



# Структуры

**Пример. Сведения о студентах группы.**

*Иллюстративный материал к  
лекциям по алгоритмизации и  
программированию*

Автор Саблина Н.Г.

2016 г.





# Содержание

Постановка задачи

Метод решения

Схема алгоритма записи данных

Текст программы ввода данных

Схема алгоритма вычисления среднего  
возраста

Текст программы

Об  
авторе





# Постановка задачи

## *Условие задачи*

Сохраним в файле следующие сведения о студентах некоторой группы: фамилию, имя, возраст, рост.

Вычислим средний возраст студентов группы и определим самого высокого студента в группе.

### **1) Постановка задачи**

- *Исходными данными* для этой программы являются сведения о студентах: фамилия, имя – строковые данные; возраст – натуральное число, рост в метрах – действительное число. Все данные вводятся с клавиатуры.
- *Выходные данные* – файл, содержащий сведения, средний возраст студентов – действительное число, фамилия, имя – строковые данные, рост самого высокого студента в группе – действительное число. Сведения о самом высоком студенте выводятся на экран





# Метод решения

- Для организации такого списка студентов создадим тип `student`, представляющий собой структуру с соответствующими полями:
  - фамилия - строковое поле;
  - имя - строковое поле;
  - возраст – поле целого положительного типа,
  - рост в метрах– действительное число.
- Сведения о студентах сохраним в файле `group.dat`. Тип элементов этого файла – `student`.
- Разделим решение этой задачи на две независимые подзадачи:
  - ввод данных о студентах и запись этих данных в файл;
  - вычисление среднего возраста и определение самого высокого студента в группе на основании данных из файла.





# Метод решения. Подзадача 1 (1 из 2)

- Основной целью этой подзадачи является организация диалога ввода исходных данных.
  - Сначала запрашивается количество студентов в группе (n).
  - Затем в цикле вводятся данные о каждом студенте.
- Эти сведения заносятся в соответствующие поля переменной person типа student.
- Для обращения к отдельному полю структуры person используются составные имена, например,
  - person.famil, person.rost и т.п.





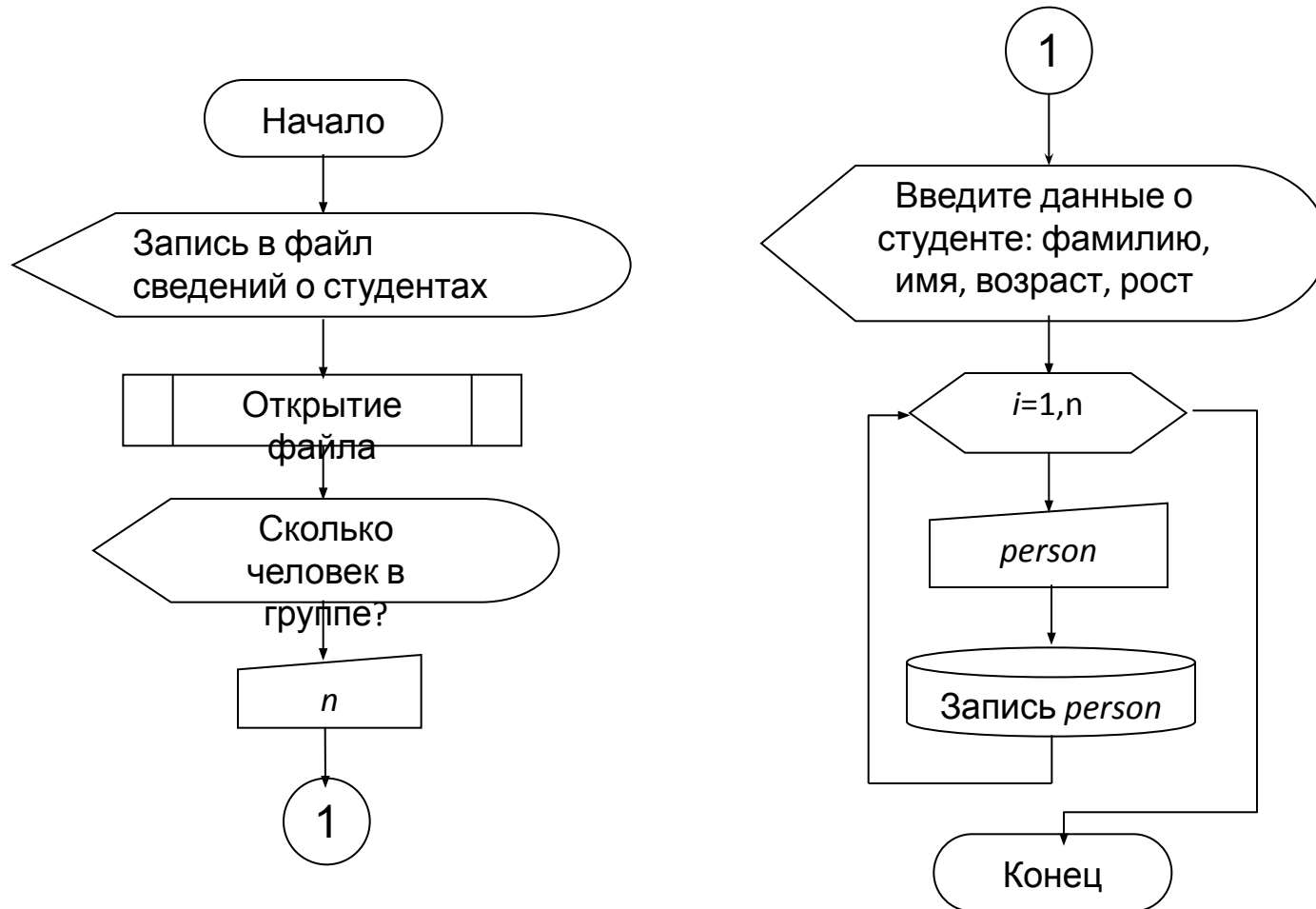
# Метод решения. Подзадача 1 (2 из 2)

