



МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА – Российский технологический университет»  
РТУ МИРЭА

---

---

Отчет по выполнению практического задания №1  
**Тема:** Стандартные типы данных для представления многоэлементных  
не однородных структур данных  
Дисциплина: Структуры и алгоритмы обработки данных

Выполнил студент Иолович Е.А

Фамилия И.О.

группа

ИНБО-03-22

Москва 2023

## СОДЕРЖАНИЕ

1 ОТЧЕТ ПО ЗАДАНИЮ 1 .....	3
1.1 Постановка задачи.....	3
1.2 Модель решения.....	4
1.3 Код основной задачи.....	7
2 ОТЧЕТ ПО ЗАДАНИЮ 2.....	10
2.1 Постановка задачи.....	10
2.2 Определение структуры записи и структуры таблицы средствами языка программирования. ....	10
2.3 Код основной задачи.....	10
3 ОТЧЕТ ПО ЗАДАНИЮ 3.....	14
3.1 Код основной задачи.....	14
4 ЗАКЛЮЧЕНИЕ .....	17
5 СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ .....	18

# 1 ОТЧЕТ ПО ЗАДАНИЮ 1

## 1.1 Постановка задачи

Разработать набор операций для управления таблицей, созданной на основе статического массива. Создать проект, добавить заголовочный файл. Разработать тип, определяющий структуру элемента таблицы (записи). Определить на базе типа struct структуру хранения данных по таблице, содержащую: текущий размер таблицы -  $n$ , максимальный размер таблицы  $N$ , массив из  $N$  элементов типа записи варианта. Разработать алгоритм и реализовать функцию, выполнения вывода таблицы на экран. Для заполнения отдельной записи с клавиатуры разработать функцию, которая принимает в качестве параметра пустую запись, а возвращает заполненную. Наполнение таблицы данными должна выполнять функция вставки или добавления записи, включенные в список операций варианта. При вставке и добавлении записей в таблицу корректировать значение  $n$ .

Вариант индивидуального задания – 10.

Записная книжка друзей:

Структура элемента (записи): Фамилия, Имя, Номер телефона, Дата рождения (три значения), Адрес.

Операции:

1) Вставить новую запись в таблицу так, чтобы она стала первой в подсписке фамилий записей таблицы, которые начинаются с той же буквы, что и фамилия во вставляемой записи. Если нет таких фамилий, то добавить запись в таблицу.

2) Удалить запись по заданному номеру телефона.

3) Сформировать список друзей, родившихся в заданном месяце (по номеру месяца).

4) Найти дату рождения и номер телефона друга по его Фамилии.

## 1.2 Модель решения

Изображение рисунком (рис. 1) структуры записи. Указаны названия полей.

```
struct human
  string name
  string surname
  string phone
  int dataD
  int dataM
  int dataY
  string address
```

Рисунок 1 – Структура записи

Изображение рисунком (рис. 2) структуры таблицы. Указаны названия полей

```
struct table
  int n
  human a[N]
```

Рисунок 2 – Структура таблицы

Определение структуры записи и структуры таблицы средствами языка программирования (рис. 3 и рис. 4).

```

struct human
{
    string name;
    string surname;
    string phone;
    int dataD;
    int dataM;
    int dataY;
    string address;
}

```

Рисунок 3 – Структура записи

```

struct table
{
    #define N 100 //глобальная
    int n;
    human a[N];
}

```

Рисунок 4 – Структура таблицы

Список подзадач задачи, которые предстоит реализовать как функции:

- Добавление информации о каждом друге (промежуточное действие).
- Вставка записи о новом человеке в список, если первая буква его фамилии совпала с введенными данными пользователя.
- Удаление записи по номеру телефона.
- Создание списка друзей, чем месяц совпал с введенными месяцем пользователя.
- Вывод даты рождения и телефона при вводе пользователем искомой фамилии.

