

Министерство науки и образования РФ  
Федеральное государственное автономное образовательное  
учреждение высшего профессионального образования  
«Санкт-Петербургский государственный электротехнический  
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»  
(СПбГЭТУ «ЛЭТИ»)  
Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчёт  
по лабораторной работе № 1  
на тему:  
“Структуры”  
по дисциплине “Программирование. Дополнительные главы”

Выполнил: Табаков А.В.

Группа 4306

Принял: к.т.н., доцент Сискович Т.И.

Санкт-Петербург  
2015 г.

## Цель

Получить практические навыки работы со структурами.

## Задание

Написать программу для создания последовательности структур, ввода данных в поля структур, контрольного вывода, обработки полей структур и вывод результата обработки.

## Уточнение задания

В программе должно быть использовано простейшее меню. Выполнение программы должно быть многократным по желанию пользователя. Пользователь вводит данные в информационные поля структур. Условия для обработки – поиск элементов в последовательности структур по значению года производства или количеству струн, вывод результата.

## Описание структуры

```
typedef struct stWood
{
    char Deck[10];    //дерево корпуса
    char Neck[10];    //дерево грифа
} WOOD;

typedef struct stGuitars
{
    char Name[10];    //название
    int Strings;      //кол-во струн
    int Year;         //год производства
    WOOD Wood;        //название дерева
} GUITARS;
```

## Контрольные примеры

Контрольные примеры представлены в Таблице 1.

Таблица 1. Контрольные примеры

№ примера	Исходные данные						Результат					
	Название	Год производства	Кол-во струн	Материал		Strings	Year	Название	Год производства	Кол-во струн	Материал	
				корпус	гриф						корпус	гриф
1	Gibson	1964	6	Ольха	Кедр		1990	Gibson	1964	6	Ольха	Кедр
	Fender	1983	6	Сосна	Клён			Fender	1983	6	Сосна	Клён
	Dean	1991	7	Липа	Клён							
2	Gibson	1964	6	Ольха	Кедр	7		Dean	1991	7	Липа	Клён
	Fender	1983	6	Сосна	Клён							
	Dean	1991	7	Липа	Клён							
3	Gibson	1964	6	Ольха	Кедр		1964	Gibson	1964	6	Ольха	Кедр
	Fender	1983	6	Сосна	Клён							
	Dean	1991	7	Липа	Клён							

## Описание главной функции

Назначение: организация управления порядком вызова функций.

## Описание переменных функции

Описание переменных представлено в Таблице 2.

Таблица 2. Описание переменных главной функции

Имя переменной	Тип	Назначение
mGuitars	Guitars*	Указатель на первый элемент исходной последовательности структур
Count	int	Количество гитар
NewGuitars	Guitars*	Указатель на первый элемент сформированной последовательности структур
NewCount	int	Количество гитар в сформированной выборке
CheckOfProcess	bool	Флаг обработки данных
Q	int	Переменная выбора меню

## Краткое описание алгоритма

Начало программы.

Шаг №1. Вывод меню.

Шаг №2. Выбор пользователем пункта меню.

Шаг №3. Переход к пункту, выбранному пользователем.

1: Справка. Переход к шагу 2

2: Ввод исходных данных. Переход к шагу 2

3: Вывод исходных данных. Переход к шагу 2

4: Обработка данных. Переход к шагу 2

5: Вывод результата. Переход к шагу 2

6: Выход. Переход к шагу 4.

Шаг №4. Конец программы.

## Блок-схема главной функции

Блок-схема главной функции представлена на рисунке 1.