- Полностью сформированная запись об  $i$ -м студенте сразу же записывается в файл с помощью одной операции записи.
- Использование структурированного типа `student` делает возможным осуществлять запись в файл всех сведений о студенте за одну операцию записи.
- После записи в файл сведений об  $i$ -ом студенте переменная `person` снова используется на следующем шаге цикла для сбора сведений об  $(i+1)$  – ом студенте.
- Т.о. не требуется массива структур для хранения данных о группе в целом. Хранилищем сведений о группе является файл





# Схема алгоритма записи данных

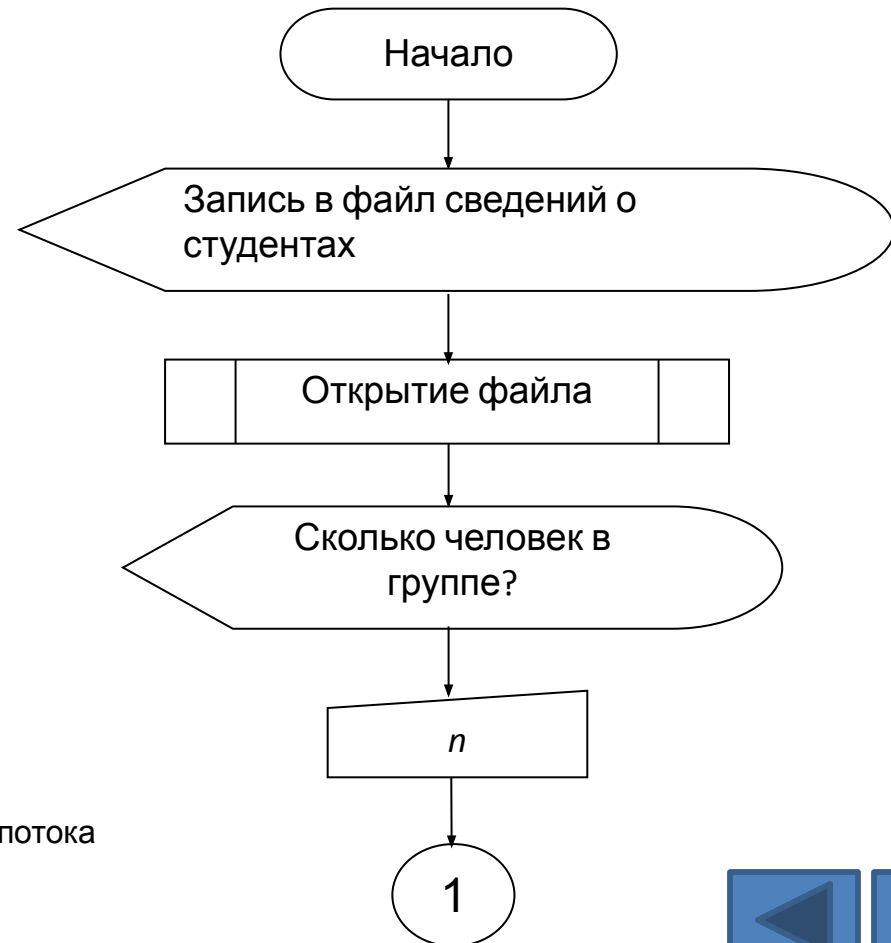




# Текст программы ввода данных (1 из 2)

```
int main()
{
    struct student{
        char famil[20];
        char name[15];
        unsigned int let;
        float rost;
    };

    int i,n;
    student person;
    FILE *group;