Прототипы функций выделенных подзадач:

```
void print()
```

Рисунок 5 – Функция заполнения информации о друге (промежуточное действие)

```
void insert(human temp)
```

Рисунок 6 – Функция добавления друга в список по первой букве фамилии

```
void Delete(string tel)
```

Рисунок 7 – Функция удаления по номеру телефона

```
void List(int M)
```

Рисунок 8 – Функция создания списка по введённому месяцу

```
void search(string sur)
```

Рисунок 9 – Функция поиска даты рождения и номера телефона по фамилии

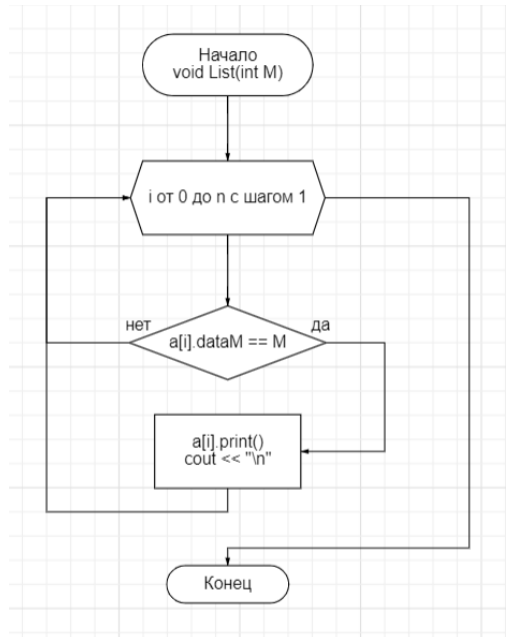


Рисунок 10 – Блок-схема функции создания списка по введённому месяцу

Скриншоты тестирования программы:

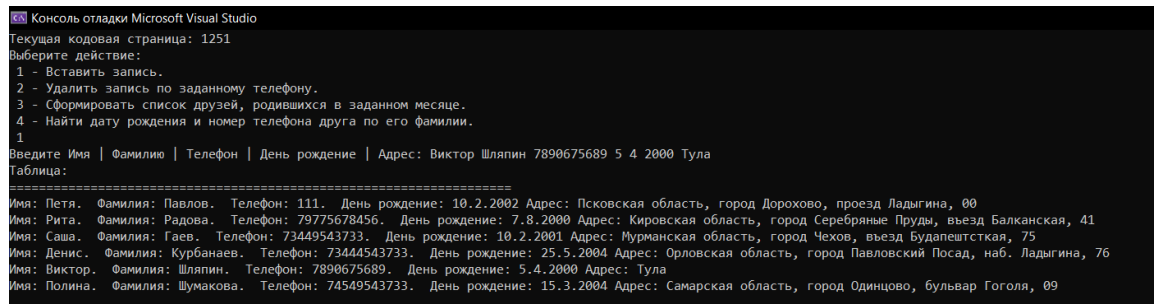


Рисунок 11 – Функция добавления друга в список по первой букве фамилии

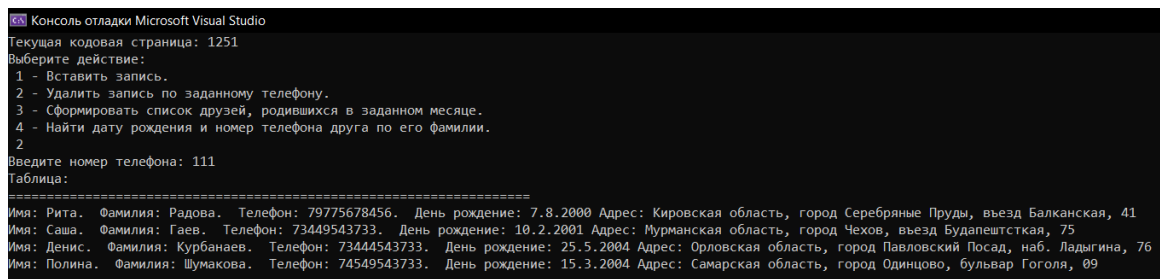


Рисунок 12 – Функция удаления по номеру телефона

```

Консоль отладки Microsoft Visual Studio
Текущая кодовая страница: 1251
Выберите действие:
1 - Вставить запись.
2 - Удалить запись по заданному телефону.
3 - Сформировать список друзей, родившихся в заданном месяце.
4 - Найти дату рождения и номер телефона друга по его фамилии.
3
Введите номер месяца: 2
Имя: Петя. Фамилия: Павлов. Телефон: 111. День рождения: 10.2.2002 Адрес: Псковская область, город Дорохово, проезд Ладыгина, 00
Имя: Саша. Фамилия: Гаев. Телефон: 73449543733. День рождения: 10.2.2001 Адрес: Мурманская область, город Чехов, въезд Будапештская, 75