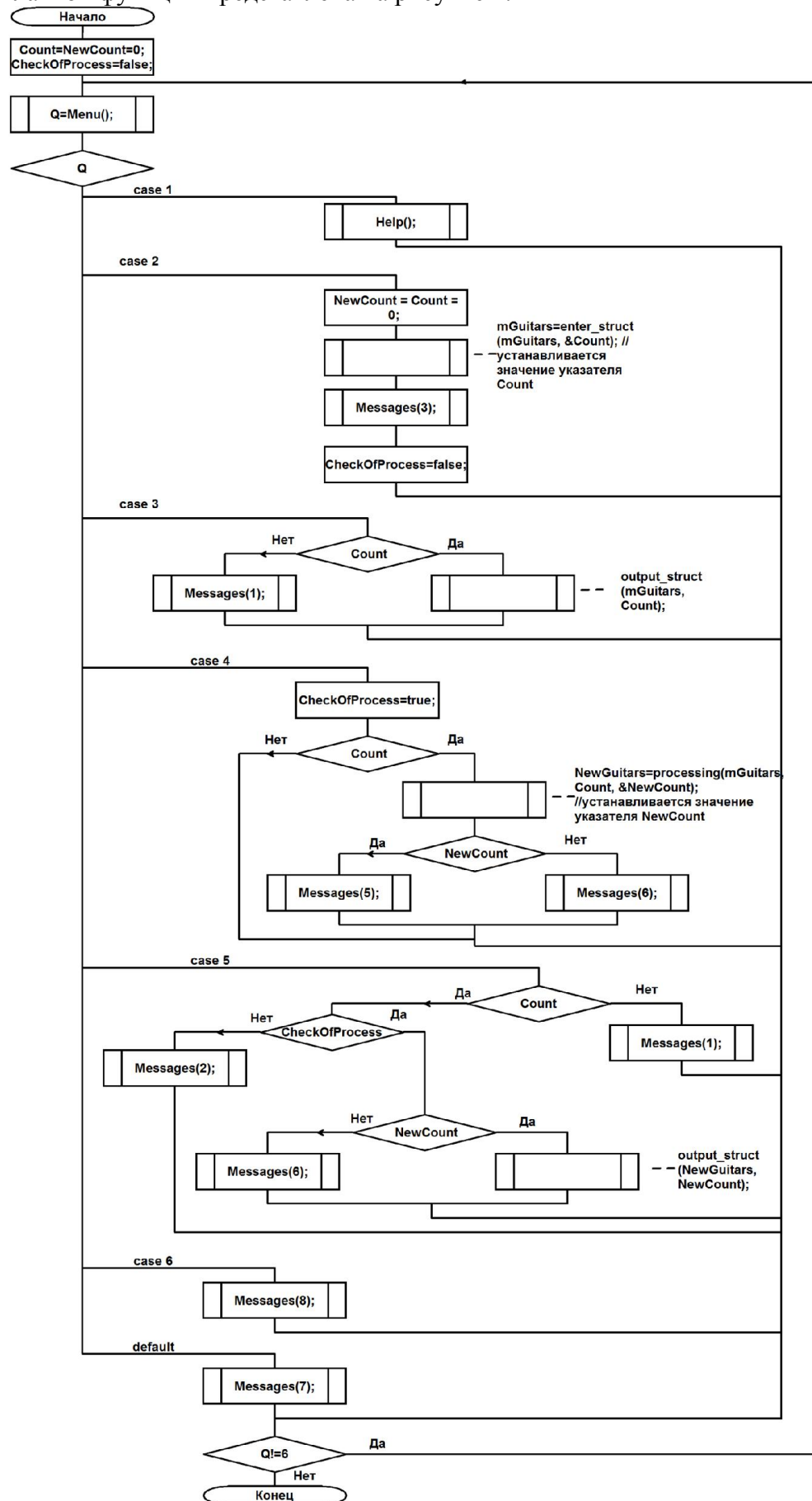


Рис. 1. Блок-схема главной функции

## Описание функций

### Описание функции Help

Назначение: Функция используется для вывода справки.

Прототип: void Help();, где void – тип не возвращаемого значения.

Пример вызова: Help();

Вызывающая функция: main.

### Описание функции Menu

Назначение: Функция используется для вывода меню программы.

Прототип: int Menu();, где int – тип возвращаемого значения.

Пример вызова: Q=Menu();, где Q – переменная выбора меню.

Вызывающая функция: main.

### Описание переменных

Описание переменных функции Menu представлены в Таблице 3.

