00;
strcpy(StudD->Name , "Петров");
PrintStudent(StudD) ;

SW = getchar(); // Сброс ENTER
goto MENU2;

break;
}

    case '5':
    {
printf("Выбор 5 ! ТЗ 5.1.8\n");
//проверяемая программа - пункта выход в меню с сохранением результата
//п.ТЗ 5.1.8
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] ) ; // Печать через указатель
    PrintOneStudent(SMas[i]) ; // Печать через структуру
}

    case '6':
    {
printf("Выбор 6 ! Т3 5.1.9\n");
//проверяемая программа - пункта выход в меню с сохранением результата
//п.Т3 5.1.8
    Student SMas[] ={{"Первый 2" , 1 , 1000.0}, {"Второй 2" , 2 , 2000.0}, {"Третий 2"
, 3 , 3000.0},
    {"Четвертый 2" , 4 , 4000.0}};
    int RazmS = sizeof(SMas)/ sizeof(Student); // Вычисление размерности массива
    // печать функцией
    StudPrintMas( SMas , RazmS);

SW = getchar(); // Сброс ENTER
goto MENU2;
break;
    }

        case '7':
        {
printf("Выбор 7!\n");
//проверяемая программа - пункта выход в меню с сохранением результата
//п.Т3 заполнение файла
    Student ZNMas[] ={{"Запись 1N" , 10 , 11000.0}, {"Запись 2N" , 20 , 12000.0},
{"Запись 3N" , 30 , 13000.0},
    {"Запись 4N" , 40 , 14000.0}};
    int RazmZ = sizeof(ZNMas)/ sizeof(Student); // Вычисление размерности массива
    printf ("Пункт МУ 11(WR исходный массив):\n");
    StudPrintMas( ZNMas , RazmZ); // Распечатка массива
    FILE * pFN;
    pFN = fopen( "BDStud.bin", "w+b"); // Открытие файла для записи
    for ( int i = 0 ; i < RazmZ; i++)
        fwrite( &ZNMas[i], sizeof(Student) , 1, pFN); // Запись в файл
    fclose(pFN); // Закрытие файла

SW = getchar(); // Сброс ENTER
goto MENU2;
break;
    }

    ///////////////
    //добавлять перед этой строкой
        case '8':
        {
printf("Выбор 8!\n");
//распечатка и запоминание в массиве файла
//п.Т3 ...
    Student MASStudWR[100]; // Массив для записи с запасом
    FILE * pFN;
    pFN = fopen( "BDStud.bin" , "r+b"); // Открытие файла для чтения
    Student SBuf1;
    int i = 0;
    printf ("Пункт МУ 11(WR чтение и распечатка файла):\n");
    int RazmS; // для Вычисления размерности массива
    while (!feof(pFN) && i < 100) // Проверка конца файла или размера массива
    {
        fread( &SBuf1, sizeof(Student) , 1, pFN); // чтение одной записи
    }

```

```

if ( !feof(pFN) )
{
    memcpy( &MASstudWR[i] , & SBuf1, sizeof(Student));
    PrintStudent(&SBuf1); // Распечатка одной записи

};
RazmS = i; // запомним размер массива
i++;
}
// печать функцией
printf (" распечатка массива сформированного из файла):\n");

    StudPrintMas( MASstudWR , RazmS);
fclose(pFN); // Закрытие файла
SW = getchar(); // Сброс ENTER
goto MENU2;
break;
    }

        case 'a':
        {
printf("Выбор a!\n");
//проверяемая программа - пункта выход в меню с сохранением результата
//п.ТЗ функция заполнения файла на основе массива
Student NZMas[] ={{"Запись 1D" , 1 , 1000.0}, {"Запись 2D" , 2 , 5000.0}, {"Запись
3D" , 3 , 3000.0},
{"Запись 4D" , 4 , 4000.0}};
int RazmZ = sizeof(NZMas)/ sizeof(Student); // Вычисление размерности массива
// Изменение записей в массиве второй записи(индекс = 1): номер и имя
printf ("Пункт МУ 13:WR\n");
StudMasToFileWR( "BDStud2.bin" , NZMas, RazmZ); // Массив записываем в файл!!!!
StudPrintFileWR( "BDStud2.bin" );

SW = getchar(); // Сброс ENTER
goto MENU2;
break;
    }

        case 'b':
        {
printf("Выбор b!\n");
//проверяемая программа - пункта выход в меню с сохранением результата