    char* file_name="gruppa.dat";
    group=fopen(file_name,"w");
    cout<<"Сколько человек в группе?";
    cin>>n;
    fflush(stdin); //очистка буфера входного потока
```



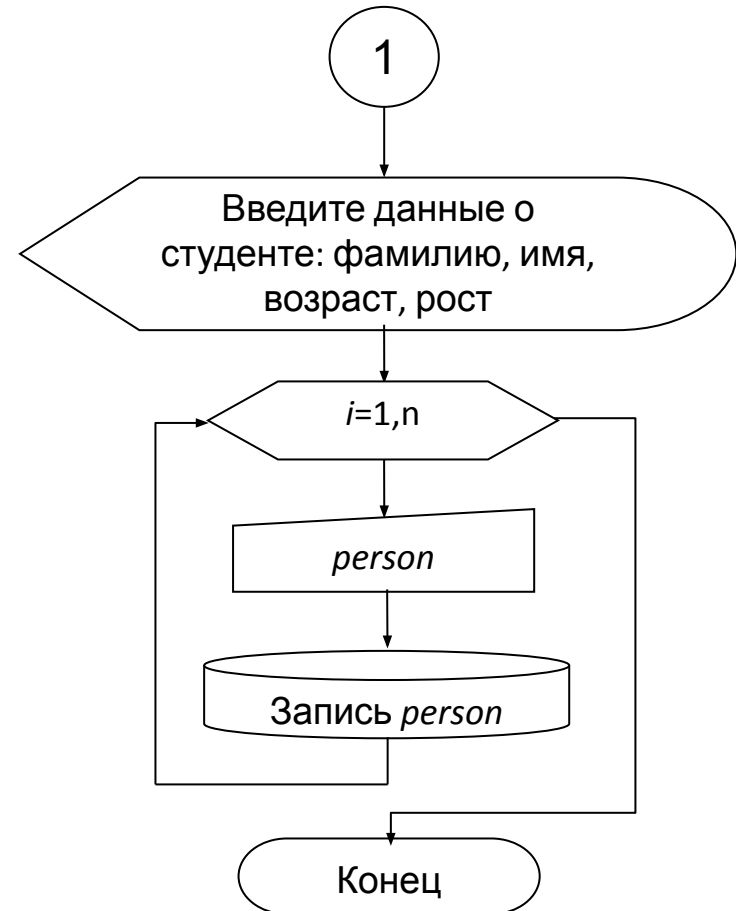




# Текст программы ввода данных (2 из 2)

```
cout<<"Введите данные о студнте:";
cout<<"фамилия, имя,возраст, рост";
```

```
for(i=1;i<=n;i++)
{cout<<"\n"<<i<<" "};
cin>>person.famil;
cin>>person.name;
cin>>person.let;
cin>>person.rost;
fwrite(&person, sizeof(student),1,group);
}
fclose(group);
}
```





# Метод решения. Подзадача 2 (1 из 4)

- Исходные данные для второй задачи вводятся из файла group.dat.
- Сведения из файла считываются последовательно по одной записи о студенте за одну операцию чтения. Так повторяется пока не будет достигнут конец файла. Т.о. процесс считывания записей из файла циклический.