Таблица 3. Описание переменных функции Menu

Имя переменной	Тип	Назначение
Локальные переменные		
Q	int	Переменная выбора пункта меню

### Описание функции Messages

Назначение: Функция используется для ввода сообщений пользователю.

Прототип: void Messages(int), где int номер вызываемого сообщения; void – тип не возвращаемого значения.

Пример вызова: Messages(1);, где «1» – номер вызываемого сообщения.

Вызывающая функция: main.

Сообщения:

Messages(1): "Сначала необходимо ввести данные."

Messages(2): "Вы ввели данные, но не обработали их."

"Вам необходимо выбрать 3 пункт меню для обработки данных."

Messages(3): "Данные введены успешно."

Messages(4): " Ошибка, данные не были введены."

"Попробуйте ввести данные повторно."

Messages(5): "Выборка из данных успешно сформирована."

Messages(6): " Выборка из данных не была сформирована."

"В исходных данных не нашлось таких результатов."

Messages(7): "Что-то пошло не так, введите пункт меню повторно."

Messages(8): "До новых встреч!"

### Описание переменных

Описание переменных функции Messages представлены в Таблице 4.

Таблица 4. Описание переменных функции Messages

Имя переменной	Тип	Назначение
Формальные переменные		
Q	int	Вспомогательная переменная

## Описание функции enter

Назначение: Функция используется для ввода данных в структуру.

Прототип: Guitars\* enter (Guitars\*, int\*);, где Guitars\*- указатель на структуру, int\* - указатель на кол-во гитар; Guitars\* – тип возвращаемого значения.

Пример вызова: mGuitars=enter (mGuitars, &Count);, где mGuitars указатель на структуру, &Count – адрес кол-ва гитар.

Вызывающая функция: main.

Вызываемая функция: enter\_struct, memory.

## Описание переменных

Описание переменных функции enter представлены в Таблице 5.

Таблица 5. Описание переменных функции enter

Имя переменной	Тип	Назначение
Локальные переменные		
Q	int	Вспомогательная переменная
Формальные переменные		
mGuitars	Guitars*	Указатель на первый элемент исходной последовательности структур
Count	int*	Указатель на кол-во гитар

## Описание функции enter\_struct

Назначение: Функция используется для ввода данных в поля структур.

Прототип: Guitars\* enter\_struct (Guitars\*, int);, где Guitars\*- указатель на структуру, int - кол-во гитар; Guitars\* – тип возвращаемого значения.

Пример вызова: mGuitars= enter\_struct (mGuitars, Count);, где mGuitars указатель на структуру, Count – кол-во гитар.

Вызывающая функция: enter.

## Описание переменных

Описание переменных функции enter\_struct представлены в Таблице 6.

Таблица 6. Описание переменных функции enter\_struct

Имя переменной	Тип	Назначение
Формальные переменные		
mGuitars	Guitars*	Указатель на первый элемент последовательности структур
Count	int	Кол-во гитар

## Описание функции enter\_num

Назначение: Функция используется для ввода данных в поля структур.

Прототип: int enter\_num(char\*, int, int);, где char\*- указатель на строку, int – начальное число, int – конечное число; int – тип возвращаемого значения.

Пример вызова: mGuitars[Count-1].Strings=enter\_num("количество струн", 1, 20);, где "количество струн" - указатель на строку, 1 – начальное число, 20 – конечное число.

Вызывающая функция: enter\_struct.

Вызываемая функция: enter\_num.

## Описание переменных

Описание переменных функции enter\_num представлены в Таблице 7.

Таблица 7. Описание переменных функции enter\_num

Имя переменной	Тип	Назначение
Локальные переменные		
num	int	Вспомогательная переменная

## Описание функции output\_struct

Назначение: Функция используется для вывода структуры.

Прототип: void output\_struct(Guitars\*, int);, где Guitars\* - указатель на структуру, int – кол-во гитар; void - тип не возвращаемого значения.

Пример вызова: output\_struct(mGuitars, Count);, где Guitars\* - указатель на структуру, int – кол-во гитар.

Вызывающая функция: main.

## Описание переменных

Описание переменных функции output\_struct представлены в Таблице 8.