```

Рисунок 13 – Функция создания списка по введённому месяцу

```

Консоль отладки Microsoft Visual Studio
Текущая кодовая страница: 1251
Выберите действие:
1 - Вставить запись.
2 - Удалить запись по заданному телефону.
3 - Сформировать список друзей, родившихся в заданном месяце.
4 - Найти дату рождения и номер телефона друга по его фамилии.
4
Введите искомую фамилию: Шумакова
День: 15 Месяц: 3 Год: 2004 Телефон: 74549543733

```

Рисунок 14 – Функция поиска даты рождения и номера телефона по фамилии

### 1.3 Код основной задачи

```

#include <iostream> //статич. массив.
#include <string>;
using namespace std;

struct human
{
    string name;
    string surname;
    string phone;
    int dataD;
    int dataM;
    int dataY;
    string address;

    void print()
    {
        cout << "Имя: " << name << ". " << "Фамилия: " << surname << ". " << "Телефон: " << phone << ". " << "День
рождения: " << dataD << "." << dataM << "." << dataY << " " << "Адрес: " << address;
    }
};

struct table
{
#define N 100 //глобальная переменная, константа.
    int n;
    human a[N];

    void print() //цикл для вывода списка друзей.
    {
        cout << "Таблица: " << "\n" <<
"===== " << "\n";
        for (int i = 0; i < n; i++)
        {
            a[i].print();
            cout << "\n";
        }
    }

    human add() //добавление информации о каждом друге.
    {
        human B;
        setlocale(LC_ALL, "Russian");
        cout << "Введите Имя | Фамилию | Телефон | День рождения | Адрес: ";
        cin >> B.name >> B.surname >> B.phone >> B.dataD >> B.dataM >> B.dataY >> B.address;
        return B;
    }
};

```

```

}

void insert(human temp) //фун-ция для поиска и вставки записи о новом человеке, если первая буква его фамилии
совпала с имеющимися данными, функ-я добавления записи.
{
    int k = -1; //флаг

    for (int i = 0; i < n; i++)
    {
        if (a[i].surname[0] == temp.surname[0])
        {
            k = i;
            break;
        }
    }

    if (k != -1)
    {
        for (int i = n; i > k; i--)
        {
            a[i] = a[i - 1];
        }
        a[k] = temp;
    }
    else
    {
        a[n] = temp;
    }
    n = n + 1;
}

```

```

void Delete(string tel) //фун-ция для удаления записи по номеру телефона.
{
    int k = -1; //флаг.

    for (int i = 0; i < n; i++)
    {
        if (a[i].phone == tel)
        {
            k = i;
            break;
        }
    }
    if (k != -1)
    {
        for (int i = k; i < n; i++)
        {
            a[i] = a[i + 1];
        }
        n = n - 1;
    }
    else
    {
        cout << "error";
    }
}

```

```

}

void List(int M) //фун-ция для создания списка друзей, чем месяц рождения совпал.
{
    for (int i = 0; i < n; i++)
    {
        if (a[i].dataM == M)
        {
            a[i].print();
            cout << "\n";
        }
    }
}

```

```

void search(string sur) //фун-ция по выводу дня\месяца\телефона искомого человека при вводе фамилии.
{