- Средний возраст студентов в группе определяется по формуле

$$w_{cp} = \frac{\sum_{i=1}^n w_i}{n}$$

, где  $w_i$  – возраст  $i$ -ого студента,  $n$  – количество

студентов в группе





# Метод решения. Подзадача 2 (2 из 4)

- Для определения самого высокого в группе будем сравнивать рост  $i$ -ого студента, сведения о котором прочитали из файла на текущем шаге цикла, с ростом самого высокого из уже прочитанных ранее из файла сведений.
- Если рост  $i$ -ого оказался больше того значения, которое считалось максимальным до этого, то его значение запоминаем как самый высокий рост, а также запоминаем номер его записи в файле -  $k$ .





# Метод решения. Подзадача 2 (3 из 4)

- Таким образом, в теле цикла выполняются следующие действия:
  - считывание данных об очередном студенте в переменную `person`, вывод на экран сведений о нем,
  - добавление его возраста к суммарному возрасту группы,
  - сравнение роста данного студента с ростом других уже просмотренных студентов.
- Кроме того, имеется счетчик студентов  $i$ .

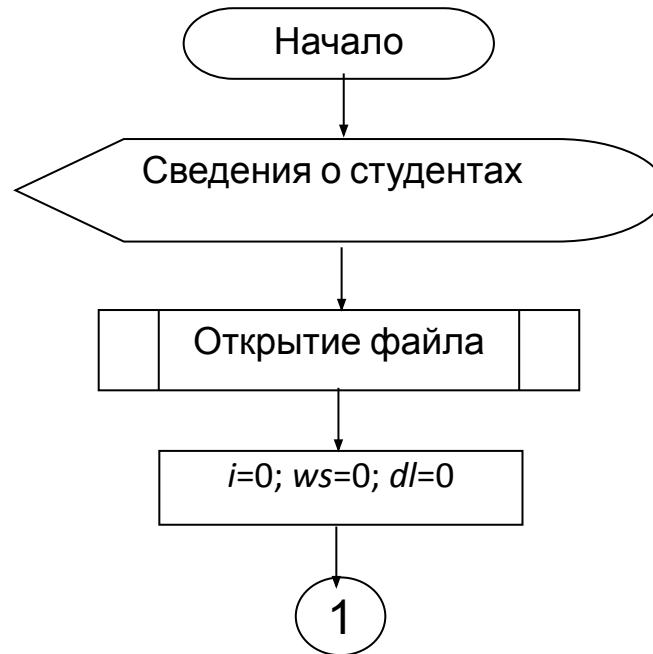


# Метод решения. Подзадача 2 (4 из 4)

- Считывание записи из файла в переменную `person` осуществляется за одну операцию чтения (так же как и запись), а при выводе на экран нужно обратиться к каждому полю записи отдельно.
- После просмотра всех записей указатель файла переводится на запись с номером  $k$ , производится считывание и вывод на экран сведений о самом высоком студенте в группе.