Таблица 8. Описание переменных функции output\_struct

Имя переменной	Тип	Назначение
Локальные переменные		
i	int	Счётчик
Формальные переменные		
mGuitars	Guitars*	Указатель на первый элемент исходной последовательности структур
Count	int	Кол-во гитар

## Описание функции processing

Назначение: Функция используется для обработки данных.

Прототип: Guitars\* processing(Guitars\*, int, int\*, int);, где Guitars\* - указатель на структуру, int - кол-во гитар, int\* - кол-во гитар в выборке; Guitars\* - тип возвращаемого значения.

Пример вызова: NewGuitars=processing(mGuitars, Count, &NewCount);, где NewGuitars – указатель на структуру выборку, mGuitars – указатель на исходную структуру, Count – кол-во гитар в исходных данных, &NewCount - адрес кол-ва гитар выборки.

Вызывающая функция: main.

Вызываемая функция: memory.

## Описание переменных

Описание переменных функции processing представлены в Таблице 9.

Таблица 9. Описание переменных функции processing

Имя переменной	Тип	Назначение
Локальные переменные		
NewGuitars	Guitars*	Указатель на первый элемент сформированной последовательности структур
i	int	Счётчик
temp	int	Вспомогательная переменная
Q	int	Переменная выбора пункта подменю
Формальные переменные		
mGuitars	Guitars*	Указатель на первый элемент исходной последовательности структур
Count	int	Кол-во гитар
NewCount	int*	Указатель на кол-во гитар выборки

## Описание функции processing\_menu

Назначение: Функция используется организации меню обработки.

Прототип: Guitars\* processing(Guitars\*, int, int\*);, где Guitars\* - указатель на структуру, int - кол-во гитар, int\* - кол-во гитар в выборке; Guitars\* - тип возвращаемого значения.

Пример вызова: NewGuitars=processing(mGuitars, Count, &NewCount);, где NewGuitars – указатель на структуру выборку, mGuitars – указатель на исходную структуру, Count – кол-во гитар в исходных данных, &NewCount - адрес кол-ва гитар выборки.

Вызывающая функция: main.

Вызываемая функция: processing.

## Описание переменных

Описание переменных функции processing\_menu представлены в Таблице 10.

Таблица 10. Описание переменных функции processing\_menu

Имя переменной	Тип	Назначение
Локальные переменные		
NewGuitars	Guitars*	Указатель на первый элемент сформированной последовательности структур
i	int	Счётчик
temp	int	Вспомогательная переменная
Q	int	Переменная выбора пункта подменю
Формальные переменные		
mGuitars	Guitars*	Указатель на первый элемент исходной последовательности структур
Count	int	Кол-во гитар
NewCount	int*	Указатель на кол-во гитар выборки

## Описание функции memory

Назначение: Функция используется для выделения памяти под структуру.

Прототип: Guitars\* memory(Guitars\*, int);, где Guitars\* - указатель на структуру, int – кол-во гитар; Guitars\* - тип возвращаемого значения.

Пример вызова: NewGuitars=memory(NewGuitars, TempCount), где NewGuitars - указатель на структуру, TempCount - кол-во гитар; NewGuitars - указатель на структуру.

Вызывающая функция: enter, processing.

## Описание переменных

Описание переменных функции memory представлены в Таблице 11.

Таблица 11. Описание переменных функции memory

Имя переменной	Тип	Назначение
Формальные переменные		
mGuitars	Guitars*	Указатель на первый элемент исходной последовательности структур
Count	int	Кол-во гитар

## Описание функции fr

Назначение: Функция используется для освобождения памяти, выделенной ранее под данные.

Прототип: Guitars\* fr(Guitars\*);, где Guitars\* - указатель на структуру; Guitars\* - тип возвращаемого значения.

Пример вызова: mGuitars =fr(mGuitars);, где mGuitars – указатель на структуру.

Вызывающая функция: main.

Блок схема функции не представлена из-за очевидности выполнения алгоритма.

## Описание переменных

Описание переменных функции fr представлены в Таблице 12.