```



```

        for (int i = 0; i < n; i++)
        {
            if (a[i].surname == sur)
            {
                cout << "День: " << a[i].dataD << " " << "Месяц: " << a[i].dataM << " " << "Год: " <<
a[i].dataY << " " << "Телефон: " << a[i].phone;
                cout << "\n";
            }
        }
    };

void userchoice(table t) //функция для оформления консоли выбора.
{
    setlocale(LC_ALL, "Russian");
    int choice;
    cout << "Выберите действие: \n 1 - Вставить запись. \n 2 - Удалить запись по заданному телефону. \n 3 -
Сформировать список друзей, родившихся в заданном месяце. \n 4 - Найти дату рождения и номер телефона друга по его фамилии. \n ";
    cin >> choice;

    string numberV;
    int mounth;
    string secondname;

    switch (choice)
    {
        case 1:
            t.insert(t.add());
            t.print();
            break;
        case 2:
            cout << "Введите номер телефона: ";
            cin >> numberV;
            t.Delete(numberV);
            t.print();
            break;
        case 3:
            cout << "Введите номер месяца: ";
            cin >> mounth;
            t.List(mounth);
            break;
            t.print();
        case 4:
            cout << "Введите искомую фамилию: ";
            cin >> secondname;
            t.search(secondname);
            break;
    }
}

int main()
{
    table t;
    setlocale(LC_ALL, "Russian");
    system("chcp 1251");
    t.n = 5; //база данных для изначального списка друзей.
    t.a[0] = { "Петя", "Павлов", "111", 10, 2, 2002, "Псковская область, город Дорохово, проезд Ладыгина, 00" };
    t.a[1] = { "Рита", "Радова", "79775678456", 7, 8, 2000, "Кировская область, город Серебряные Пруды, въезд
Балканская, 41" };
    t.a[2] = { "Саша", "Гаев", "73449543733", 10, 2, 2001, "Мурманская область, город Чехов, въезд Будапештская, 75" };
    t.a[3] = { "Денис", "Курбанаев", "73444543733", 25, 5, 2004, "Орловская область, город Павловский Посад, наб.
Ладыгина, 76" };
    t.a[4] = { "Полина", "Шумакова", "74549543733", 15, 3, 2004, "Самарская область, город Одинцово, бульвар Гоголя,
09" };
    userchoice(t);
}

```

## 2 ОТЧЕТ ПО ЗАДАНИЮ 2

### 2.1 Постановка задачи

Создать новый проект в составе решения задания 1. Выполнить все требования задания 1, только для динамической таблицы. Скопировать функции, которые можно использовать без изменения для динамической таблицы. Изменить функции вставки и добавления записей в динамическую таблицу. Алгоритм формирования списка (указано в некоторых вариантах), выполнить на динамическом массиве. Разработать алгоритм выполнения программы, демонстрирующей выполнение операций над таблицей по разработанным тестам.

### 2.2 Определение структуры записи и структуры таблицы средствами языка программирования.

```
struct human
{
    string name;
    string surname;
    string phone;
    int dataD;
    int dataM;
    int dataY;
    string address;
```

Рисунок 15 – Структура записи

```
struct table
{
    int n = 100;
    human* a = new human[n];
```

Рисунок 16 – Структура таблицы

### 2.3 Код основной задачи

```
#include <iostream> //динамич. массив.
#include <string>;
using namespace std;

struct human
{
    string name;
    string surname;
    string phone;
    int dataD;
    int dataM;
    int dataY;
    string address;

    void print()
    {
        cout << "Имя: " << name << ". " << "Фамилия: " << surname << ". " << "Телефон: " << phone << ". " << "День рождения: " << dataD << ". " << dataM << ". " << dataY << " " << "Адрес: " << address;
    }
};
```

```

struct table
{
    int n = 100;
    human* a = new human[n];

    void print() //фун-ция для вывода списка друзей.
    {
        cout << "Таблица: " << "\n" << "===== " <<
"\n";
        for (int i = 0; i < n; i++)
        {
            a[i].print();
            cout << "\n";
        }
    }

    human add() //добавление информации о каждом друге.
    {
        human B;
        setlocale(LC_ALL, "Russian");
        cout << "Введите Имя | Фамилию | Телефон | День рождение | Адрес: ";
        cin >> B.name >> B.surname >> B.phone >> B.dataD >> B.dataM >> B.dataY >> B.address;
        return B;
    }

    void insert(human temp) //фун-ция для поиска и всвтавки записи о новом человеке, если первая буква его фамилии совпала с
имеющимися данными, функ-я добавления записи.
    {
        int k = -1; //флаг.

        for (int i = 0; i < n; i++)
        {
            if (a[i].surname[0] == temp.surname[0])
            {
                k = i;
                break;
            }
        }

        if (k != -1)
        {
            for (int i = n; i > k; i--)
            {
                a[i] = a[i - 1];
            }
            a[k] = temp;
        }
        else
        {
            a[n] = temp;
        }
        n = n + 1;
    }

    void Delete(string tel) //фун-ция для удаления записи по номеру телефона.
    {
        int k = -1; //флаг.

        for (int i = 0; i < n; i++)
        {
            if (a[i].phone == tel)
            {
                k = i;
                break;
            }
        }

        if (k != -1)
        {
            for (int i = k; i < n; i++)
            {