StudPrintFileWR( "BDStud.bin" );
SW = getchar(); // Сброс ENTER
goto MENU2;
break;
    }

        case 'c':
        {
printf("Выбор c!\n");
//проверяемая программа - пункта выход в меню с сохранением результата
//п.ТЗ Swap
printf ("Пункт МУ 18 (Swap - до):\n");
Student StudA ={"Лаптева" , 1 , 1000.0};
Student StudB ={"Иванова" , 2 , 2000.0};
PrintStudent(&StudA) ;
PrintStudent(&StudB) ;
printf ("Пункт МУ 18 (Swap - после):\n");
SwapStudent( &StudA , &StudB); // Обмен двух отдельных записей
PrintStudent(&StudA) ;
PrintStudent(&StudB) ;
SW = getchar(); // Сброс ENTER
goto MENU2;

```

```

break;
    }

    case 'd':
    {
printf("Выбор d!\n");
//проверяемая программа - пункта выход в меню с сохранением результата
//п.ТЗ ...Сортировка массива
//////////
Student NZMasSORT[] ={{"Запись 1" , 1 , 1000.0}, {"Запись 2" , 2 , 5000.0}, {"Запись
3" , 3 , 3000.0},
{"Запись 4" , 4 , 4000.0}};
int StudCount;
StudCount= sizeof(NZMasSORT)/ sizeof(Student);
printf ("Пункт МУ 19 (SORT - NZMasSORT - до):Возрастание\n");
StudPrintMas( NZMasSORT, StudCount);
// Сортировка по окладу
for( int k =0 ; k< StudCount - 1 ; k++)
    for ( int i =0 ; i< StudCount - 1 ; i++ )
// Сортировка по окладу
        if ( (NZMasSORT + i)->Oklad > (NZMasSORT + i + 1)->Oklad ) //
Возрастание
            SwapStudent(NZMasSORT + i , NZMasSORT + i + 1 );
printf ("Пункт МУ 19 (SORT - NZMasSORT - после Oklad):Возрастание\n");
StudPrintMas( NZMasSORT, StudCount);
//
// Сортировка по номеру
for( int k =0 ; k< StudCount - 1 ; k++)
    for ( int i =0 ; i< StudCount - 1 ; i++ )
        if ( (NZMasSORT + i)->Num < (NZMasSORT + i + 1)->Num ) //
Убывание
            SwapStudent(NZMasSORT + i , NZMasSORT + i + 1 );
printf ("Пункт МУ 19 (SORT - NZMasSORT - после Num):Убывание\n");
StudPrintMas( NZMasSORT, StudCount);
//
// Сортировка по имени
for( int k =0 ; k< StudCount - 1 ; k++)
    for ( int i =0 ; i< StudCount - 1 ; i++ )
        if ( strcmp((NZMasSORT + i)->Name , (NZMasSORT + i + 1)->Name ) > 0 ) //
Возрастание
            SwapStudent(NZMasSORT + i , NZMasSORT + i + 1 );
printf ("Пункт МУ 19 (SORT - NZMasSORT - после Name):Возрастание\n");
StudPrintMas( NZMasSORT, StudCount);
//////////

//////////
SW = getchar(); // Сброс ENTER
goto MENU2;
break;
    }

    case 'f':
    {
printf("Выбор f!\n");
//проверяемая программа - пункта выход в меню с сохранением результата
//п.ТЗ чтение 2-й записи...
//////////
////////// first.cpp
// МУ п.22.1 чтение одной записи 2-й (0 - 4 )
////////// (СИНИЙ)
int nFind = 2 ; // Читаем 2-ю запись считая ч 1 - й
int Pos1;
Student SBuf; // Буферная структура для чтения
//
StudPrintFileWR( "BDStud.bin" ); // Предварительная распечатка файла

```

```

printf ("Пункт МУ 22:WR - ЧТЕНИЕ - 2-я\n");
FILE * pFN = fopen( "BDStud.bin" , "rb"); // Открытие файла для чтения
// Перемещение указателя
if (filelength(pFN->_file)/sizeof(Student) >= nFind )
{
    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 (2-й)
    PrintStudent(&SBuf);
}
else
    printf( "Ошибка номера записи!- %d !\n" , nFind ); // Ошибка номера

// Заккрытие файла
    fclose( pFN );
//////////

//////////
SW = getchar(); // Сброс ENTER
goto MENU2;
break;
    }

                                case 'g':
    {
        printf("Выбор g!\n");
        //проверяемая программа - пункта выход в меню с сохранением результата
        //п.ТЗ ...Интегральные вычисление сумма окладов
        //////////
        // МУ п.22.3 (СИНИЙ)
        StudPrintFileWR( "BDStud.bin" ); // Предварительная распечатка файла
        ////////// Чтение и подсчет непосредственно из файла
        FILE * pFS;
        double SumOk = 0.0 ; // Переменная для суммирования стипендии всех студентов из файла
        Student SFromFile; // Структура для чтения из файла
        pFS = fopen( "BDStud.bin" , "r+b"); // Открытие файла для чтения
        if ( pFS != NULL ) // Проверка открытия файла
        {
            while (!feof(pFS) ) // Проверка конца файла
            {
                fread( &SFromFile, sizeof(Student) , 1, pFS); // чтение одной записи
                if ( !feof(pFS) ) // Проверка конца файла для исключения лишнего
                суммирования для последней записи
                    SumOk = SumOk + SFromFile.Oklad;
            };
            printf ("Фонд зарплаты в вычисленный цикл по файлу = %7.3lf \n", SumOk );
            // Заккрытие файла
            fclose( pFS );
        };