Таблица 12. Описание переменных функции fr

Имя переменной	Тип	Назначение
Формальные переменные		
mGuitars	Guitars*	Указатель на первый элемент исходной последовательности структур



## Структура вызова функций

Структура вызова функций представлена на рисунке 2.

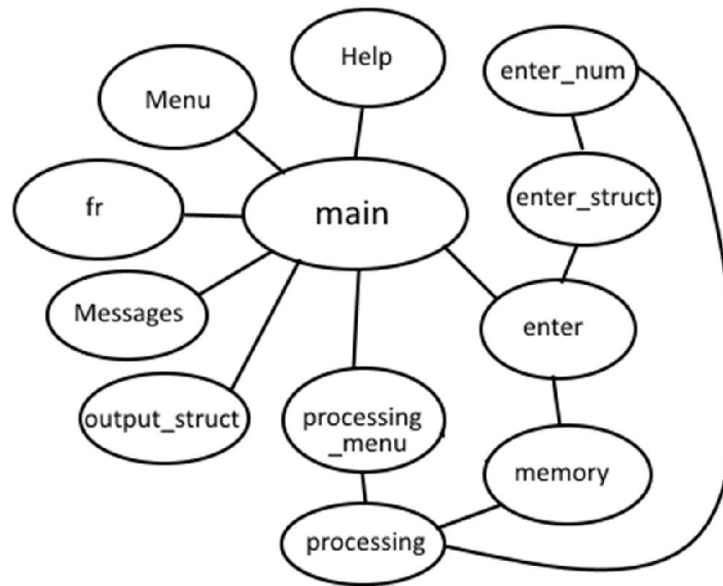


Рис. 2. Структура вызова функций

## Текст программы с комментариями

```
#include <stdio.h>
#include <locale.h>
#include <stdlib.h>
#include <malloc.h>
#include <conio.h>
#include <string.h>
#include <windows.h>

typedef struct stWood
{
    char Deck[10];           //Дерево корпуса
    char Neck[10];          //Дерево грифа
} WOOD;

typedef struct stGuitars
{
    char Name[10];           //Название
    int Strings;             //Кол-во струн
    int Year;                //Год производства
    WOOD Wood;               //Структура дерево
} GUITARS;

void Help();                //Прототип функции справка
int Menu();                 //Прототип функции меню
GUITARS* memory(GUITARS*, int); //Прототип функции выделения памяти
GUITARS* enter(GUITARS*, int*); //Прототип функции ввода данных в структуру
GUITARS* enter_struct(GUITARS*, int); //Прототип функции ввода данных в поля структуры
int enter_num(char*, int, int); //Прототип функции ввода числовых данных
GUITARS* processing(GUITARS*, int, int*, key); //Прототип функции обработки данных
void output_struct(GUITARS*, int); //Прототип функции вывода данных
void Messages(int);         //Прототип функции вывода сообщений
GUITARS* fr(GUITARS*);      //Прототип функции освобождения памяти
GUITARS* processing_menu(GUITARS*, int, int*); //Прототип функции подменю обработки
```

```

int main()
{
    GUITARS *mGuitars=NULL, *NewGuitars=NULL;
    int Q, Count=0, NewCount=0;
    bool CheckOfProcess=false;
    setlocale(LC_ALL, "RUS");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    do
    {
        Q = Menu();
        switch (Q)
        {
            case 1:
                Help();
                break;
            case 2:
                mGuitars=fr(mGuitars);
                NewGuitars=fr(NewGuitars);
                NewCount = Count = 0;
                mGuitars = enter(mGuitars, &Count);
                Messages(3);
                CheckOfProcess=false;
                break;
            case 3:
                if(Count)
                    output_struct(mGuitars, Count);
                else
                    Messages(1);
                break;
            case 4:
                NewGuitars=fr(NewGuitars);
                if(Count)
                {
                    NewCount=0;
                    NewGuitars=fr(NewGuitars);
                    NewGuitars=processing_menu(mGuitars, Count, &NewCount);
                    if(NewCount)
                        Messages(5);
                    else
                        Messages(6);
                }
                else
                    Messages(1);
                break;
            case 5:
                if(Count)
                    if(CheckOfProcess)
                        if(NewCount)
                            output_struct(NewGuitars, NewCount);
                    else
                        Messages(6);
                else
                    Messages(2);
                else
                    Messages(1);
                break;
            case 6:
                Messages(8);
                break;
            default:
                Messages(7);
        }
    }
    while (Q!=6);
    fr(mGuitars);
    fr(NewGuitars);
    return 0;
}

//*****
//Функция справка
void Help()
{