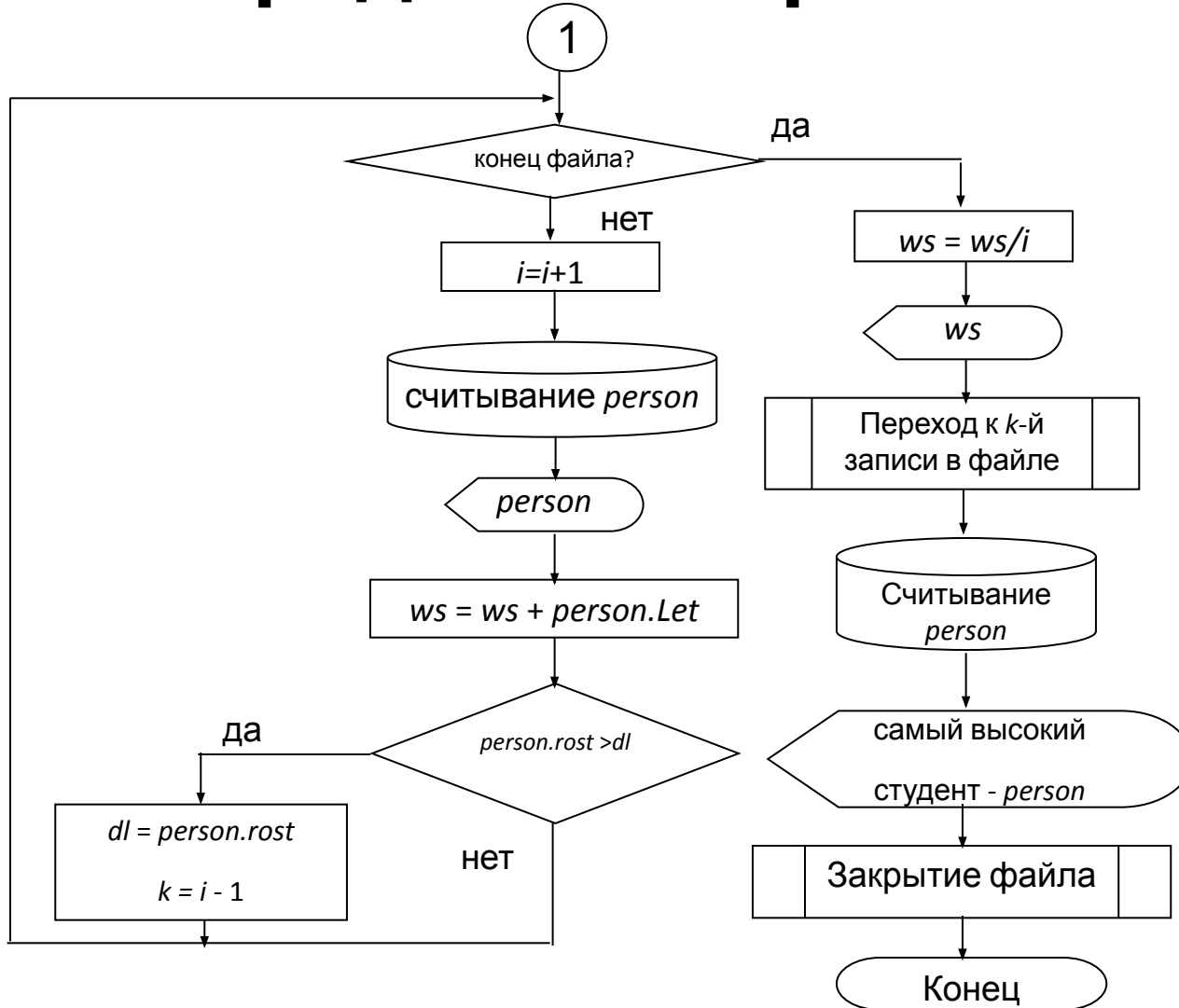


# Схема алгоритма вычисления среднего возраста





# Схема алгоритма вычисления среднего возраста





# Текст программы (1 из 4)

```
#include <conio.h>
#include <stdio.h>
#include <io.h>
#include <iostream.h>
int main()
{ struct student
    {
        char famil[20];    //фамилия
        char name[15];    //имя
        unsigned int let;    //возраст
        float rost;    // рост
    } ;
```







# Текст программы (2 из 4)

```
student person;  
  
    FILE *group;  
  
    int i,k,ws;  
  
    float dl;  
  
cout<<"\n Сведения о студентах группы";  
  
//открытие файла для чтения  
  
group=fopen("gruppa.dat","r");  
  
i=0; dl=0; ws=0;
```





# Текст программы (3 из 4)

```
while (!feof(group)) // достигнут конец файла
{ // считывание записи из файла
    fread(&person,sizeof(person),1,group);
    if (feof(group)) break;
    ++i; //счетчик студентов
    ws=ws+person.let; //суммарный возраст
    if (person.rost>dl) { dl=person.rost; k=i-1; }
    //вывод на экран
    cout<<"\n"<<i<<" " <<person.famil<<" " <<person.name;
    cout<<" , vozrast- " <<person.let; cout<<" let, rost- " <<person.rost<<" м.";
}
```





# Текст программы (4 из 4)

```
float sr=ws/(i);  
cout<<"\n Средний возраст- "<<sr<<" лет";  
long p=k*sizeof(student)+1; fseek (group,p,0);  
fread(&person,sizeof(student),1,group);  
cout<<"\n Самый высокий в группе - "<<person.famil<<"  
    "<<person.name;  
cout<<". Его рост- "<<person.rost<<" м.";  
fclose(group); // {закрытие файла}  
getch();  
return 0;}
```





Автор:

Саблина Наталья Григорьевна

Ст. преподаватель

каф. РТС УрФУ