        //////////
        SW = getchar(); // Сброс ENTER
        goto MENU2;
        break;
    }

    //////////// добавлять перед этой строкой

    /// Шаблоны для case
    /*
    // Шаблон без очистки результата !!!!! убрать комментарии и заменить *
    // начало *****
    case '*':
    {

```



```

printf("Выбор *! ТЗ 5.1.*\n");
//проверяемая программа - пункта выход в меню без сохранения результата
//п.ТЗ ...

SW = getchar(); // Сброс ENTER
goto MENU2;
break;

    }
    // конец *****
    */

    /*
    // Шаблон с очисткой результата !!!!! убрать комментарии и заменить *
    // начало *****
    case '*':

    {
printf("Выбор *!\n");
//проверяемая программа - пункта выход в меню с сохранением результата
//п.ТЗ ...

SW = getchar(); // Сброс ENTER
goto MENU2;
break;

    }

    */
    // убрать комментарии
    // конец *****
    //////////////////////////////////////////
    case '0':
    {
printf("menu!\n");
// После этого пункта очистка и возврат в меню
SW = getchar(); // Сброс ENTER
goto MENU;
break;
break;
    }
    case 'y':
    case 'e':
    case 'в':
    {
printf("Выбор в !\n");
// После этого пункта выход из программы
exit (0);
    }
    default:
    {
printf("Выбор по умолчанию!!\n");
    }

    };

system(" PAUSE");
}

```

second.cpp Второй файл проекта

second.cpp – Второй файл

```

// Комплексная лабораторная работа по ОП СУЦ
#include <math.h>
#include <process.h>
#include <stdio.h>
#include <string.h>
#include <io.h>
#include "header.h"

////////// РЕАЛИЗАЦИЯ ФУНКЦИИ
// МУ п.5 (СИНИЙ)
// Распечатка отдельной структурной переменной студента
////////// Структура
void PrintOneStudent(Student S)
{
    printf( "Запись: Имя = %-15s Номер = %2d Стипендия = %8.2lf \n",
        S.Name , S.Num, S.Oklad );

};
//////////
////////// Указатель
void PrintStudent(Student * pS)
{
    printf( "Запись: Имя = %-15s Номер = %2d Стипендия = %8.2lf \n",
        pS->Name , pS->Num, pS->Oklad );

};
////////// РЕАЛИЗАЦИЯ ФУНКЦИИ
////////// ФАЙЛ second.cpp
// МУ п.8 (СИНИЙ)
//////////
void StudPrintMas( Student * pMas , int Razm)
{
    for (int i = 0 ; i < Razm ; i++ )
    {
        PrintStudent(pMas + i); // Используется готовая функция
    };
};
////////// ФАЙЛ second.cpp
////////// РЕАЛИЗАЦИЯ ФУНКЦИИ
//Перезапись массива структур в файл (Файл очищается!)
// МУ п.13 (СИНИЙ)
//////////
//Перезапись массива структур в файл (Файл предварительно очищается!) WR
///// Вариант функций ( fopen, fclose , fwrite, fread и feof)
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);
};
////////// РЕАЛИЗАЦИЯ ФУНКЦИИ second.cpp
// МУ п.14 (СИНИЙ)
//////////
// Печать файла для структуры Student
///// Вариант функций ( fopen, fclose , fwrite, fread и feof)
void StudPrintFileWR( const char * FileName)
{
    FILE * pF;
    pF = fopen( FileName , "r+b"); // Открытие файла для чтения
    Student SBuf;
    while ( !feof(pF) ) // Проверка конца файла

```

```

{
    fread( &SBuf, sizeof(Student) , 1, pF); // чтение одной записи
    if ( !feof(pF) )
        PrintStudent(&SBuf);
    };
// Заккрытие файла
    fclose( pF );
};

////////// РЕАЛИЗАЦИЯ ФУНКЦИИ
// МУ п.18 (СИНИЙ) second.cpp
//////////
#include <memory.h> // Библиотека для memсpy
// Обмен на основе копирования структурных переменных (так можно только без
динамики!!)
void SwapStudent ( Student * pA , Student * pB )
{
    Student Temp;
    memсpy( &Temp , pA, sizeof(Student) );    // Обмен копированием в памяти
    memсpy( pA , pB, sizeof(Student) );
    memсpy( pB , &Temp, sizeof(Student) );    };