```

```

        system("cls");
        puts("\n\n Данная программа предназначена для организации структуры и обработки данных.");
        puts(" Выборка составляется из гитар до выбранного года производства.");
        puts(" Если возникли проблемы обращайтесь, пожалуйста, на электронную почту:");
        puts(" komdosh@gelezo2.ru\n");
        system("pause");
    }

//*****
//Функция меню
int Menu()
{
    int Q;
    system("cls");
    puts("Главное меню");
    puts("1 - Справка");
    puts("2 - Ввод данных в структуру");
    puts("3 - Вывод исходных данных");
    puts("4 - Обработка данных");
    puts("5 - Вывод результата");
    puts("6 - Выход");
    printf("Введите номер пункта - ");
    scanf("%d", &Q);
    printf("\n");
    fflush(stdin);
    return Q;
}

//*****
//Функция выделения памяти
GUITARS* memory(GUITARS *mGuitars, int Count)
{
    mGuitars=(GUITARS*)realloc(mGuitars, Count*sizeof(GUITARS));
    return mGuitars;
}

//*****
//Функция ввода данных
GUITARS* enter(GUITARS* mGuitars, int *Count)
{
    int Q;
    do
    {
        (*Count)++;
        mGuitars=memory(mGuitars, *Count);
        system("cls");
        mGuitars=enter_struct(mGuitars, *Count);
        puts("Добавить объект? Введите 1 для продолжения, 2 для завершения ввода.");
        scanf("%d", &Q);
        fflush(stdin);
    }
    while(Q!=2);
    return mGuitars;
}

//*****
//Функция ввода данных в поля
GUITARS* enter_struct(GUITARS* mGuitars, int Count)
{
    printf("Введите марку %d гитары (не более 10 символов): ", Count);
    do
    {
        gets(mGuitars[Count-1].Name);
        fflush(stdin);
        if(strlen(mGuitars[Count-1].Name)>10)
            printf("Возможно вы ошиблись при вводе?\n(кол-во символов не больше 10)\nПовторите ввод: ");
    }
    while(strlen(mGuitars[Count-1].Name)>10);
    mGuitars[Count-1].Strings=enter_num("количество струн", 1, 20);
    mGuitars[Count-1].Year=enter_num("год производства", 1899, 2015);
    printf("Введите название дерева грифа (не более 10 символов): ");
    do
    {
        gets(mGuitars[Count-1].Wood.Neck);
        fflush(stdin);
    }
}

```

```

        if(strlen(mGuitars[Count-1].Wood.Neck)>10)
            printf("Возможно вы ошиблись при вводе?\n(кол-во символов не больше 10)\nПовторите ввод: ");
    }
    while(strlen(mGuitars[Count-1].Wood.Neck)>10);
    printf("Введите название дерева корпуса (не более 10 символов): ");
    do
    {
        gets(mGuitars[Count-1].Wood.Deck);
        fflush(stdin);
        if(strlen(mGuitars[Count-1].Wood.Deck)>10)
            printf("Возможно вы ошиблись при вводе?\n(кол-во символов не больше 10)\nПовторите ввод: ");
    }
    while(strlen(mGuitars[Count-1].Wood.Deck)>10);
    return mGuitars;
}

//*****
//Функция ввода целочисленных переменных
int enter_num(char* str, int first, int last)
{
    int num;
    printf("Введите %s (от %d до %d): ", str, first, last);
    do
    {
        scanf("%d", &num);
        fflush(stdin);
        if(num <= first || num > last)
            printf("Возможно вы ошиблись при вводе?\nВведите число от %d до %d\nПовторите ввод: ", first, last);
    }
    while(num <= first || num > last);
    return num;
}

//*****
//Функция подменю обработки
GUITARS* processing_menu(GUITARS *mGuitars, int Count, int *NewCount)
{
    GUITARS *NewGuitars=NULL;
    int Q, temp, i;
    do
    {
        system("cls");
        puts("Меню поиска, выберите по какому пункту сделать выборку");
        puts("1 - Год производства");
        puts("2 - Количество струн");
        printf("Введите номер пункта - ");
        scanf("%d", &Q);
        fflush(stdin);
        switch(Q)
        {
            case 1:
                NewGuitars=processing(mGuitars, Count, NewCount, 1);
                break;
            case 2:
                NewGuitars=processing(mGuitars, Count, NewCount, 2);
                break;
            default:
                Messages(7);
        }
    }
    while(Q!=1 && Q!=2);
    return NewGuitars;
}

//*****
//Функция обработки данных
GUITARS* processing(GUITARS *mGuitars, int Count, int *NewCount, int key)
{
    GUITARS *NewGuitars=NULL;
    int Q, temp, i;
    if(key==1)
    {
        temp=enter_num("до какого года производства выводить результаты", 1899, 2015);
    }
}