```

```

        a[i] = a[i + 1];
    }
    n = n - 1;
}
else
{
    cout << "error";
}
}

void List(int M) //функция для создания списка друзей, чем месяц рождения совпал.
{
    for (int i = 0; i < n; i++)
    {
        if (a[i].dataM == M)
        {
            a[i].print();
            cout << "\n";
        }
    }
}

void search(string sur) //функция по выводу дня\месяца\телефона искомого человека при вводе фамилии.
{
    for (int i = 0; i < n; i++)
    {
        if (a[i].surname == sur)
        {
            cout << "День: " << a[i].dataD << " " << "Месяц: " << a[i].dataM << " " << "Год: " << a[i].dataY << " " <<
"Телефон: " << a[i].phone;
            cout << "\n";
        }
    }
};

void userchoice(table t) //функция для оформления консоли выбора.
{
    setlocale(LC_ALL, "Russian");
    int choice;
    cout << "Выберите действие: \n 1 - Вставить запись. \n 2 - Удалить запись по заданному телефону. \n 3 - Сформировать список
друзей, родившихся в заданном месяце. \n 4 - Найти дату рождения и номер телефона друга по его фамилии. \n ";
    cin >> choice;

    string numberV;
    int mounth;
    string secondname;

    switch (choice)
    {
    case 1:
        t.insert(t.add());
        t.print();
        break;

    case 2:
        cout << "Введите номер телефона: ";
        cin >> numberV;
        t.Delete(numberV);
        t.print();
        break;

    case 3:
        cout << "Введите номер месяца: ";
        cin >> mounth;
        t.List(mounth);
        break;
        t.print();

    case 4:
        cout << "Введите искомую фамилию: ";
        cin >> secondname;
        t.search(secondname);
        break;
    }
}
}

```

```
int main()
{
    table t;
    setlocale(LC_ALL, "Russian");
    system("chcp 1251");
    t.n = 5; //база данных для изначального списка друзей.
    t.a[0] = { "Петя", "Павлов", "111", 10, 2, 2002, "Псковская область, город Дорохово, проезд Ладыгина, 00" };
    t.a[1] = { "Рита", "Радова", "79775678456", 7, 8, 2000, "Кировская область, город Серебряные Пруды, въезд Балканская, 41" };
    t.a[2] = { "Саша", "Гаев", "73449543733", 10, 2, 2001, "Мурманская область, город Чехов, въезд Будапештская, 75" };
    t.a[3] = { "Денис", "Курбанаев", "73444543733", 25, 5, 2004, "Орловская область, город Павловский Посад, наб. Ладыгина, 76" };
    t.a[4] = { "Полина", "Шумакова", "74549543733", 15, 3, 2004, "Самарская область, город Одинцово, бульвар Гоголя, 09" };
    userchoice(t);
}
```