////////// РЕАЛИЗАЦИЯ ФУНКЦИИ
//////////
// Поиск и выборка одной записи по номеру
// МУ п.22.2 (СИНИЙ) second.cpp
//////////
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
        pS->Num = 0; // Очистка полей структуры
        pS->Oklad = 0.0;
        strcpy(pS->Name , "");
    };
// Заккрытие файла
    fclose( pF );
};

```

15.3. Результаты работы главного модуля примеров

ДЗ по ОП №!!!

Stud1: Имя = Лаптева Номер = 1 Стипендия = 1000.00

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

Запись: Имя = Лаптева Номер = 1 Стипендия = 1000.00

Запись: Имя = Аксенова Номер = 2 Стипендия = 2000.00

Пункт МУ 7:

Запись: Имя = Первый Номер = 1 Стипендия = 1000.00

Запись: Имя = Первый Номер = 1 Стипендия = 1000.00

Запись: Имя = Второй Номер = 2 Стипендия = 2000.00

Запись: Имя = Второй Номер = 2 Стипендия = 2000.00

Запись: Имя = Третий Номер = 3 Стипендия = 3000.00

Запись: Имя = Третий Номер = 3 Стипендия = 3000.00

Запись: Имя = Четвертый Номер = 4 Стипендия = 4000.00

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 11(WR исходный массив):

Запись: Имя = Запись 1N Номер = 10 Стипендия = 11000.00

Запись: Имя = Запись 2N Номер = 20 Стипендия = 12000.00

Запись: Имя = Запись 3N Номер = 30 Стипендия = 13000.00

Запись: Имя = Запись 4N Номер = 40 Стипендия = 14000.00

Пункт МУ 11(WR чтение и распечатка файла):

Запись: Имя = Запись 1N Номер = 10 Стипендия = 11000.00

Запись: Имя = Запись 2N Номер = 20 Стипендия = 12000.00

Запись: Имя = Запись 3N Номер = 30 Стипендия = 13000.00

Запись: Имя = Запись 4N Номер = 40 Стипендия = 14000.00

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

Запись: Имя = Запись 1N Номер = 10 Стипендия = 11000.00

Запись: Имя = Запись 2N Номер = 20 Стипендия = 12000.00

Запись: Имя = Запись 3N Номер = 30 Стипендия = 13000.00

Запись: Имя = Запись 4N Номер = 40 Стипендия = 14000.00

Пункт МУ 13:WR

Пункт МУ 14:WR

Запись: Имя = Запись 1 Номер = 11 Стипендия = 51000.00

Запись: Имя = Изменение 2-й Номер = 5 Стипендия = 55000.00

Запись: Имя = Запись 3 Номер = 13 Стипендия = 53000.00

Запись: Имя = Запись 4 Номер = 14 Стипендия = 54000.00

Пункт МУ 17 снова добавим записи из массива:

Запись: Имя = Запись 1 Номер = 11 Стипендия = 51000.00

Запись: Имя = Изменение 2-й Номер = 5 Стипендия = 55000.00

Запись: Имя = Запись 3 Номер = 13 Стипендия = 53000.00

Запись: Имя = Запись 4 Номер = 14 Стипендия = 54000.00

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 - NZMasSW - до):

Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00

Пункт МУ 18 (Sawp - NZMasSW - после):

Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00
Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00
Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00

Пункт МУ 19 (SORT - NZMasSORT - до):

Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00

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

Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00

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

Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00
Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00

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

Запись: Имя = Запись 1	Номер = 1	Стипендия = 1000.00
Запись: Имя = Запись 2	Номер = 2	Стипендия = 5000.00
Запись: Имя = Запись 3	Номер = 3	Стипендия = 3000.00
Запись: Имя = Запись 4	Номер = 4	Стипендия = 4000.00

Пункт МУ 22: Распечатка файла

Запись: Имя = Запись 1	Номер = 11	Стипендия = 51000.00
Запись: Имя = Изменение 2-й	Номер = 5	Стипендия = 55000.00
Запись: Имя = Запись 3	Номер = 13	Стипендия = 53000.00
Запись: Имя = Запись 4	Номер = 14	Стипендия = 54000.00

Пункт МУ 22:WR - ЧТЕНИЕ - 2-я

Запись: Имя = Изменение 2-й	Номер = 5	Стипендия = 55000.00
-----------------------------	-----------	----------------------

Чтение из функции (номер = 1):

Запись: Имя = Запись 1	Номер = 11	Стипендия = 51000.00
------------------------	------------	----------------------

Ошибка номера записи- 20 !

Запись: Имя =	Номер = 0	Стипендия = 0.00
---------------	-----------	------------------

Фонд зарплаты в вычисленный цикл по файлу = 213000.000

Для продолжения нажмите любую клавишу . . .