```

```

        for(i=0; i<Count; i++)
        if(mGuitars[i].Year<=temp)
        {
            (*NewCount)++;
            NewGuitars=memory(NewGuitars, *NewCount);
            NewGuitars[*NewCount-1]=mGuitars[i];
        }
    }
    else
    {
        temp=enter_num("до сколько струн выводить результаты", 0, 20);
        for(i=0; i<Count; i++)
            if(mGuitars[i].Strings==temp)
            {
                (*NewCount)++;
                NewGuitars=memory(NewGuitars, *NewCount);
                NewGuitars[*NewCount-1]=mGuitars[i];
            }
    }
    return NewGuitars;
}

//*****
//Функция вывода данных
void output_struct(GUITARS *mGuitars, int Count)
{
    system("cls");
    printf("=====");
    printf("%12s | %18s | %14s | %17s\n", " ", " ", " ", "Дерево:");
    printf("%12s | %18s | %14s | %s\n", "Название", "Год производства", "Кол-во струн", " ");
    printf("%12s | %18s | %14s | %11s | %6s\n", " ", " ", " ", "Корпус", "Гриф");
    printf("=====");
    for(int i=0; i<Count; i++)
    {
        printf("%12s | %18d | %14d | %11s | %6s", mGuitars[i].Name, mGuitars[i].Year, mGuitars[i].Strings,
            mGuitars[i].Wood.Deck, mGuitars[i].Wood.Neck);
        printf("\n=====");
    }
    system("pause");
}

//*****
//Функция вывода сообщений пользователю
void Messages(int Q)
{
    system("cls");
    switch(Q)
    {
        case 1:
            puts("Сначала необходимо ввести данные.");
            break;
        case 2:
            puts("Вы ввели данные, но не обработали их.");
            puts("Вам необходимо выбрать 4 пункт меню для обработки данных.");
            break;
        case 3:
            puts("Данные введены успешно.");
            break;
        case 4:
            puts("Ошибка, данные не были введены.");
            puts("Попробуйте ввести данные повторно.");
            break;
        case 5:
            puts("Выборка из данных успешно сформирована.");
            break;
        case 6:
            puts("Выборка из данных не была сформирована.");
            puts("В исходных данных не нашлось таких результатов.");
            break;
        case 7:
            puts("Что-то пошло не так, введите пункт меню повторно.");
    }
}

```

```

        break;
    case 8:
        puts("До новых встреч!");
        break;
    }
    system("pause");
}

//*****
//Функция освобождения памяти
GUITARS* fr(GUITARS* mGuitars)
{
    if (mGuitars!=NULL)
        free(mGuitars);
    return NULL;
}

```

### **Результаты решения задачи**

При выполнении программы были получены результаты, совпадающие со значениями, приведенными в Таблице 1. Ошибок не обнаружено.

### **Вывод**

При выполнении лабораторной работы были получены практические навыки работы со структурами на языке программирования «C/C++».