## 3 ОТЧЕТ ПО ЗАДАНИЮ 3

### 3.1 Код основной задачи

```
#include <iostream> //3 векторный массив.
#include <string>;
#include <iostream>
#include <vector>

using namespace std;

struct human
{
    string name;
    string surname;
    string phone;
    int dataD;
    int dataM;
    int dataY;
    string address;

    void print()
    {
        cout << "Имя: " << name << ". " << "Фамилия: " << surname << ". " << "Телефон: " << phone << ". " << "День рождение:
" << dataD << ". " << dataM << ". " << dataY << " " << "Адрес: " << address;
    }
};

struct table
{
    vector <human> a;

    void print() //фун-ция для вывода списка друзей.
    {
        cout << "Таблица: " << "\n" << "===== " <<
"\n";
        for (int i = 0; i < a.size(); i++)
        {
            a[i].print();
            cout << "\n";
        }
    }

    human add() //добавление информации о каждом друге.
    {
        human B;
        setlocale(LC_ALL, "Russian");
        cout << "Введите Имя | Фамилию | Телефон | День рождение | Адрес: ";
        cin >> B.name >> B.surname >> B.phone >> B.dataD >> B.dataM >> B.dataY >> B.address;
        return B;
    }

    void insert(human temp) //фун-ция для поиска и всвтавки записи о новом человеке, если первая буква его фамилии совпала с
имеющимися данными, функ-я добавления записи.
    {
        int k = -1; //флаг.

        for (int i = 0; i < a.size(); i++)
        {
            if (a[i].surname[0] == temp.surname[0])
            {
                k = i;
                break;
            }
        }

        if (k != -1)
        {
            for (int i = a.size(); i > k; i--)
```

```

        {
            a[i] = a[i - 1];
        }
        a[k] = temp;
    }
    else
    {
        a[a.size() - 1] = temp;
    }
}

void Delete(string tel) //фун-ция для удаления записи по номеру телефона.
{
    int k;

    for (int i = 0; i < a.size(); i++)
    {
        if (a[i].phone == tel)
        {
            a.erase(a.begin()+i);
            break;
        }
    }
}

void List(int M) //фун-ция для создания списка друзей, чей месяц рождения совпал.
{
    for (int i = 0; i < a.size(); i++)
    {
        if (a[i].dataM == M)
        {
            a[i].print();
            cout << "\n";
        }
    }
}

void search(string sur) //фун-ция по выводу дня\месяца\телефона искомого человека при вводе фамилии.
{
    for (int i = 0; i < a.size(); i++)
    {
        if (a[i].surname == sur)
        {
            cout << "День: " << a[i].dataD << " " << "Месяц: " << a[i].dataM << " " << "Год: " << a[i].dataY << " " <<
"Телефон: " << a[i].phone;
            cout << "\n";
        }
    }
}

};

void userchoice(table t) //фун-ция для оформления консоли выбора.
{
    setlocale(LC_ALL, "Russian");
    int choice;
    cout << "Выберите действие: \n 1 - Вставить запись. \n 2 - Удалить запись по заданному телефону. \n 3 - Сформировать список
друзей, родившихся в заданном месяце. \n 4 - Найти дату рождения и номер телефона друга по его фамилии. \n ";
    cin >> choice;

    string numberV;
    int mounth;
    string secondname;

    switch (choice)
    {
    case 1:
        t.insert(t.add());
        t.print();
        break;
    case 2:
        cout << "Введите номер телефона: ";

```

```

        cin >> numberV;
        t.Delete(numberV);
        t.print();
        break;
    case 3:
        cout << "Введите номер месяца: ";
        cin >> mounth;
        t.List(mounth);
        break;
        t.print();
    case 4:
        cout << "Введите искомую фамилию: ";
        cin >> secondname;
        t.search(secondname);
        break;
    }
}

int main()
{
    table t;
    setlocale(LC_ALL, "Russian");
    system("chcp 1251");
    t.a.push_back({ "Петя", "Павлов", "111", 10, 2, 2002, "Псковская область, город Дорохово, проезд Ладыгина, 00" });
    t.a.push_back({ "Рита", "Радова", "79775678456", 7, 8, 2000, "Кировская область, город Серебряные Пруды, въезд Балканская,
41" });
    t.a.push_back({ "Саша", "Гаев", "73449543733", 10, 2, 2001, "Мурманская область, город Чехов, въезд Будапештская, 75" });
    t.a.push_back({ "Денис", "Курбанаев", "73444543733", 25, 5, 2004, "Орловская область, город Павловский Посад, наб. Ладыгина,
76" });
    t.a.push_back({ "Полина", "Шумакова", "74549543733", 15, 3, 2004, "Самарская область, город Одинцово, бульвар Гоголя, 09" });
    userchoice(t);
}

```



#### **4 ЗАКЛЮЧЕНИЕ**

В ходе работы была выявлена разница по реализации многоэлементных структур данных разными способами. Наиболее эффективным и удобным оказался способ использования контейнера `<vector>`.

## **5 СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ**

1. Материалы по дисциплине (Сорокин А.В.) / Занятие - быстрые сортировки
2. Скворцова Л.А., Гусев К.В., Трушин С.М., Филатов А.С. Учебно-методическое пособие СИАОД часть 1